Süleyman Demirel Üniversitesi
Fen Bilimleri Enstitüsü Dergisi
Cilt 29, Sayı 2, 388-396, 2025

Süleyman Demirel University
Journal of Natural and Applied Sciences
Volume 29, Issue 2, 388-396, 2025

DOI: 10.19113/sdufenbed.1655066

# Sistem Çağrısı Verilerinde Derin Öğrenme Mimarileri Kullanılarak Anomali Tespitinin Değerlendirilmesi: Performans ve Enerji Verimliliği

**İbrahim ŞANLIALP*[1], Mehmet YILDIZ[2]**

[1,2] Kırşehir Ahi Evran Üniversitesi, Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, 40100, Kırşehir, Türkiye

(Alınış / Received: 10.03.2025, Kabul / Accepted: 30.06.2025, Online Yayınlanma / Published Online: 25.08.2025)

**Anahtar Kelimeler**
Sistem çağrısı,
Anomali tespiti,
Enerji verimliliği,
Derin öğrenme

**Öz:** Sistem çağrıları, işletim sistemi ile yazılım arasındaki etkileşimleri temsil eden kritik bir veri kaynağı görevi görmektedir. Bir veri kaynağındaki olağan dışı kalıpların belirlenmesi, sistem performansını iyileştirmek açısından önemli olan anomali tespiti olarak adlandırılmaktadır. Özellikle, derin öğrenme tabanlı yaklaşımlar sistem çağrılarında anomali tespitinin doğruluğunu artırırken, kullanılan derin öğrenme modellerinin enerji verimliliği göz ardı edilemez bir değerlendirme ölçütü olarak öne çıkmaktadır. Bu çalışmada, üç derin öğrenme modelinin (Tekrarlayan Sinir Ağı (RNN), Kapılı Tekrarlayan Birim (GRU) ve Uzun Kısa Süreli Bellek (LSTM)) sistem çağrısı verilerinde anomali tespiti için performansı ve enerji verimliliği karşılaştırılmıştır. Her model, Asgari-Azami Normalleştirmesi (Min-Max Normalization) ile ölçeklendirilmiş verilerle eğitilmiştir. Model performansı, Determinasyon Katsayısı ($R^2$), Ortalama Kare Hatası (MSE) ve Ortalama Mutlak Hata (MAE) değerlendirme metrikleri kullanılarak ölçülmüştür. Enerji tüketimi tahmini ise Intel Power Gadget (IPG) ile gerçekleştirilmiştir. Bu karşılaştırmalı analiz, test edilen mimarilerin göreceli etkinliği hakkında ampirik içgörüler sunmaktadır. Bulgular, LSTM'nin anomali tespiti açısından daha iyi bir performans sergilediğini, ancak enerji verimliliği açısından RNN'nin daha avantajlı olduğunu göstermektedir.

# Evaluation of Anomaly Detection in System Call Data Using Deep Learning Architectures: Performance and Energy Efficiency

**Keywords**
System call,
Anomaly detection,
Energy efficiency,
Deep learning

**Abstract:** System calls serve as a critical data source that represents the interactions between the operating system and software. The identification of unusual patterns within a data source is referred to as anomaly detection, which is essential for improving system performance. Particularly, deep learning-based approaches enhance the accuracy of anomaly detection in system calls, while the energy efficiency of the employed deep learning models emerges as a crucial evaluation criterion that cannot be overlooked. In this study, the performance and energy efficiency of three deep learning models—Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), and Long Short-Term Memory (LSTM)—were compared for anomaly detection in system call data. Each model was trained using data scaled with Min-Max Normalization. Model performance was measured using the evaluation metrics Coefficient of Determination ($R^2$), Mean Squared Error (MSE), and Mean Absolute Error (MAE). Energy consumption estimation was conducted using Intel Power Gadget (IPG). This comparative analysis provides empirical insights into the relative effectiveness of the tested architectures. The findings show that LSTM achieves better performance in anomaly detection, while RNN is more advantageous in terms of energy efficiency.

## 1. Giriş

Modern bilişim sistemlerinde anomali tespiti veri güvenliğinin ve sistem performansının temel yapı

*Sorumlu yazar: ibrahim.sanlialp@ahievran.edu.tr

taşlarındandır. Anomaliler, genellikle sistemde beklenen duruma aykırı olan davranışları ifade eder. Bu anomaliler sistem performansını olumsuz etkileyebilir veya sisteme yönelik gerçekleştirilen saldırıları işaret edebilir [1]. Sistem çağrıları (system calls), işletim sistemi ile çalışan uygulamalar arasında gerçekleştirilen temel etkileşimleri temsil eder ve bu nedenle anomali tespiti için önemli bir veri kaynağıdır.

Sistem çağrısı verilerinde anomalileri tespit etmek bu etkileşimlerin sıralı bir yapıya sahip olması sebebiyle zordur. Bu tür sıralı verilerde, her bir adımın öncekilerle olan ilişkisini ve uzun vadeli bağımlılıklarını anlamak oldukça önemlidir. Makine öğrenimindeki hızlı ilerleme ile anomali tespitinin entegrasyonu, sistem güvenliğini ve performans verimliliğini artırmak için son derece etkili çözümler sunmuştur [2]. Yapay sinir ağları, özellikle sıralı veriler için optimize edilmiş olan tekrar eden sinir ağı türleri, bu uzun bağımlılıkları anlamak ve anormallikleri tespit etmek için güçlü modeller sunar [3, 4]. Fakat modellerin enerji tüketiminin tespit edilmesiyle, modellerin performans ve enerji tüketiminin birlikte ele alınması ihtiyacını beraberinde getirmektedir [5, 6]. Çünkü, veri odaklı teknolojilerin ve dijital sistemlerin hızlı büyümesi aynı zamanda karmaşık hesaplama görevlerini minimum çevresel ve finansal etkiyle ele alabilen enerji açısından verimli algoritmalara ihtiyaç vardır.

Uzun Kısa Süreli Bellek (LSTM), Tekrarlayan Sinir Ağı (RNN) ve Kapılı Tekrarlayan Birim (GRU) modelleri, sistem çağrısı veri analizi de dahil olmak üzere çeşitli alanlarda zaman serisi tahmini ve anomali tespiti için yaygın olarak kullanılan, öne çıkan RNN tabanlı mimarilerdir.

RNN modellerinin anomali tespitinde uygulanması, bu mimarilerin ardışık verilerdeki uzun vadeli bağımlılıkları yakalama konusundaki başarısı nedeniyle etkili bulunmuştur. Örneğin, Nguyen ve arkadaşları [2], LSTM'nin tedarik zinciri yönetim sistemlerindeki anomalileri tespit etme potansiyelini göstererek, geleneksel makine öğrenimi yöntemlerine kıyasla üstün performansını vurgulamıştır. Benzer şekilde, Kaya ve Yıldız [7], LSTM ağlarının İnsansız Hava Aracı (İHA) sistem çağrısı verilerindeki anormallikleri belirlemedeki etkinliğini ortaya koymuş, uzun vadeli bağımlılıkları tespit etme ve sistem güvenliğini iyileştirme konularındaki yetkinliğini göstermiştir.

Başka bir çalışmada ise, RNN ve GRU modelleri karşılaştırılmış ve GRU'nun daha uzun vadeli bağımlılıkları işleme kapasitesine sahip olduğu gösterilmiştir. Ancak, daha kısa eğitim süreleri ve düşük hesaplama maliyetleri nedeniyle belirli uygulamalarda RNN tercih edilmiştir [8]. Bu modeller yüksek doğruluk sağlamalarına rağmen, özellikle LSTM, genellikle yüksek hesaplama gücü ve enerji tüketimi gerektirdiği için eleştirilmektedir. Yeşil bilişime artan ilgi, enerji açısından verimli algoritmaların önemini ön plana çıkarmaktadır. Örneğin, Rodriguez ve arkadaşları [9], algoritmik performans ve enerji tüketimi arasında bir denge çağrısında bulunarak, işlevsellikten ödün vermeden karbon ayak izlerini en aza indiren "Yeşil AI" çözümlerinin geliştirilmesini önermiştir.

Enerji tüketimi bağlamında, çeşitli çalışmalar makine öğrenimi modellerinin enerji verimliliğini değerlendirmek ve optimize etmek için çeşitli yaklaşımlar sunmuştur. Örneğin, Fahad ve arkadaşları [10], enerji ölçüm araçlarının doğruluğunu analiz etmiş ve tahmin modelleri ile donanım düzeyindeki sensörler arasındaki tutarsızlıkları vurgulamıştır. Ayrıca, Intel Power Gadget (IPG) [11] gibi yenilikçi araçların kullanımı, özellikle farklı makine öğrenimi mimarilerinin performansını kıyaslamak için yararlı olan hassas enerji profillemesini mümkün kılmıştır.

Son yıllarda yapılan çalışmalar, derin öğrenme mimarileri kullanılarak sistem çağrısı verilerinde yüksek doğrulukla anomali tespiti yapılabildiğini, ancak bu süreçte model mimarilerinin enerji tüketimi gibi kaynak verimliliği açısından da optimize edilmesi gerektiğini göstermektedir. Örneğin, Linux çekirdeği için özel olarak hazırlanmış bir sistem çağrısı veri kümesinde, sistem çağrısı dizileri kullanılarak anomali tespiti gerçekleştirilmiş ve derin öğrenme modellerinin çekirdek seviyesinde uygulanabilirliği analiz edilmiştir. Çalışma, yüksek doğrulukta sonuçlar sunan modellerin gerçek zamanlı sistemlere entegrasyonu için mimari ve kaynak verimliliği optimizasyonuna ihtiyaçlarını vurgulamaktadır [12]. Buna ek olarak, yazılım uygulamalarında enerji tüketimini öngörmek amacıyla performans sayacı olayları ve sistem çağrısı verilerini birleştirerek makine öğrenimi modelleri geliştirilmiştir. Geliştirilen çalışma, enerji tüketimini tahmin etmede sistem çağrılarının önemli bir özellik olarak kullanılabileceğini ve bu verilerin işlenmesinde kullanılan model mimarilerinin enerji verimliliği açısından optimize edilmesi gerektiği vurgulamıştır [13]. Bu bulgular, sistem çağrısı verilerinin işlendiği derin öğrenme uygulamaları açısından hem enerji verimliliği hem de performans konularının kritik önem taşıdığını ortaya koymaktadır.

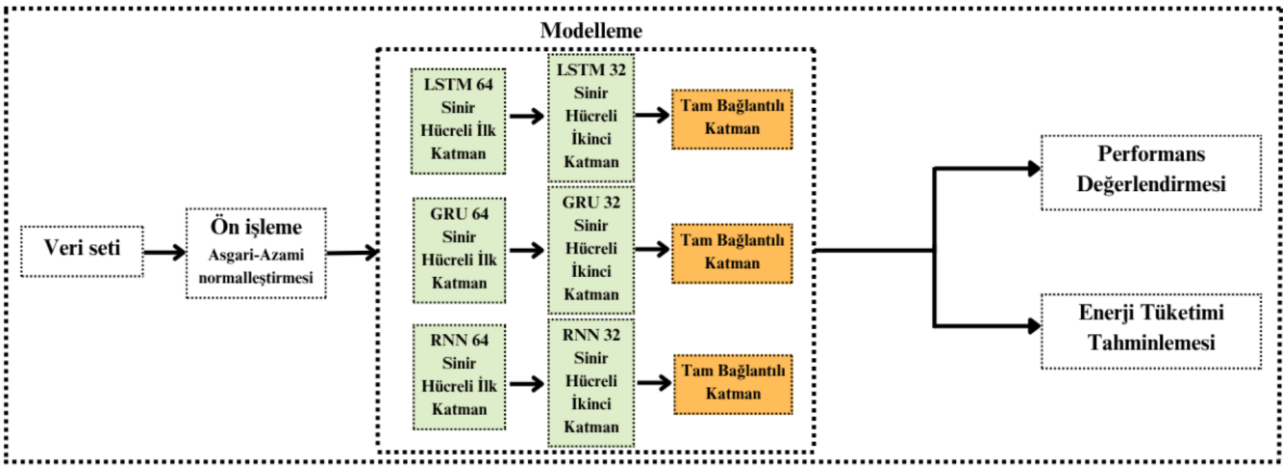
Bu çalışma, sistem çağrısı veri kümelerinde anomali tespitinde LSTM [2], GRU [14] ve RNN [8] mimarilerinin performanslarını karşılaştırmakta ve analiz etmektedir. Çalışma, yalnızca anomali tespitindeki doğruluk oranlarını ele almakla kalmayıp, aynı zamanda artan sürdürülebilirlik ve verimli bilişim odakları göz önünde bulundurularak enerji tüketimine de vurgu yapmaktadır. Çalışma ayrıca sürdürülebilir ve verimli makine öğrenimi uygulamalarının geliştirilmesine katkıda bulunmayı amaçlamaktadır. Bu araştırmanın üç temel katkısı vardır. İlk olarak, sistem çağrıları üzerinde LSTM, GRU ve RNN mimarilerinin anomali tespit performansları

üzerinde kapsamlı bir karşılaştırmayı ele alır. İkinci olarak, her modelin enerji verimliliği değerlendirilerek yürütme süresi ile enerji tüketimi arasındaki korelasyona ilişkin detaylı bilgiler sağlar. Üçüncüsü, çalışmanın sonuçları sürdürülebilir bilişim açısından verimli anomali tespit sistemleri oluşturmak için pratik öneriler sunar.

Makalenin geri kalanı şu şekilde organize edilmiştir: 2. bölümde çalışmada kullanılan materyaller ve çalışma için geliştirilen yöntem açıklanmaktadır. 3. bölüm çalışmanın bulgularını içermektedir. 4. bölümde ise tartışma ve sonuçlar yer almaktadır.

## 2. Materyal ve Metot

Bu bölüm, deneysel çalışmanın kavramsal çerçevesinin ayrıntılı bir gösterimi, veri seti tanımı, uygulanan RNN tabanlı mimarileri, değerlendirme metrikleri bilgisi, sistem ve araç yapılandırması dahil olmak üzere materyal ve metodu içermektedir. Şekil1 bu çalışmanın kavramsal çerçevesini göstermektedir.




**Şekil 1**. Çalışmanın kavramsal çerçevesinin genel görünümü

### 2.1. Veri seti tanımı

Bu çalışmada, anomali tespiti için sistem çağrıları üzerine odaklanan bir veri seti kullanılmıştır. Kullanılan veri seti [15], bir İHA'nın simüle edilmiş bir platform üzerinde çalıştırılması sırasında kaydedilen sistem çağrısı olaylarını içermektedir. Veri seti, bağlam modelleme sürecine odaklanarak, süreçler ve çekirdek arasındaki karmaşık etkileşimleri modellemek için 434.309 sistem çağrısı olayını içermektedir. Sistem çağrılarının sırası ve türleri ayrıntılı şekilde kaydedildiğinden, bağlam modelleme ve anomali tespiti açısından zengin bir içerik sunmaktadır. Ayrıca veri seti, İHA uygulamasının davranışını, durum makineleri aracılığıyla başlangıç durumundan bitiş durumuna kadar gözlemlemek için uygun bir altyapı sağlamaktadır.

### 2.2. Veri ön işleme

Model eğitimi öncesi sistem çağrısı verilerine çeşitli ön işleme adımları uygulanmıştır. Sayısal özellikler Asgari-Azami Normalleştirmesi (Min-Max Normalizasyonu) [18] ile [0,1] aralığına ölçeklendirilerek farklı değişkenlerin model üzerinde baskınlık kurması önlenmiştir. Kategorik sistem çağrısı kimlikleri ise uygun şekilde sayısal temsillere dönüştürülmüştür. Ardından, zaman serisinin ardışık yapısını koruyacak biçimde sabit uzunlukta kayma pencereleri ile giriş-çıktı çiftleri oluşturulmuştur. Bu işlem sayesinde modelin sıralı bağımlılıkları öğrenmesi ve anomali paternlerini tanıyabilmesi sağlanmıştır. Uygulanan ön işleme adımları, modelin hem öğrenme sürecini hızlandırmış hem de tahmin doğruluğunu artırmıştır.

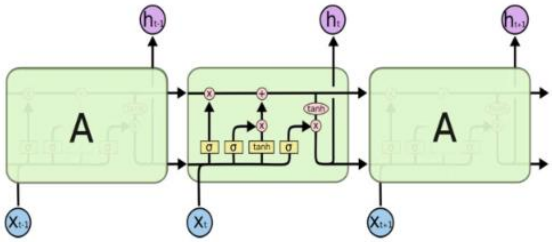
### 2.3. Modelleme

Bu aşama, sistem çağrısı veri kümeleri üzerinde uygulanan RNN, GRU ve LSTM olmak üzere üç farklı derin öğrenme mimarisiyle geliştirilen anomali tespiti modellerinin açıklanmasını içerir. Her modelin zaman serisi verilerini işlemede kendine özgü avantajları vardır. Modeller, anomalileri tespit etmedeki performansları ve enerji tüketim tahminlemeleri açısından değerlendirilmiştir. Mimarilerin modelleme süreci, hiperparametre ayarı, eğitim ve model değerlendirmesi olmak üzere alt aşamalardan oluşmaktadır. Bu çalışmada, RNN, GRU ve LSTM modellerinin her biri için hiperparametreler aynı şekilde belirlenmiştir. Her bir modeldeki son katman, anomali tahminlerini çıkış olarak üreten tek bir nörona sahip yoğun (dense) veya tam bağlantılı (fully connected) katmandır. Bu katman, önceki katmanlardan gelen tüm bilgileri birleştirerek anomalileri tahmin eder. Giriş dizi uzunluğu (sequence length) 20 olarak seçilmiş; her modelde 64 sinir hücreli(nöron) birinci katman ve 32 sinir hücreli ikinci katman kullanılmıştır. Aşırı öğrenmeyi (overfitting) önlemek amacıyla her katmandan sonra %20 oranında seyreltme (dropout) uygulanmıştır. Optimizasyon sürecinde öğrenme oranı 0.001 olarak belirlenmiştir. Eğitim/test oranı %70 ve %30 olarak düzenlenmiş; eğitim işlemi 30 epok boyunca ve 32 örnekten oluşan küme boyutuyla (batch size)

gerçekleştirilmiştir. Kayıp fonksiyonu olarak ortalama kare hata seçilmiş; model performansı doğrulama verisi de kullanılarak detaylı bir şekilde analiz edilmiştir.

### 2.3.1. Uzun kısa süreli bellek (LSTM)

LSTM, Hochreiter ve Schmidhuber tarafından 1997 yılında, RNN mimarisinin bir uzantısı olarak tanıtılmıştır ve özellikle uzun vadeli bağımlılıkları öğrenmek amacıyla tasarlanmıştır [16]. LSTM'deki her nöron, önceki gizli durumundan gelen girdiyi yeni bir gizli duruma dönüştürmek yerine, kendi "bellek" durumunu koruyarak bilgi depolayabilir. Bu mimari, geleneksel RNN'lerin kaybolan gradyan (vanishing gradient) problemi nedeniyle uzun süreli bağımlılıkları öğrenememesi sorununa çözüm sunar [17].




**Şekil 2**. LSTM mimarisi [14]

Şekil 2'de gösterildiği gibi, LSTM mimarisi, sıralı girdileri işleyebilme ve sıralı çıktılar üretebilme yeteneğiyle öne çıkar. Her bir zaman adımında, LSTM hücresine bir giriş vektörü $x_t$ verilir ve bu hücre, içinde bulunan çeşitli kapılar ve fonksiyonlar aracılığıyla bir durum vektörü $h_t$ oluşturur. LSTM'lerde, çıktı vektörü $h_t$ , durum vektöründen farklı bir fonksiyon tarafından üretilir ve bu çıktı, bir sonraki zaman adımında yeni giriş vektörü $x_t$ ile birlikte kullanılır. LSTM'nin çalışma prensibi, dört temel adıma dayanır. İlk olarak, unutma kapısı (forget gate), bir önceki zaman adımının durum vektörü $h_{t-1}$ ve mevcut giriş vektörü $x_t$ 'yi birleştirerek hangi bilgilerin unutulacağını belirler [19]. Bu işlem, denklem (1) formülüyle ifade edilir [14]. İkinci adımda, giriş kapısı (input gate), yeni bilgilerin hücre durumuna ne ölçüde ekleneceğini belirler. Bu, denklem (2) ve denklem (3) formülleriyle hesaplanır. Üçüncü adımda, hücre durumu güncelleme (cell state update), unutma kapısı ve giriş kapısının çıktılarını kullanarak hücre durumunu günceller: denklem (4) . Son olarak, çıkış kapısı (output gate), denklem (5) formülüyle hesaplanır ve mevcut zaman adımının çıktısı denklem (6) şeklinde üretilir. Bu süreçte kullanılan tüm ağırlık matrisleri, modelin eğitim aşamasında optimizasyon yoluyla belirlenir [14, 20].

$$f_t = \sigma_f\left(W_f \cdot [h_{t-1}, x_t] + b\right) \quad (1)$$

burada $f_t$ unutma kapısını temsil ederken, $\sigma_f$ sigmoid aktivasyon fonksiyonunu, $W_f$ ağırlık matrisini, $h_{t-1}$bir önceki zaman adımının gizli durumunu, $x_t$ mevcut zaman adımının girişi ve $b$ bias terimini temsil eder.

$$i_t = \sigma_i(W_i \cdot [h_{t-1}, x_t] + b) \quad (2)$$

$$N_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b) \quad (3)$$

burada $i_t$ giriş kapısını temsil ederken, $N_t$ yeni hücre durumu adayını, $W_i$ ve $Wc$ ağırlık matrislerini, tanh hiperbolik tanjant aktivasyon fonksiyonunu, $b$ bias terimlerini temsil eder.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot N_t \quad (4)$$

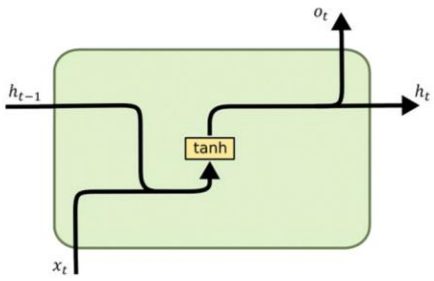
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b) \quad (5)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (6)$$

burada $o_t$ çıkış kapısının temsil ederken, $h_t$ mevcut zaman adımının gizli durumunu, $W_o$ ağırlık matrisini, tanh hiperbolik tanjant aktivasyon fonksiyonunu, $C_t$ hücre durumu güncellemesini ve $b$ bias terimini temsil eder.

### 2.3.2. Tekrarlayan sinir ağı (RNN)

RNN, en temel tekrarlayan sinir ağı mimarisidir ve kısa vadeli bağımlılıkları öğrenme konusunda etkilidir. Bu yapıda, her zaman adımında, bir önceki gizli durum ile mevcut girdinin doğrusal bir kombinasyonu hesaplanır ve bu bilgi sonraki adıma aktarılır. Ancak RNN, uzun dizilerde bilgiyi taşımakta zorlanır; çünkü zamanla gradyanlar kaybolur. Bu mimari, düşük hesaplama maliyetiyle temel görevler için etkili bir çözüm sunar. Bununla birlikte, RNN'nin bu sınırlamaları, karmaşık zaman serisi analizlerinde dezavantaj oluşturabilir [8].




**Şekil 3.** RNN Mimarisi [19]

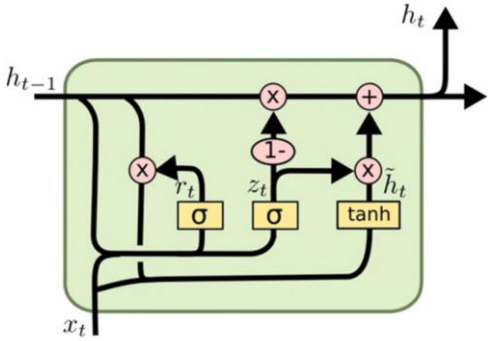
RNN'ler, her bir zaman adımında bir giriş vektörü $x_t$ alır ve bir önceki zaman adımının gizli durum vektörü $h_{t-1}$ ile birleştirerek yeni bir gizli durum vektörü $h_t$ üretir. Bu süreç, aşağıdaki formülle ifade edilir [19]:

$$h_t = \tanh(W_h \cdot h_{t-1} + W_x \cdot x_t + b) \quad (7)$$

burada $h_t$ unutma kapısının çıktısını temsil ederken, $W_h$ ağırlık matrisini, $h_{t-1}$ bir önceki zaman adımının gizli durumunu, $x_t$ mevcut zaman adımının girişini ve $b_f$ bias terimini temsil eder.

### 2.3.3. Kapılı tekrarlayan birim (GRU)

GRU, LSTM mimarisinin daha sade bir versiyonu olarak Cho ve arkadaşları tarafından geliştirilmiştir [21]. GRU, LSTM'e benzer şekilde giriş kapısı ve unutma kapısı içerir, ancak bu kapıları birleştirerek daha az parametre ile çalışır. GRU'nun avantajlarından biri, basit yapısı nedeniyle LSTM'e kıyasla daha az hesaplama yüküne sahip olmasıdır. Ayrıca, GRU LSTM ve diğer RNN modellerine kıyasla daha hızlı bir eğitim süreci sunmakta ve belirli uygulamalarda benzer doğruluk oranları sağlamaktadır. Bu özellikleri, kaynak kısıtlamalarının olduğu uygulamalarda GRU'nun tercih edilme nedenlerindendir [22].




**Şekil 4**. GRU Mimarisi [19]

GRU mimarisi, iki temel kapı (reset gate ve update gate) kullanır ve bu kapılar, bilgi akışını kontrol eder. Her bir zaman adımında, GRU hücresine bir giriş vektörü $x_t$ verilir ve bu hücre, bir önceki zaman adımının gizli durum vektörü $h_{t-1}$ ile birleştirilerek yeni bir gizli durum vektörü $h_t$ oluşturur. Bu işlem, reset kapısı denklem (8) ve update kapısı denklem (9) aracılığıyla kontrol edilir. Yeni gizli durum adayı denklem (10) formülüyle hesaplanır ve gizli durum güncellemesi denklem (11) şeklinde gerçekleştirilir [14].

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b) \quad (8)$$

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b) \quad (9)$$

burada $r_t$ reset kapısının çıktısını temsil ederken $z_t$ update kapısının çıktısını, σ sigmoid aktivasyon fonksiyonunu, $W_r$ ve $W_z$ ağırlık matrislerini, $h_{t-1}$ bir önceki zaman adımının gizli durumunu, $x_t$ mevcut zaman adımının girişini, $b$ bias terimini temsil etmektedir.

$$\widetilde{h_t} = \tanh\left(W_{\widetilde{h}} \cdot [h_{t-1}, x_t] + b\right) \quad (10)$$

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \widetilde{h_t} \quad (11)$$

burada $\widetilde{h_t}$ yeni gizli durum adayını temsil ederken $h_t$ son gizli durumu temsil eder. $W_{\widetilde{h}}$ ağırlık matrisini, tanh hiperbolik tanjant aktivasyon fonksiyonunu, $h_{t-1}$ bir önceki zaman adımının gizli durumunu, $x_t$ mevcut zaman adımının girişini ve $b$ bias terimini temsil etmektedir.

### 2.4. Değerlendirme metrikleri

Modellerin performans ve enerji tüketiminin birlikte ele alınması önemlidir. Çünkü kaynak kısıtlamalarının ve sürdürülebilirlik hedeflerinin ön planda olduğu taşınabilir cihazlar ve gömülü sistemlerde kritik bir öneme sahiptir. Deneyler, Windows 10 Pro [23] çalıştıran bir dizüstü bilgisayar kullanılarak çevrimdışı olarak gerçekleştirilmiştir. Enerji tüketim değerleri IPG aracı kullanılarak kaydedilmiştir. Önemli miktarda bilgi işlem kaynağına sahip dizüstü bilgisayar donanımı, 16 GB RAM, 256 GB disk depolama alanı ve dokuzuncu nesil Intel Core i7-9750H işlemciye (2,20 GHz) sahiptir. Toplam enerji tüketim verilerini kapsayan güç profilleri .csv biçiminde kaydedilmiş ve daha sonra tekrarlayan sinir ağı tabanlı modellerin performansını değerlendirmek için çevrimdışı olarak analiz edilmiştir.

### 2.4.1. Performans değerlendirmesi

Bu çalışmada modellerin performansı, Determinasyon Katsayısı ($R^2$ - R Squared), Ortalama Kare Hata-(MSE - Mean Squared Error) ve Ortalama Mutlak Hata (MAE – Mean Absolute Error) metrikleri kullanılarak değerlendirilmiştir. $R^2$, çıktıdaki varyansın ne kadarının girdi kaynaklı varyansla açıklandığını ölçen istatistiksel bir metrik olup, model doğrulamada önemli bir araçtır. $R^2$ değeri 1'e yaklaştıkça modelin veriyi daha iyi açıkladığı kabul edilir [24]. MSE, modelin tahmin ettiği değerler ile gerçek değerler arasındaki hataların karelerinin ortalamasıdır. MSE değeri büyüdükçe modelin tahminlerinde yaptığı hatalar da büyür. MAE ise tahmin edilen değerler ile gerçek değerler arasındaki hataların mutlak değerlerinin ortalamasıdır [25]. MAE değeri yükseldikçe modelin ortalama tahmin hatası da artar [26]. Performansı ölçme formülleri şu şekilde verilir:

$$R^2 = 1 - \left(\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2}\right) \quad (12)$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (13)$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \quad (14)$$

burada $\bar{y}$, bağımlı değişken $Y$'nin ortalama değeri; $\hat{y}$, $Y$'nin öngörülen değeri; $y$, $Y$'nin gerçek değeri ve $N$, veri örneklerinin sayısını temsil eder.

### 2.4.2. Enerji tüketim tahminlenmesi

Intel işlemcileri, Sandy Bridge nesliyle başlayarak, Çalışan Ortalama Güç Sınırı (Running Average Power Limit-RAPL) teknolojisini kullanmaya başlamıştır [27]. RAPL'nin öncelikli amacı güç sınırlaması olsa da enerji sayaçlarını da içermektedir. Ayrıca geniş kullanılabilirliği ve ek araçlara ihtiyaç duyulmaması nedeniyle güç ve enerji tahmini için yaygın olarak kullanılmaktadır [10]. IPG ise özellikle işlemci güç tüketimini analiz etmek ve işlemcilerin kötü amaçlı yazılım yürütme gibi farklı iş yükleri altında frekanslarını nasıl dinamik olarak ayarladığını anlamak isteyen kullanıcılar için değerli bir araçtır [28]. IPG'nin en önemli avantajlarından biri, enerji tüketimini gerçek zamanlı ve görsel olarak sunabilmesidir. Ayrıca bu araç, kullanıcıların kayıtlı verileri Excel biçiminde dışa aktarmalarına olanak tanır. Özellikle, bu yardımcı program dahili olarak enerji ölçümü için RAPL teknolojisini kullanmakta olup, saniyede 10 örnek frekansında güç tüketimi verilerini yakalayarak yüksek çözünürlüklü günlük kaydı gerçekleştirir [29].

Bu çalışmada, üç farklı RNN tabanlı mimarilerin tüketimi IPG sürüm 3.6 [9] kullanılarak tahmin edilmiştir. Bu mimarilerin enerji tüketimini tahminlemesi aşağıdaki verilen denklem ile bulunmuştur [30]:

$$E_c = P_c \times t \tag{15}$$

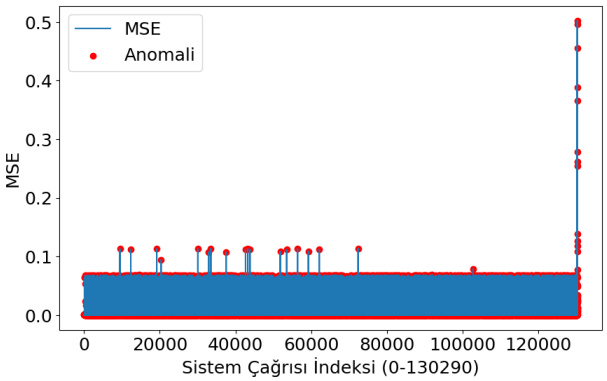
burada $E_c$ üç farklı RNN tabanlı mimarilerin algoritmalarını işlemek için tahmini enerji tüketimini temsil eder, $P_c$ algoritmalar tarafından ihtiyaç duyulan işlem gücünü belirtir ve $t$ bu algoritmaları işlemek için gereken zaman aralığını ifade eder.

Yürütme süresi, bir algoritma kullanarak bir problemi çözerken bir bilgisayarın hesaplamaları gerçekleştirmesi için gereken süreyi ifade eder. Deneyler sırasında, veri setindeki sistem çağrıları eğitim sürecindeki modellerin algoritmalarının yürütme süreleri ölçülmüştür. Kaydedilen sonuçlarda olası aykırı değerlerin etkisini en aza indirmek için her bir RNN mimarisi 10 kez yinelemeye tabi tutulmuştur. Hesaplamalar için watt cinsinden ölçülen ortalama güç tüketimi kullanılmıştır. Hesaplamalarda Watt (W) cinsinden ölçülen ortalama güç değeri esas alınmıştır. Toplam enerji tüketimi ise ortalama güç ile saniye (s) cinsinden ölçülen yürütme süresinin çarpımı yoluyla Joule (J) cinsinden belirlenmiştir.

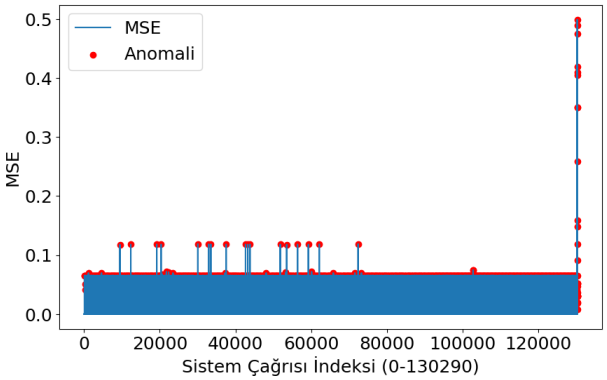
## 3. Bulgular

Bu çalışmada, RNN, GRU ve LSTM algoritmaları kullanılarak sistem çağrılarındaki anomalilerin tespit edilmesi ve bu işlemlerin enerji tüketimi açısından karşılaştırılması hedeflenmiştir. Her bir mimari, anomali tespit performansı ve enerji verimliliği açısından analiz edilmiştir. Yapılan analizler sonucunda, her bir modelin farklı başarı düzeyleri sergilediği gözlemlenmiştir. Ayrıca, bu modellerin anomali tespitinde etkinlikleri görselleştirilmiştir. Enerji tüketimlerini değerlendirmek için IPG aracı kullanılarak ölçümler gerçekleştirilmiştir. Bulgular, modellerin zaman serisi verilerinde farklı bağımlılık yapılarını öğrenme kabiliyetlerini ortaya koyarken, enerji tüketimindeki farklılıkları da açıkça göstermiştir.

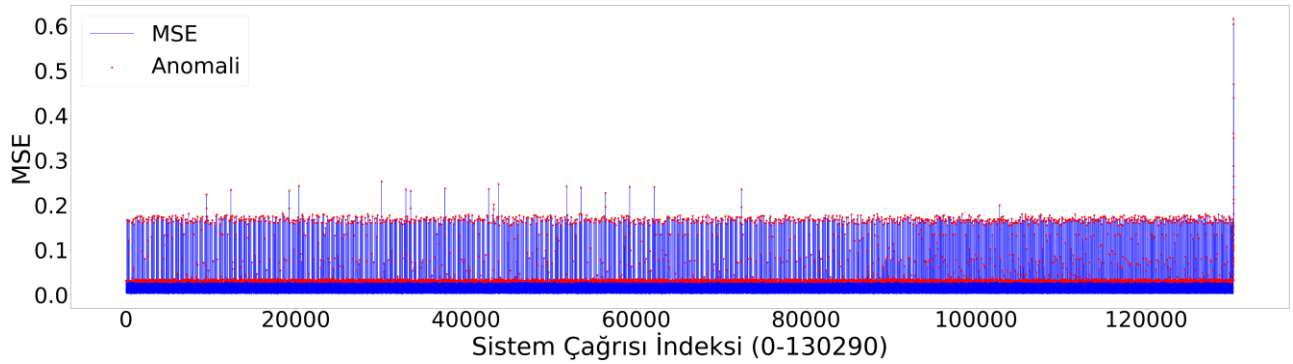
Üç farklı algoritma, test veri setinin üzerinde çalıştırıldıktan sonra elde edilen sonuçlar incelendiğinde, sistem çağrılarındaki anomalilerin tespitinde her bir algoritmanın farklı performans sergilediği bulgusu elde edilmiştir. Şekil 5, 6 ve 7, elde edilen anomalileri histogram üzerinde göstermektedir. Bu kapsamda, LSTM toplamda 6.509 anomali tespit etmiştir. GRU, 5.559 anomaliyi tanımlarken, RNN ise toplamda 3.358 anomali tespit etmiştir.




**Şekil 5.** LSTM modeli kullanılarak anomalilerin histogram üzerinde gösterimi

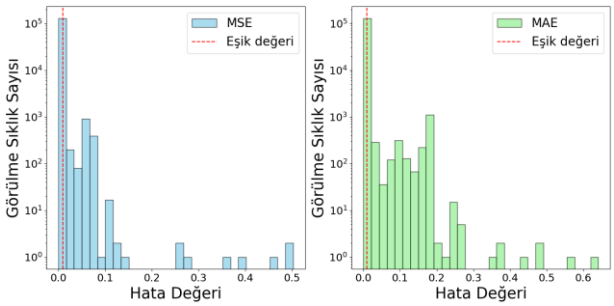



**Şekil 6.** RNN modeli kullanılarak anomalilerin histogram üzerinde gösterimi

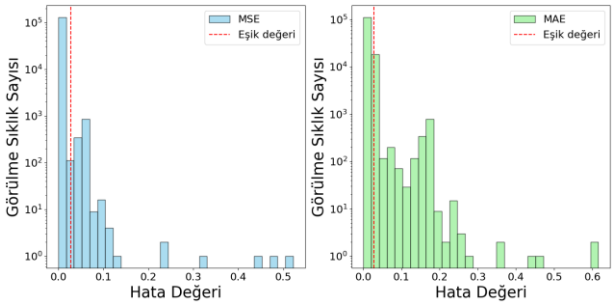



**Şekil 7.** GRU modeli kullanılarak anomalilerin histogram üzerinde gösterimi

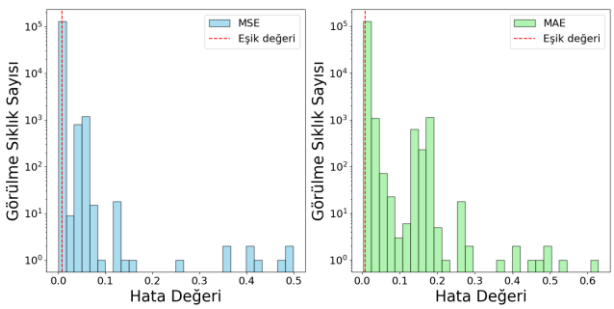
Tespit performansının daha ayrıntılı bir analizini yapmak için, MSE ve MAE histogramları her bir mimari için çıkarılmıştır. Şekil 8, 9 ve 10'da logaritmik ölçekte görülen MSE histogramları, hata değerlerinin çoğunluğunun belli bir eşik değerin altında yoğunlaştığını, daha yüksek değerlerin ise anomalileri temsil ettiğini göstermektedir.




**Şekil 8 .** LSTM modeline ait MAE ve MSE dağılımını gösteren performans histogramı




**Şekil 9.** GRU modeline ait MAE ve MSE dağılımını gösteren performans histogramı




**Şekil 10.** RNN modeline ait MAE ve MSE dağılımını gösteren performans histogramı

LSTM, GRU ve RNN modellerine kıyasla daha yüksek bir performans sergilemiştir. Gerçek anomalilerin tespitindeki hassasiyet ve hata dağılımlarının netliği, bu sonuçları desteklemektedir. Ayrıca, eşik değer bazlı sınıflandırma gibi ileri tekniklerin kullanılması, LSTM modelinin bu bağlamda doğruluğunu önemli ölçüde artırmıştır.

Tablo 1 RNN, GRU ve LSTM için $R^2$, MSE ve MAE değerlerini göstermektedir. LSTM için $R^2$ değeri 0.96, MSE değeri 3.18 ve MAE değeri 0.48; GRU için $R^2$ değeri 0.94, MSE değeri 3.21 ve MAE değeri 0.52; RNN için $R^2$ değeri 0.91, MSE değeri 3.96 ve MAE değeri 0.86 olarak elde edilmiştir. Bu 3 model arasında performans açısından en başarılı modelin LSTM olduğu tespit edilmiş, onu sırayla GRU ve RNN takip etmiştir.

**Tablo 1**. LSTM, GRU ve RNN modellerine ait performans karşılaştırılması.

| | *$R^2$* | *MSE* | *MAE* |
|---|---|---|---|
| RNN | 0.91 | 3.96 | 0.86 |
| GRU | 0.94 | 3.21 | 0.52 |
| LSTM | 0.96 | 3.18 | 0.48 |

Tablo 2 her model için enerji tüketim verilerini sunmaktadır. RNN, GRU ve LSTM için enerji tüketim değerleri sırasıyla 13.320,75 J, 20.934,47 J ve 27.984,75 J olarak ölçülmüştür. Tablo 2'de de gösterildiği üzere, bu modeller arasında en yüksek enerji verimliliğini RNN sağlamış; onu GRU ve LSTM takip etmiştir.

**Tablo 2**. RNN mimarisine dayalı üç farklı modelin enerji tüketim sonuçları (tüm deneyler 10 kez gerçekleştirilmiştir).

| | *Toplam Enerji Tüketimi (J)* | *Ortalama Güç (W)* | *Yürütme süresi (s)* |
|---|---|---|---|
| RNN | 13320.75 | 23.02 | 578.66 |
| GRU | 20934.47 | 24.10 | 868.65 |
| LSTM | 27984.75 | 23.80 | 1175.83 |

## 4. Tartışma ve Sonuç

Bu çalışma, LSTM, RNN ve GRU modellerinin sistem çağrısı verileri~~nde~~ üzerinde anomali tespiti performanslarını ve enerji verimliliklerini detaylı bir şekilde incelemiştir.

Çalışma sonuçlarına göre, RNN modeli tahminlerde daha yüksek hata üretmiş ve veri setindeki varyansı

açıklama oranı sınırlı kalmıştır. GRU, özellikle zaman serisi veya sıralı verilerde uzun vadeli bağımlılıkları daha iyi yakalayarak hatayı azaltmış ve RNN'ye kıyasla belirgin şekilde daha iyi sonuçlar elde etmiştir. LSTM modeli ise, hem MSE (3.18) hem de MAE (0.48) değerleri açısından en iyi performansı sergilemiştir. Ayrıca, $R^2$ katsayısının 0.96 olması, modelin veri setindeki varyansın büyük bir kısmını başarıyla yakaladığını göstermektedir.

Enerji verimliliği açısından LSTM, en yüksek enerji tüketimi değerine ulaşarak sistem çağrısı verilerinde anomali tespiti için enerji açısından verimli bir algoritma olmadığını ortaya koymuştur. Diğer taraftan RNN, enerji verimliliği açısından hem GRU hem de LSTM modellerine kıyasla daha iyi performans göstermiştir. Modellerin ortalama güç tüketimi sonucunda elde edilen nispeten küçük farklar, yürütme süresi ile toplam enerji tüketimi arasındaki güçlü korelasyonu vurgulamıştır. Bu doğrultuda, algoritmaların yürütme süresinin enerji tüketimleriyle doğru orantılı olduğu sonucuna varılmıştır.

Gelecek~~te~~ çalışmalar kapsamında, bu modellerin hibrit yaklaşımları veya dikkat mekanizmaları ile zenginleştirilmesi, büyük veri kümelerinde daha yüksek performans başarı ve verimlilik sağlayabilir. Ayrıca, karmaşık hesaplama görevlerini minimum yürütme süresi ile ele alabilen hibrit yaklaşımlar, enerji açısından daha verimli çözümler sunabilir.

## Etik Beyanı/Declaration of Ethical Code

*Bu çalışmada, "Yükseköğretim Kurumları Bilimsel Araştırma ve Yayın Etiği Yönergesi" kapsamında uyulması gerekli tüm kurallara uyulduğunu, bahsi geçen yönergenin "Bilimsel Araştırma ve Yayın Etiğine Aykırı Eylemler" başlığı altında belirtilen eylemlerden hiçbirinin gerçekleştirilmediğini taahhüt ederiz.*

## Kaynakça


[1] Fan, L., Wang, H., Zhao, Y., Xin, K. 2025. Application of Deep Learning in Public Network Security Management. Journal of Computer, Signal, and System Research, 2(1), 1-8.

[2] Nguyen, H. D., Tran, K. P., Thomassey, S., Hamad, M. 2021. Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management. International Journal of Information Management, 57, 102282.

[3] Tan, K., Zhan, D., Yu, Z., Ye, L., Zhang, H., Fang, B. 2024. Multi-Stage Defense: Enhancing Robustness in Sequence-Based Log Anomaly Detection. In ICC 2024-IEEE International Conference on Communications, 09-13 June, Denver, USA, 2725-2730.

[4] Tian, M., Verma, S., Gao, Y. 2025. Enhancing 3D seismic facies interpretation through a modified patched deep learning approach leveraging spatio-temporal dependencies. Computational Geosciences, 29(1), 8.

[5] Rodrigues, C. F., Riley, G., & Luján, M. 2018. SyNERGY: An energy measurement and prediction framework for Convolutional Neural Networks on Jetson TX1. In *Proceedings of the international conference on parallel and distributed processing techniques and applications (PDPTA)* (pp. 375-382). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp). 30 July-2 August, Las Vegas, USA, 375-382.

[6] Desislavov, R., Martínez-Plumed, F., & Hernández-Orallo, J. 2023. Trends in AI inference energy consumption: Beyond the performance-vs-parameter laws of deep learning. Sustainable Computing: Informatics and Systems, 38, 100857.

[7] Akkuzukaya, G., Yıldız, M. 2023. Time Series Anomaly Detection Embedded Systems By Using LSTM. International Journal of Multidisciplinary Studies and Innovative Technologies, 7(2), 90-96.

[8] Switrayana, I. N., Hammad, R., Irfan, P., Sujaka, T. T., Nasri, M. H. 2025. Comparative Analysis of Stock Price Prediction Using Deep Learning with Data Scaling Method. JTIM: Jurnal Teknologi Informasi dan Multimedia, 7(1), 78-90.

[9] Rodriguez, C., Degioanni, L., Kameni, L., Vidal, R., Neglia, G. 2024. Evaluating the energy consumption of machine learning: Systematic literature review and experiments. arXiv preprint arXiv:2408.15128.

[10] Fahad, M., Shahid, A., Manumachu, R. R., Lastovetsky, A. 2019. A comparative study of methods for measurement of energy of computing. Energies, 12(11), 2204.

[11] Intel® Power Gadget, 2023. https://archive.org/details/intel-power-gadget-windows (Erişim Tarihi: 21.01.2025).

[12] Duan, G., Fu, Y., Cai, M., Chen, H., & Sun, J. (2023). DongTing: a large-scale dataset for anomaly detection of the Linux kernel. *Journal of Systems and Software*, *203*, 111745.

[13] Zhang, T., Dipanzan, S. I., Tahmooresnejad, L., Ezzati-Jivan, N. 2024. Assessing Predictive Models for Energy Consumption Across Varied Software Environments. In *2024 IEEE*

*International Conference on Big Data (BigData)*, 15-18 December, Washington, 5233-5242.

[14] Asghar, R., Fulginei, F. R., Quercio, M., Mahrouch, A. 2024. Artificial neural networks for photovoltaic power forecasting: a review of five promising models. IEEE Access.

[15] O. Ezeme, Q. Mahmoud, A. Azim, and M. Lescisin. Syscall veri seti: Sistem Çağrıları kullanarak bağlam modelleme ve anomali tespiti için bir veri seti. Mendeley Data, V2, 2019. https://data.mendeley.com/datasets/vfvw7g8s8h/2 (Erişim Tarihi: 01.01.2025).

[16] Memory, L. S. T. 1997. Sepp hochreiter and jürgen schmidhuber. Neural Computation, 9(8), 1735.

[17] Aggarwal, S. 2023. LSTM based Anomaly Detection in Time Series for United States exports and imports. Munich Personal RePEc Archive ( MPRA), 117149.

[18] Puteri, D. I. 2023. Implementasi Long Short Term Memory (LSTM) dan Bidirectional Long Short Term Memory (BiLSTM) Dalam Prediksi Harga Saham Syariah. Euler J. Ilm. Mat. Sains dan Teknol, 11(1), 35-43.

[19] Golshanrad, P., Faghih, F. 2024. DeepCover: Advancing RNN test coverage and online error prediction using state machine extraction. Journal of Systems and Software, 211, 111987.

[20] Siami-Namini, S., Tavakoli, N., Namin, A. S. (2019, December). The performance of LSTM and BiLSTM in forecasting time series. In 2019 IEEE International conference on big data (Big Data), 09-12 December, Los Angeles, USA, 3285-3292.

[21] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

[22] Mateus, B. C., Mendes, M., Farinha, J. T., Assis, R., Cardoso, A. M. 2021. Comparing LSTM and GRU models to predict the condition of a pulp paper press. Energies, 14(21), 6958.

[23] Microsoft, 2025. https://www.microsoft.com/tr-tr/software-download/windows10 (Erişim Tarihi: 01.06.2025).

[24] Aswal, V. S., Singh, B. K., Maheshwari, R. 2025. Machine learning-based model for prediction of concrete strength. Multiscale and Multidisciplinary Modeling, Experiments and Design, 8(1), 48.

[25] Suresh, B. S., Suresh, M. 2024. AI Based Retail Sales Management: Leveraging Optimized Metaheuristic Algorithms for Forecasting and Recommendations. In 2024 Second International Conference on Networks, Multimedia and Information Technology (NMITCON), 09-10 August, Bengaluru, India, 1-5.

[26] Adeogun, O. Y., Adeoti, L., Allo, O. 2024. Machine Learning Approach in Predicting Water Saturation Using Well Data at "Tm" Niger Delta. Available at SSRN 4927838.

[27] Smejkal, T., Hähnel, M., Ilsche, T., Roitzsch, M., Nagel, W. E., Härtig, H. 2017. Practical Energy Accounting for {Multi-Core} Systems. In 2017 USENIX Annual Technical Conference (USENIX ATC 17), 12–14 July, Santa Clara, USA, 589-601.

[28] Buestán-Andrade, P. A., Peñacoba-Yagüe, M., Sierra-García, J. E., Santos, M. 2024. Wind power forecasting with machine learning algorithms in low-cost devices. Electronics, 13(8), 1541.

[29] PranavRaj, R., Syam, S. R. 2024. Identification Of Malware Families Using Energy Consumption. In 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), 24-28 June, Kamand, India, 1-6.

[30] Tekin, N., Acar, A., Aris, A., Uluagac, A. S., & Gungor, V. C. 2023. Energy consumption of on-device machine learning models for IoT intrusion detection. Internet of Things, 21, 100670.