International Journal of Multidisciplinary Studies and Innovative Technologies

e-ISSN: 2602-4888
dergipark.org.tr/en/pub/ijmsit
Research Article

2023, 7 (2), 90-96
DOI: 10.36287/ijmsit.7.2.90
Received: November 21, 2023; Accepted: December 22, 2023

Gömülü Sistemlerde LSTM Kullanımı ile Zaman Serisi Anormallik Tespiti

Gülsüm Akkuzu Kaya[1*], Mehmet Yıldız [2]

[1*]*Kırşehir Ahi Evran Üniversitesi, Mühendislik ve Mimarlık Fakültesi,Bilgisayar Mühendisliği, Kırşehir, Türkiye (gulsum.akkuzukaya@ahievran.edu.tr) (ORCID: 0000-0003-1806-7759)*
[2] *Kırşehir Ahi Evran Üniversitesi, Mühendislik ve Mimarlık Fakültesi,Bilgisayar Mühendisliği, Kırşehir, Türkiye (mehmet.yildiz.jap@gmail.com)*

Özet **–**İnsansız Hava Araçları (İHA) için anomali tespiti önemli bir araştırma alanı olmuştur. Anormallikleri tespit etme tekniklerinden biri, geleneksel Makine Öğrenimi (ML) algoritmalarını uygulamaktır, ancak geleneksel ML yaklaşımları, özellikle uzun vadeli bağımlı noktalardaki anormallikleri tespit edemez. Bu çalışma, İHA sistem çağrılarının zaman serisindeki anormallikleri tespit etmek için Uzun Kısa Süreli Bellek (LSTM) yöntemini kullanır. Bunu yapmak için, LSTM ağı, bir İHA sistemindeki olayların zaman aralıklarındaki verilerin uzun vadeli bağımlılıklarını öğrenmek için birbiriyle çalışan birden fazla LSTM hücresinden oluşur. Bu makalede kullanılan veri seti, sistem çağrılarının sırasını ve türünü, sistem çağrısı olaylarının zaman damgalarını, işlem kimliklerini ve isteğe bağlı argümanları içeren bir İHA'dan sistem çağrısı olaylarından toplanmıştır. LSTM tekniği ile derinlemesine modern bir siber tehdit analizi sağlamayı amaçladığımız için veri seti bu çalışmanın amacına uygun bir veri setidir. Deneysel sonuçlar, LSTM tekniğinin sistem çağrılarının zaman serisindeki anormallikleri tespit etmedeki üstün performansını kanıtlıyor.

Anahtar Kelimeler *–LSTM, Anormallik tespiti, İnsansız hava aracı, Derin öğrenme, Sistem çağrısı*

Atıf: Akkuzu Kaya, G., Yıldız, M. (2023). Gömülü Sistemlerde LSTM Kullanımı ile Zaman Serisi Anormallik Tespiti Journal of Multidisciplinary Studies and Innovative Technologies, 7(2): 90-96.

Time Series Anomaly Detection Embedded Systems By Using LSTM

Extended Abstract

Anomaly detection for Unmanned Aerial Vehicles (UAVs) has been an important research area. One of the techniques to detect anomalies is to apply traditional Machine Learning (ML) algorithms however traditional ML approaches could not detect anomalies especially long-term dependent points. This study uses the Long Short-Term Memory (LSTM) method to detect anomalies on time series of UAV systemcalls. To do so, the LSTM network is comprised of multiple LSTM cells that work with each other to learn the long-term dependencies of the data in the timestamps of events in a UAV system. The dataset used in this paper, systemcall events from a UAV which includes the order and type of system calls, timestamps of the system call events, process IDs and optional arguments. The dataset is a suitable for the aim of this study as we aim to provide a depth modern cyber threat analysis with LSTM technique. The experimental results prove the superior performance of LSTM technique to detect anomalies on time series of system calls.

Keywords *– LSTM, Anomaly detection, UAV, System calls, Deep learning*

Citation: Akkuzu Kaya, G., Yıldız, M. (2023). Time Series Anomaly Detection Embedded Systems By Using LSTM, Journal of Multidisciplinary Studies and Innovative Technologies, 7(2): 90-96.

I. GİRİŞ

Siber tehditler ve kötü niyetli saldırılar son yıllarda bilgi teknolojileri şirketleri başta olmak üzere, finans, enerji, havacılık ve sağlık sektörlerine kadar çok sayıda alana yönelik büyük ölçüde arttmıştır. Bu sistemler, bilinen ve keşfedilmemiş birçok saldırılara karşı oldukça hassastır. Bu sistemlerden en güvenlisi havacılık sistemleri olarak bilinir.

Havacılık sistemleri en güvenli ve minimum yazılımsal saldırı yapılabilecek sistemler iken günümüzde işlevselliği

arttırarak maliyeti düşürmek için gerçekleştirilen işlemler hava araçlarının sistemlerini de yazılımsal saldırılara savunmasız hale getirmiştir. Bir hava aracının kapasitesini hava durumu verisi güncellemesi için arttırmak veya kritik platformlar üzerindeki fonksiyonelliği artırmak gibi işlemler saldırıya sebep olan işlemlere örnek gösterilebilir [1]. Gelişmiş saldırı teknikleri ile en güvenli domain alanına sahip havacılık sistemlerinin de güvenliğinin tehdit altında olduğu yapılan araştırmalar ile gösterilmiştir [2, 3]. Saldırılar veya sapmalar anormallik ya da izinsiz giriş (intrusion) olarak sınıflandırılmıştır [4]. Anormallik tespiti ile izinsiz giriş birbirlerinden farklı yapılar ve tekniklerdir. İzinsiz giriş tespit teknikleri, keşfedilen anormalliklerin imzalarını saklayarak kötüye kullanım davranışını tespit etmek için imzaların kullanımına dayanırken, anormallik tespit modelleri, bilinen standart özellikleri kullanarak profiller oluşturur ve oluşturulan profillerden herhangi bir sapmayı anormal olarak etiketler [5]. İmza-tabanlı yaklaşımların en açık ve belirgin problemi yalnızca gözlemlenmiş izlerin tespitinin gerçekleştirilmesidir. Her iki sınıf problemleri için uygulanan aykırı değer tespit teknikleri, beklenen davranışlara uyum sağlamayan kalıpları keşfetme problemini çözmeye çalışır. Gömülü sistemlerdeki en yaygın olarak yapılan saldırılar, ağ tabanlı (NIDS) ve ana bilgisayar tabanlı (HIDS) saldırılar olmak üzere ikiye ayrılır. Ağ saldırı sistemleri (NIDS), gelen ağ trafiğini analiz eden sistemleri içerirken ana bilgisayar tabanlı saldırı (HIDS) sistemleri işletim sistemleri dosyalarını izleyen sistemleri içerir.

Bu tespit sistemleri, normal sistem davranışlarından modeller öğrenerek tehdit oluşturabilecek davranışları algılayabilirler. Ana bilgisayar tabanlı sistemlerde, sistem çağrısı uygulamaların işletim sistemine girmesinin tek yoludur. Bu sebeple, HIDS'te yeni saldırıların tespit edilmesi için geliştirilecek modelleri eğitmek için sistem çağrısı dizilerini içeren farklı veri kümeleri kullanılır [6]. Örneğin DARPA, ağ tabanlı saldırı tespiti için kullanılan yaygın veri kümelerinden biridir [7]. Diğer gömülü sistem çağrıları verisi içeren erişime açık veri setleri NGIDS-DS [8], ADFA-LD[9], NSL-KDD ve KDD99 [10], CIC-IDS 2018 [11] ve PLAID [12] olarak sıralanabilir. Bu veri kümeleri, gömülü sistem verisi içeren erişime açık veri kümeleri oldukları için birçok araştırmada kullanılmışlardır ve bu veri kümelerinin ortak özellği ağ tabanlı gömülü sistemlere ait veri kümleri olmalarıdır. Bu çalışmada, ana bilgisayar tabanlı (host-based) Ezeme ve arkadaşları tarafından toplanan hava aracına ait sistem çağrıları, zaman bilgisi ve sistem çağrılarına ait farklı özellikleri içeren veri seti kullanılmıştır [13].

Ezeme ve arkadaşları, anormallik tespit tekniklerinden kümelenme tekniğini insansız hava aracı gömülü sistem çekirdek durumlarındaki anormallik tespiti için kullanmışlardır [14]. Kümelemeye dayalı tekniklerin ana fikri, verilen verilerden aykırı değerleri tespit etmek için standart kümeleme tekniklerinin uygulanmasıdır. Aykırı değerler, herhangi bir büyük veya yoğun kümenin içinde veya yakınında olmayan gözlemler olarak kabul edilir [15].

II. MATERYAL VE METOT

A. LSTM

LSTM (Long Short-Term Memory), tekrarlayan sinir ağlarının bir uzantısı ve uzun kısa süreli hafıza olarak kabul edilir. Tekrarlayan sinir ağları, önceki bilgilerin belirli bir noktada mevcut iş için kullanılmasına olanak sağlayan kısa süreli bellek yeteneğine sahiptirler. Tekrarlayan sinir ağlarından genişletilmiş LSTM mimarisi, mevcut sinir düğümü için önceki bütün bilgilerin bir listesinin mevcut olduğu "uzun süreli bellek" yeteneğine sahiptir. En yaygın olan LSTM mimarisi; bir hücre, giriş kapısı, bir çıkış kapısı ve bir unutma kapısından oluşur.

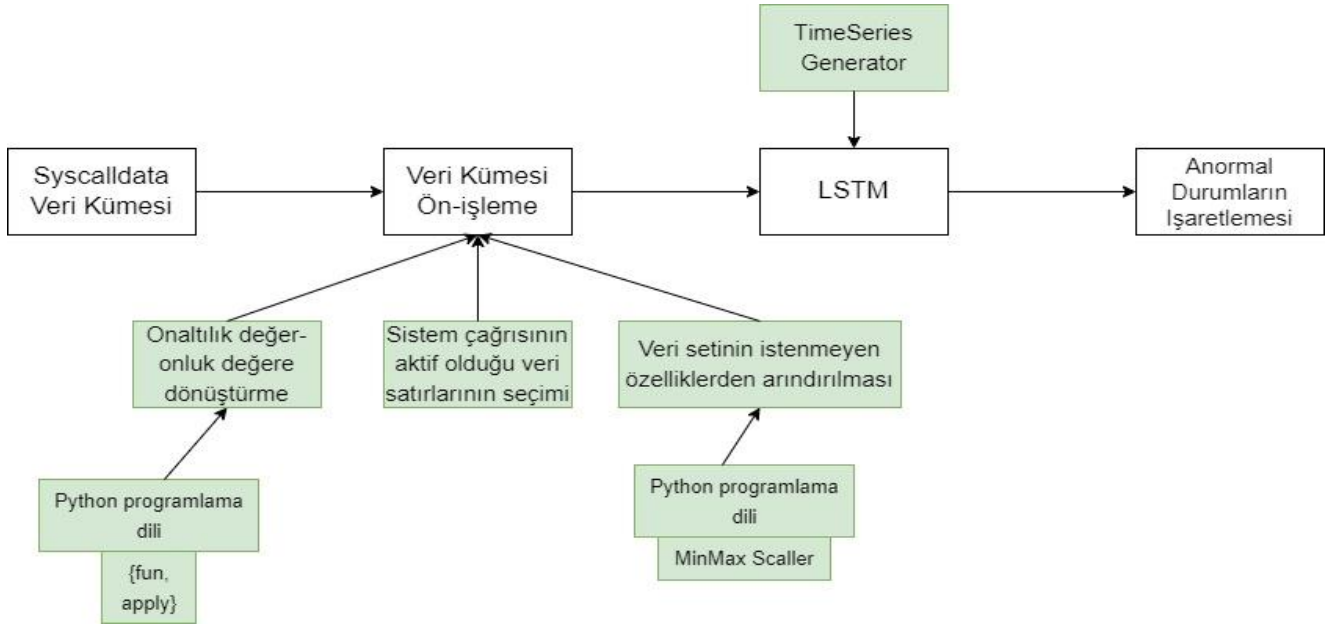
Forget gate olarak bilinen unutma kapısının amacı; hem önceki gizli katman verileri hem de yeni girdi verileri ele alındığında, hangi bitlerin verilerinin yararlı olduğuna karar vermektir. Eğer veri faydalı ise; sigmoid fonksiyonu kullanarak 1 e yakın bir değer üretir, aksi halde ise 0 ayakın değer üretir.

Giriş kapısı, iki katlı amaca sahiptir. Birincisi, yeni veri veya önceki gizli katman durum verisinin hücre durumunda tutulup tutulmayacağının değer olup olmadığını kontrol eder. Eğer kayda değer veri var ise; hangi yeni verilerin ekleneceğine de ikinci amaç hizmet eder.

Çıkış kapısının temel amacı ise, yeni gizli duruma ihtiyaç olup olmadığına karar vermektir. (Her bir kapı için kullanılan matematiksel formüller için [20]).

B. Metot

Şekil 1'de bu çalışmanın tamamlanması için izlenen yol ve adımlar verilmiştir. Veri kümesi üzerinde analiz işlemi gerçekleştirmek için orjinal veri setinde; veri setinin model eğitimi ve anormallik tespiti için uygun hale getirilmesi amacıyla çeşitli düzenlemeler ve değişiklikler yapıldı. Veri seti üzerinde hexadecimal (onaltılık tabandaki) değerler decimal (onluk taban) değerlere dönüştürüldü. Bunu sağlamak için "fun" yardımcı işlev tanımlandı ve apply yöntemi kullanılarak SystemCallID sütunu üzerine uygulandı. Böylece, SystemCallID sütunu orjinal veri setindeki onaltılık sistem temsiliyle tutulan değerlerden ondalık sistem temsiline dönüştürüldü. Veri setindeki zaman farklarını hesaplama ve yeni bir sütun eklemek için "Timestamp" sütunu kullanıldı. Zaman farkı hesaplamaları için diff() fonksiyonu kullanıldı ve elde edilen farklar "TimeDiff" adlı yeni bir sütunda saklandı. Veri setinin ölçeklendirilmesi için MinMaxScaler kullanıldı. TimeDiff sütunu ölçeklendirilerek "ScaledTimeDiff" adlı yeni bir sütun oluşturuldu. Ölçeklendirme işlemi, verilerin farklı ölçeklerde olmamasını ve modelin daha iyi performans göstermesini sağlamak için oluşturuldu.



Şekil 1. Metot Akış diyagramı (Flowchart)

C. *Veri Seti*

Page numbers, headers and footers must not be used.

Bu çalışmada kullanılan veri seti Ezeme ve arkadaşları tarafından bir insanısız hava aracı sistemi simüle edilerek toplanmıştır [21]. Veri setindeki verilere ilişkin detayların örnekleri şöyledir;

Normal Profil: Bu profildeki (moddaki) veri örnekleri elde edilirken İHA uygulaması, herhangi bir dahili veya harici enjeksiyon veya kesinti olmaksızın çalıştırılmıştır.

Delay profil: Bu profildeki veri örnekleri, denetleyicinin zaman zaman uygulamanın parametrelerini algılamada gecikmeye zorlamasıyla ve hesaplama açısından pahalı işlemler oluşturulmasıyla elde edilmektedir. Bu pahalı sapma hesaplamaları, İHA sensörlerinin varsayıldığı gibi sorgulanmamasına neden olur. Bu nedenle, İHA hedef, seyir ve irtifa ile ilgili parametreleri ayarlamak için mücadele ederken kararsızlık yaşamaktadır. Bu görevler, oluşturulan sistem çağrılarının türünü ve sırasını etkiler, çünkü yürütülmesi, Normal profili takip edebilen veya etmeyebilen bilinen veya bilinmeyen sistem çağrılarının oluşturulmasına yol açar. Bu profil, normal durum geri yüklenemediğinde İHA'nın çökmesine neden olduğu için gizli olmayan bir saldırı örneğidir. Bu profildeki veri örnekleri, Pseudo-Random profilindeki örneklerden daha az karmaşıktır, çünkü gizli olmayan yapıya sahiptirler.

Pseudo-Random Profile: Bu profildeki veri örnekleri, saldırı senaryolarının karmaşıklığı arttırılarak ve gizli operasyona dayalı bir anomali oluşturularak, sözde rastgele aralıklarla İHA'nın durumlarını ve parametrelerini bir UDP soketi aracılığıyla sızdıran bir işlem ile elde edilmektedir. Veri örnekleri elde edilirken, İHA'yı çökertmeyen, ancak başka amaçlar için etkinliğini izleyen gizli operasyon taklit edilmektedir. Böylece İHA uygulamasını izleme sürecinde, sistem çağrılarının sırası, türü ve argüman yapısı açısından farklı olabilecek bazı sistem çağrıları üretilir. Delay profilinden farklı olarak, bu profil gizli modda çalışır ve İHA kontrol uygulamasının çökmesine yol açmaz.

D. *Veri Ön İşleme*

Şekil 2'de normal profile ait ham verilere ait ilk satırlar örnek olarak gösterilmektedir. Bu şekilde, ilk sütunda zaman damgası, ikinci sütunda işlemin türü, üçüncü sütunda işlemin ID'si ve diğer sütunlarda opsiyonel argümanlar temsil edilmektedir.

```
455697, SYSCALL,0x53,0x7ffd4d461f26,0x1ff,0x0,0x8080808080808080L,0x0,0x0,0x1f861000
455714, SYSRET,0x1f861000
455714, SYSCALL,0xe7,0x0,0x0,0x7ffd4d461a88,0x8080808080808080L,0x0,0x0,0x1f861000
455724, SYSCALL,0xf,0x11,0x7ffc2f478c70,0x7ffc2f478b40,0x0,0x0,0x0,0x1d491000
455724, SYSRET,0x1d491000
455724, SYSCALL,0x10,0xa,0x5410,0x7ffc2f4790bc,0x0,0x0,0x0,0x1d491000
455724, SYSRET,0x1d491000
```

Şekil 2. Normal profilin ham verilerine ait ilk satırlar

Veri ön işleme aşamasında, normal, delay ve pseudo-random örneklerinin ham günlük dosyaları işlenmiştir. Veri ön işleme aşamasında ilk olarak, SYSRET klasörünü temsil eden satırlar silinmiştir. Daha sonra, zaman damgaları tek kalacak şekilde, diğer sütundaki bilgiler silinmiştir. Şekil 3'te veri kümesinin ön işleme aşamasından sonra kullanılan bir örneği verilmiştir

	Timestamp	Type	SystemCallID	TimeDiff
0	455697	SYSCALL	83	0.0
2	455714	SYSCALL	231	17.0
3	455724	SYSCALL	15	10.0
5	455724	SYSCALL	16	0.0
7	455725	SYSCALL	4	1.0
9	455726	SYSCALL	4	1.0
11	455726	SYSCALL	4	0.0
13	455728	SYSCALL	4	2.0
15	455728	SYSCALL	14	0.0
17	455728	SYSCALL	57	0.0

Şekil 3. Veri Kümesi Ön İşleme Sonrası Örneği

Şekil 4'de kullanılan LSTM (Long short-term memory) metodunun eğitim tablusonun bir örneği verilmiştir. LSTM

metodu bilgileri uzun süre hatırlama yeteneğinden dolayı uzun vadeli bağımlılıklı problemleri çözmede oldukça başarılı bir tekniktir. Bu sebeple, LSTM birçok zaman analizli çalışmalarda yaygın olarak uygulanmış ve oldukça yüksek başarı elde etmiştir [16-18]. Şekil 4' de verilen LSTM modeli için;

Python keras kütüphanesi kullanılarak Sequential modeli kullanımı

LSTM katmanı 128 hücre içermektedir

İnput_shape parametresi modelin girdi boyutunu belirtir. Lookback kısmı modelin geçmiş verilere bakma süresini belirten kısımdır

Dropout katmanı modelin ikinci katmanıdır. Dropout, eğitim sırasında rastgele bir kısım nöronları etkisiz hale getirerek aşırı uyum (overfitting) sorununu azaltmaya yardımcı olur. Dropout katmanının rate parametresi, etkisiz hale getirilecek nöron oranını belirtir ve burada %20 olarak ayarlanmıştır.

Modelin üçüncü katmanı RepeatVector katmanıdır. Bu katman, önceki LSTM katmanının çıktılarını bir zaman dizisi olarak tekrarlar ve giriş boyutunu (None, 1, 128) olarak değiştirir.

Dördüncü katmanda, retur_sequences=True olarak ayarlanarak bu katmanın çıktısının bir zaman dizisi olarak dönmesi sağlanmıştır.

Beşinci katman yine bir Dropout katmanıdır ve parametreleri ilk Dropout katmanıyla aynıdır. Altıncı ve son katman TimeDistributed katmanıdır. Bu katman, tam bağlı bir (dense) katmandır ve çıktı boyutunu (None, 1, 1) olarak değiştirir. Bu, her zaman adımı için ayrı bir çıktı değeri elde etmemizi sağlar.

Modelin kaybını ve optimize ediciyi belirlemek için compile yöntemi kullanıldı. Burada kayıp fonksiyonu olarak ortalama karesel hata (mean squared error - MSE) kullanıldı.

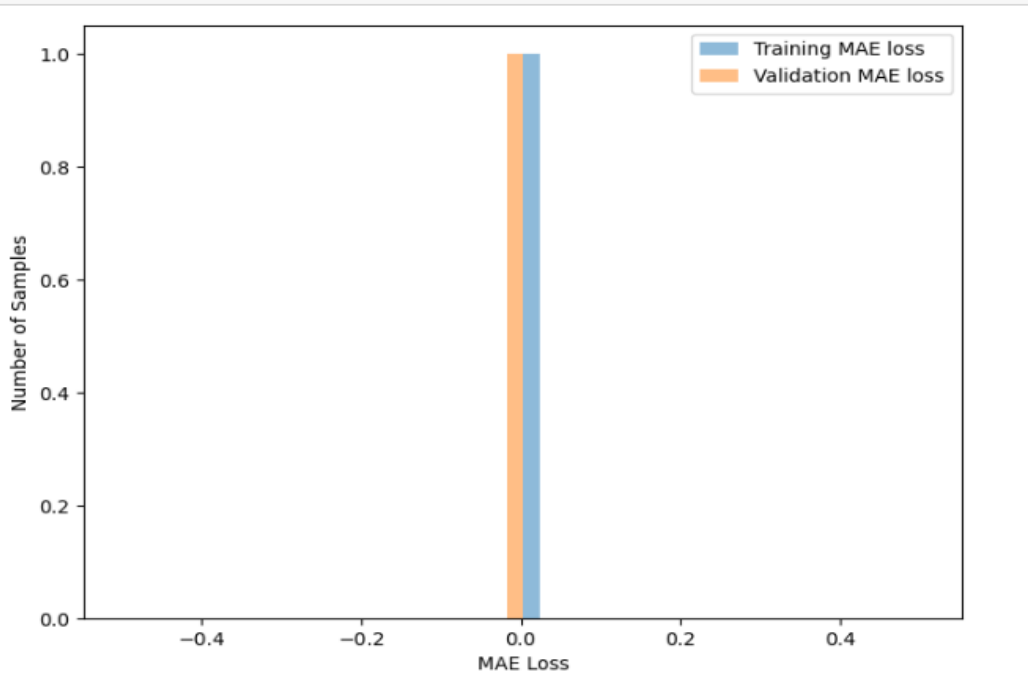
Model, fit yöntemi kullanılarak eğitildi: train_gen ve val_gen veri üreteçleri, eğitim ve doğrulama verilerini sağlamak için kullanıldı. epochs parametresi 10 olarak ayarlanmıştır. batch_size parametresi burada 32 olarak ayarlanmıştır. callbacks parametresi, eğitim sırasında kullanılan geri çağrı işlevlerini belirtir ve bu çalışmada EarlyStopping geri çağrısı kullanıldı. Bu geri çağrı, val_loss değerinin 3 dönem boyunca iyileşmediği durumlarda eğitimi durdurur ve aşırı uyumu önlemeye yardımcı olur. shuffle parametresi, veri setinin her dönemde karıştırılıp karıştırılmayacağını belirtir, burada False olarak ayarlanmıştır, yani veri seti her dönemde karıştırılmayacaktır. Eğitim sırasında her dönemdeki kayıp (loss) ve doğrulama kaybı (val_loss) değerleri çıktı olarak görüntülenir. Model eğitim aşaması toplamda 10 dönem aşamasından oluşmaktadır. Her bir dönem için eğitim kaybı, doğrulama kaybı ve geçen süre görüntülendi. Çünkü bu değerler; modelin yapısı, katmanaların çıktı şekilllerini ve eğitim sürecindeki kayıp değerlerini gösterir. Şekilde yer alan "Param #" ifadesi, her bir katmandaki toplam parametre sayısını temsil eder. Parametreler, modelin eğitim sırasında öğrenmesi gereken ağırlıkları ve bias değerlerini ifade eder. Daha fazla parametre genellikle daha karmaşık bir modelin olduğunu gösterir. "Total params" ifadesi, tüm katmanlardaki toplam parametre sayısını gösterir. Bu, modelin toplamda kaç parametreye sahip olduğunu belirtir. "Trainable params" ifadesi, eğitilebilir (trainable) parametrelerin sayısını gösterir. Bu, modelin eğitim sırasında optimize edilecek ve güncellenecek parametrelerdir. "Non-trainable params" ifadesi ise eğitilmeyen (non-trainable) parametrelerin sayısını gösterir. Bu, önceden eğitilmiş veya dondurulmuş ağırlıklar gibi, eğitim sırasında güncellenmeyecek parametrelerdir. Bu durumda, 0 olduğunu görüyoruz, yani modelde eğitilmeyen parametre bulunmamaktadır. Bu parametreler, modelin karmaşıklığını ve eğitilebilir ağırlıkların sayısını gösterir. Daha fazla parametre genellikle daha fazla esneklik sağlar, ancak aynı zamanda daha fazla veri ve hesaplama gücü gerektirebilir.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 128)               66560

 dropout (Dropout)           (None, 128)               0

 repeat_vector (RepeatVector  (None, 1, 128)           0
 )

 lstm_1 (LSTM)               (None, 1, 128)            131584

 dropout_1 (Dropout)         (None, 1, 128)            0

 time_distributed (TimeDistr  (None, 1, 1)             129
 ibuted)

=================================================================
Total params: 198,273
Trainable params: 198,273
Non-trainable params: 0
```

Şekil 4. LSTM Eğitim Örneği

III. DENEYSEL SONUÇLAR

Şekil 5, eğitim ve doğrulama veri kümeleri için Ortalama Mutlak Hata (MAE) kayıplarını histogram olarak görselleştirilmesidir. Bu şekilde, MAE kaybının dağılımını ve örnek sayısını gösterme amaçlanmıştır. Verilen histogram grafiklerinden biri eğitim veri kümesi MAE kayıplarını, diğeri doğrulama veri kümesi MAE kayıplarını temsil etmektedir. Şekilde verilen çıktı; eğitim ve doğrulama veri kümelerindeki MAE kaybının dağılımını karşılaştırmak ve birbirleriyle ilişkilendirmek için kullanılabilir. Örneğin, eğitim ve doğrulama MAE kayıpları arasındaki benzerlikleri veya farklılıkları görmek gibi işlem için kullanılabilir.



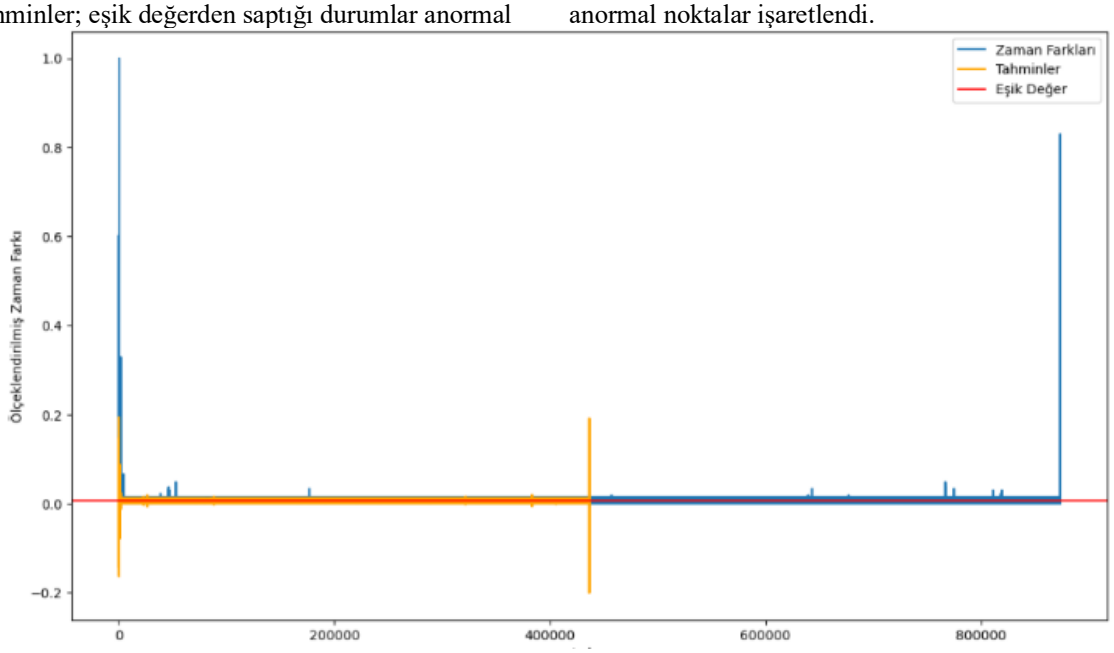
Şekil 5. LSTM MSE değerleri

Anormallik tespiti için, modelin tahminleri ile gerçek değerler arasındaki hata metriklerini kullanarak eşik değer çıkarıldı. Bunun için izlenen adımlar şöyedir;

- Veri setindeki tüm zaman adımlarınıkapsayan girdi ve hedef zaman serisi için TimeseriesGenerator oluşturuldu
- Veri kümesindeki veriler için tahminleme işlemi gerçekleştildi

Eşik değer, veri setindeki zaman serisinin normal veya anormal olup olmadığını belirlemek için kullanıldı. Veri setindeki tahminler; eşik değerden saptığı durumlar anormal noktalar dğier surumlar ise normal durumlar olarak kabul edildi.

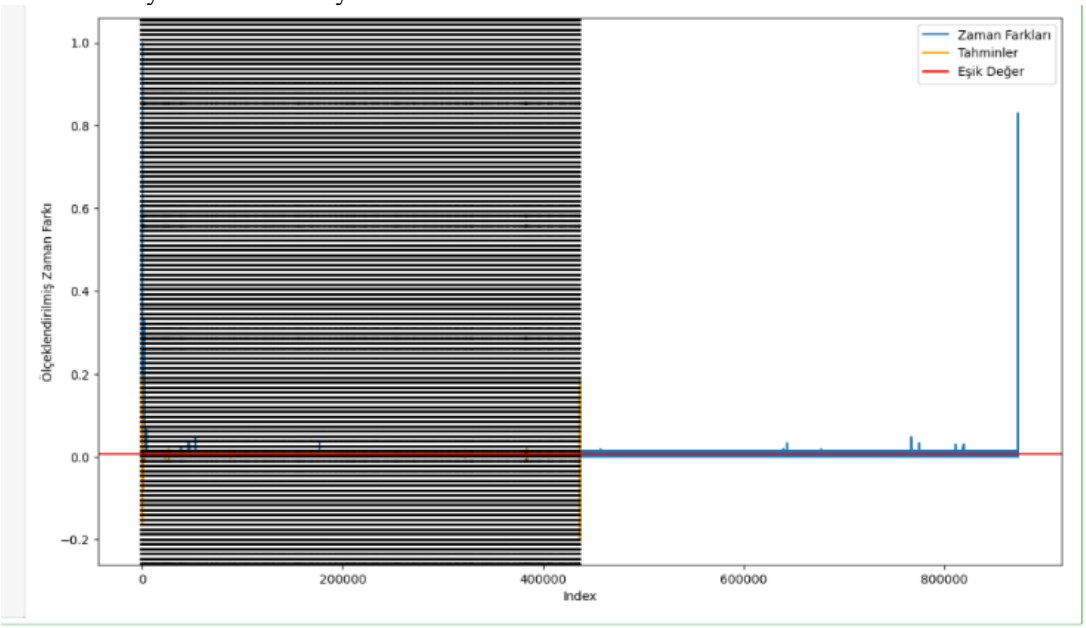
Şekil 6'da eşik değer kırmız çizgi ile gösterilmiştir. Tahmin edilen değerler ve gerçek değerler arasındaki sapma miktarı bu şekil ile açıkça görülebilir. Eşik değer bize; tahmin edilen verilerin gerçek değerlerden ne kadar uzakta olduğunu gösterir. Ayrıca, tahmin edilen değerlerin kabul edilebilirlik sınır değerleri eşik değer ile belirlenebilir. Veri setindeki değerler, eşik değer altında veya eşik değer üstünde ise anormal noktalar işaretlendi.



Şekil 6. Eşik değer ve veri kümesindeki dağılımı

Veri kümesi üzerinde LSTM metodu uygulandıktan sonra anormal noktaların gösterimi işlemi Şekil 7'te gösterilmiştir. Zaman farkları (gerçek değerler) mavi renkli çizgi ve tahmin edilen değerler sarı renkli çizgi ile gösterilmiştir. Eşik değer yatay bir çizgi ile gösterilerek, kabul edilebilir stahminleme sınırını temsil etmektedir. Anormallikleri göstermek için her bir anormalliğe karşılık gelen zaman adımlarında dikey çizgiler çizilmiştir. Bu çizgiler, tahminlerin eşik değerini aştığı veya altına düştüğü noktaları göstermektedir. Şekil üzerindeki dikey olarak görülen çizgiler, anormalliğin net bir şekilde tespit edildiği noktaları gösterirler. Sistem çağrıları arasındaki zaman farklarının beklenen desenlerden sapması durumunda

ortaya çıkan noktalar anormallik olarak işaretlenmiştir. Bu grafik, anormalliklerin görsel olarak işaretlenmesi, sistem güvenliği uzmanlarına veya analistlere olayları inceleme ve müdahale etme konusunda bir rehber sağlamak için kullanılabilir.



Şekil 7. Veri kümesi üzerinde anormal noktaların gösterimi

IV. SONUÇ

Bir sistemdeki normal dışı davranışların veya noktaların tespit edilmesi anormallik tespit teknikleri, uygulanarak gerçekleştirilmektedir. Geleneksel Makine Öğrenmesi tekniklerinin uzun vadeli bağımlı noktalardaki anormallikleri tespit etmesindeki yetersizliklerinden dolayı derin öğrenme teknikleri günümüzde daha başarılı sonuçlar üretmektedirler. Bu çalışma, derin öğrenme tekniklerinde LSTM tekniğini kullanarak , İHA sistem çağrılarının zaman serisindeki anormallikleri tespit etmeyi hedeflemiştir.

Kullanılan veri seti İHA sistem çağrı bilgilerini şöyle ki; sistem çağrı zaman damgası, sistem çağrısının durumu (onay veya ret) ve opsiyonel argümanlarını içermektedir. Veri ön işleme aşamasında, normal, delay ve pseudo-random örneklerinin ham günlük dosyaları işlenmiştir. Veri seti ön işleme aşamasından geçirildikten sonra Python programlama dili kullanılarak LSTM tekniği uygulandı. LSTM tekniği ile derinlemesine modern bir siber tehdit analizi sağlamayı amaçladığımız için veri seti bu çalışmanın amacına uygun bir veri setidir. Deneysel sonuçlar, LSTM tekniğinin sistem çağrılarının zaman serisindeki anormallikleri tespit etmedeki üstün performansını kanıtladı.

Statement of Conflicts of Interest

There is no conflict of interest between the authors.

Statement of Research and Publication Ethics

The authors declare that this study complies with Research and Publication Ethics

REFERENCES

[1] Damien, A., Fumey, M., Alata, E., Kaâniche, M., & Nicomette, V. (2018, November). Anomaly based intrusion detection for an avionic embedded system. In Aerospace Systems and Technology Conference (ASTC-2018).

[2] Biesecker, C. (2017). Boeing 757 testing shows airplanes vulnerable to hacking, DHS says. Avionics International, Nov.

[3] Schellekens, M. (2016). Car hacking: Navigating the regulatory landscape. Computer law & security review, 32(2), 307-315.

[4] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM computing surveys (CSUR), 41(3), 1-58.

[5] Esmaeili, F., Cassie, E., Nguyen, H. P. T., Plank, N. O., Unsworth, C. P., & Wang, A. (2023). Anomaly Detection for Sensor Signals Utilizing Deep Learning Autoencoder-Based Neural Networks. Bioengineering, 10(4), 405

[6] Ezeme, M., Azim, A., & Mahmoud, Q. H. (2017, December). An imputation-based augmented anomaly detection from large traces of operating system events. In Proceedings of the Fourth IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (pp. 43-52).

[7] Lippmann, R. P., Fried, D. J., Graf, I., Haines, J. W., Kendall, K. R., McClung, D., ... & Zissman, M. A. (2000, January). Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation. In Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00 (Vol. 2, pp. 12-26). IEEE.

[8] Boukerche, A., Zheng, L., & Alfandi, O. (2020). Outlier detection: Methods, models, and classification. ACM Computing Surveys (CSUR), 53(3), 1-37.

[9] Creech, G., & Hu, J. (2013). A semantic approach to host-based intrusion detection systems using contiguousand discontiguous system call patterns. IEEE Transactions on Computers, 63(4), 807-819.

[10] Meena, G., & Choudhary, R. R. (2017, July). A review paper on IDS classification using KDD 99 and NSL KDD dataset in WEKA. In 2017 International Conference on Computer, Communications and Electronics (Comptelix) (pp. 553-558). IEEE.

[11] Hafeez, I., Antikainen, M., Ding, A. Y., & Tarkoma, S. (2020). IoT-KEEPER: Detecting malicious IoT network activity using online traffic analysis at the edge. IEEE Transactions on Network and Service Management, 17(1), 45-59.

[12] Ring IV, J. H., Van Oort, C. M., Durst, S., White, V., Near, J. P., & Skalka, C. (2021). Methods for host-based intrusion detection with deep learning. Digital Threats: Research and Practice (DTRAP), 2(4), 1-29.

[13] Ezeme, O. M., Mahmoud, Q. H., Azim, A., & Michael, L. (2019). SysCall dataset: A dataset for context modeling and anomaly detection using system calls.

[14] Ezeme, O. M., Lescisin, M., Mahmoud, Q. H., & Azim, A. (2019). Deepanom: An ensemble deep framework for anomaly detection in system processes. In Advances in Artificial Intelligence: 32nd Canadian Conference on Artificial Intelligence, Canadian AI 2019, Kingston, ON, Canada, May 28–31, 2019, Proceedings 32 (pp. 549-555). Springer International Publishing.

[15] 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, 111745.

[16] Mvula, P. K., Branco, P., Jourdan, G. V., & Viktor, H. L. (2023). Evaluating Word Embedding Feature Extraction Techniques for Host-Based Intrusion Detection Systems. Discover Data, 1(1), 2.

[17] Terbuch, A., O'Leary, P., Khalili-Motlagh-Kasmaei, N., Auer, P., Zöhrer, A., & Winter, V. (2023). Detecting Anomalous Multivariate Time-Series via Hybrid Machine Learning. IEEE Transactions on Instrumentation and Measurement.

[18] Kim, J., Kang, H., & Kang, P. (2023). Time-series anomaly detection with stacked Transformer representations and 1D convolutional network. Engineering Applications of Artificial Intelligence, 120, 105964.

[19] Ma, Y., Xie, Z., Chen, S., Qiao, F., & Li, Z. (2023). Real-time detection of abnormal driving behavior based on long short-term memory network and regression residuals. Transportation research part C: emerging technologies, 146, 103983

[20] Aggarwal, S. (2023). LSTM based Anomaly Detection in Time Series for United States exports and imports.

[21] Ezeme, Okwudili; Mahmoud, Qusay; Azim, Akramul; Lescisin, Michael (2019), "SysCall Dataset: A Dataset for Context Modeling and Anomaly Detection using System Calls", Mendeley Data, V2, doi: 10.17632/vfvw7g8s8h.2