Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı 22, S. 204-211, Ocak 2021

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue 22, pp. 204-211, January 2021

Research Article

Zamansal Evrişimli Ağlarla Siber Saldırı Tespiti: Karşılaştırmalı Bir Analiz

Berna Çakır[1], Pelin Angin[2*]

[1] Orta Doğu Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümü, Ankara, Türkiye, (ORCID: 0000-0001-9610-459X), bernacakir01@gmail.com
[2*] Orta Doğu Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümü, Ankara, Türkiye (ORCID: 0000-0002-6419-2043), pangin@ceng.metu.edu.tr

(İlk Geliş Tarihi Aralık 2020 ve Kabul Tarihi Ocak 2021)

(**DOI:** 10.31590/ejosat.848784)

ATIF/REFERENCE: Çakır, B. & Angin, P. (2021). Zamansal Evrişimli Ağlarla Siber Saldırı Tespiti: Karşılaştırmalı Bir Analiz. *Avrupa Bilim ve Teknoloji Dergisi*, (22), 204-211.

Öz

Son yıllarda Nesnelerin İnterneti paradigmasının hızlı yükselişi ve bu yükselişin yarattığı büyük siber saldırı yüzeyi, otomatik saldırı tespit sistemlerinin önemini arttırmıştır. Özellikle daha önce gözlenmemiş sıfırıncı gün saldırılarının tespitinde klasik imza tabanlı saldırı tespit sistemleri yetersiz kalmaktadır. Bu durum siber güvenlik araştırmacılarını özellikle anomali tespiti için makine öğrenme tabanlı yöntemlere yönlendirmiştir. Literatürde derin öğrenme yöntemlerini bilgisayar ağlarında saldırı tespiti için kullanan birçok yöntem önerilmiş ve yüksek başarım elde etmiştir. Yakın zamanda ilk olarak videolarda aksiyon segmentasyonu için önerilen zamansal evrişimli ağlar (TCN), zaman serisi içeren öğrenme görevlerinde yüksek başarı elde ettiği halde, siber saldırı tespiti alanındaki etkinlikleri detaylı analiz edilmemiştir. Bu çalışmada TCN'nin saldırı tespiti konusunda başarımı irdelenmiştir. TCN'nin hem ikili sınıflandırma hem de anomali tespiti problemlerindeki başarımı, birçok saldırı tespiti probleminde yüksek başarım elde etmiş tekrarlayan sinir ağları ve tam bağlı sinir ağları yöntemleriyle kıyaslanmıştır. Elde edilen sonuçlar TCN'nin yüksek doğruluklu saldırı tespiti için ümit vaat eden bir yöntem olduğunu göstermektedir.

Anahtar Kelimeler: Derin sinir ağları, Zamansal evrişimli ağlar, Saldırı tespiti.

Cyber Attack Detection Using Temporal Convolutional Networks: A Comparative Analysis

Abstract

The rapid rise of the Internet of Things paradigm in recent years and the large attack surface created by this rise have increased the importance of automated detection of cyber attacks. Legacy signature-based intrusion detection systems are inadequate in detecting especially zero-days, which are attacks previously unobserved in computer networks. This has directed cyber security researchers towards machine learning based methods, especially for anomaly detection. Intrusion detection methods based on deep learning algorithms have been proposed, achieving high performance in a variety of tasks. Recently, temporal convolutional networks (TCN) were proposed for action segmentation in videos and have achieved great success in a variety of learning tasks on time series data. Their performance in intrusion detection tasks has not been analyzed in depth though. In this paper we analyze the performance of TCN for attack detection in networks. We compare the performance of TCN in both binary classification and anomaly detection problems with the performance of recurrent neural networks and fully connected feedforward neural networks. The results demonstrate that TCN is a promising method for high-accuracy attack detection.

Keywords: Deep neural networks, Temporal convolutional networks, Attack detection.

[*] Sorumlu Yazar: pangin@ceng.metu.edu.tr

1. Giriş

Son yıllarda Nesnelerin İnterneti paradigmasının hızlı yükselişi ve bu yükselişin yarattığı büyük siber saldırı yüzeyi, otomatik saldırı tespit sistemlerinin önemini arttırmıştır. Özellikle daha önce gözlenmemiş sıfırıncı gün saldırılarının tespitinde klasik imza tabanlı saldırı tespit sistemleri yetersiz kalmaktadır. Bu durum siber güvenlik araştırmacılarını özellikle anomali tespiti için makine öğrenme tabanlı yöntemlere yönlendirmiştir. Geçtiğimiz yıllarda derin öğrenme alanında yaşanan gelişmeler, birçok alanda (Eldem, 2020; Erduman vd., 2020) gözlemlenen yüksek başarımdan sonra bilgisayar ağlarında saldırı tespiti için de derin öğrenme yöntemlerini popüler bir araç haline getirmiştir.

Alanyazında derin öğrenme yöntemlerini bilgisayar ağlarında saldırı tespiti için kullanan birçok yöntem önerilmiştir (Thapa vd., 2020; Su vd., 2020). Gao vd. (2014) saldırı tespiti için derin inanç ağlarına dayalı bir model geliştirmiş ve destek vektör makineleri (SVM) ve çok katmanlı algılayıcılara (MLP) kıyasla üstün performansını göstermişlerdir. Derin sinir ağları ve spektral kümelemeden oluşan hibrit bir model Ma vd. (2016) tarafından önerilmiştir. Bu model NSL-KDD veri kümesinde %72 doğruluk elde etmiştir. Chuan-long vd. (2017), zaman bağımlılıkları olan verilerde saldırı tespitinde tekrarlayan sinir ağları (RNN) kullanmayı önermiş ve başarılı sonuçlar elde etmiştir. Yin vd. (2017) RNN tabanlı saldırı tespit modelleriyle KDD Cup'99 veri kümesinde yüksek doğruluk elde etmişlerdir. Evrişimli sinir ağları (CNN) tabanlı bir saldırı tespit modeli Li vd. (2017) tarafından önerilip, NSL-KDD üzerinde başarılı sonuçlar elde etmiştir. Behera vd. (2018) ağlarda izinsiz giriş tespiti için yine CNN kullanımını önermiş ve NSL-KDD veri kümesinde yüksek doğruluk elde etmiştir. Ayrıca, yaklaşımlarının sıfırıncı gün saldırılarını tespit edecek şekilde uyarlanabileceğini de belirtmişlerdir. Özel bir RNN yapısı olan uzun kısa-süreli bellek ağları (LSTM) tabanlı bir model Li vd. (2019) tarafından NSL-KDD veri kümesinde denenmiş ve %83 doğruluk ve F-1 skoru elde etmiştir. Vinayakumar vd. (2019) tarafından derin sinir ağlarını kullanan ölçeklenebilir, hibrit bir saldırı tespit yaklaşımı önerilmiştir. Dağıtılmış derin sinir ağları tabanlı modelin, bir dizi kıyaslama üzerinde geleneksel makine öğrenme tabanlı sınıflandırıcılardan daha iyi performans elde ettiği gösterilmiştir. Lopez-Martin vd. (2020), NSL-KDD'de %80,10 doğruluk sağlayan, denetimsiz saldırı tespiti için koşullu değişken otomatik kodlayıcı tabanlı bir model önermiştir.

Lea vd. (2016) tarafından ilk olarak videolarda aksiyon segmentasyonu için önerilen zamansal evrişimli ağlar (TCN), bu alanın yanı sıra uydu görüntülerinden zaman serisi tahminlemesi (Yan vd., 2020), dinamik tavsiye sistemleri (You vd., 2019), fizyolojik zaman serilerinde hastalık tespiti (Sandhiya ve Palani, 2020) gibi alanlarda da yüksek başarımlı sonuçlar elde etmiştir. TCN'yi şimdiye kadar siber saldırı tespiti alanında uygulamış çok az çalışma bulunmaktadır (Li vd., 2019) ve derin öğrenme alanında nispeten yeni olan bu modelin ağ saldırısı tespitinde önemli metriklerin tamamı temelinde farklı modellerle karşılaştırmalı analizi yapılmamıştır.

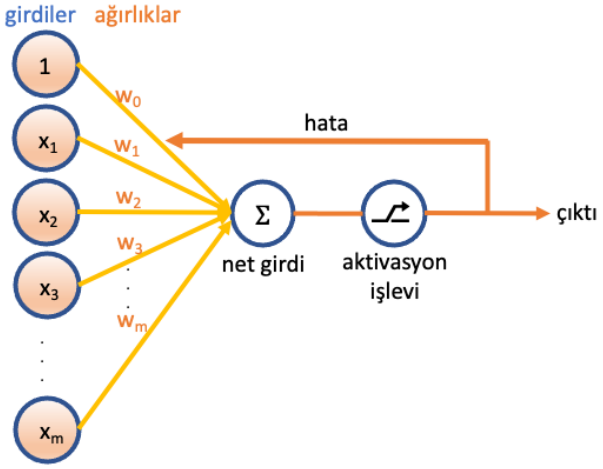
Bu çalışmada, alanyazındaki diğer derin öğrenme algoritmalarına kıyasla yeni bir yöntem olan TCN'nin, bilgisayar ağlarında saldırı tespiti konusunda başarımı irdelenmiştir. TCN'nin hem ikili sınıflandırma hem de anomali tespiti problemlerindeki başarımı, birçok saldırı tespiti probleminde yüksek başarım elde etmiş tekrarlamalı sinir ağları ve tam bağlı sinir ağları yöntemleriyle kıyaslanmıştır. Elde edilen sonuçlar TCN'nin yüksek doğruluklu saldırı tespiti için ümit vaat eden bir yöntem olduğunu göstermektedir.

2. Materyal ve Metot

Bu bölümde öncelikle derin sinir ağları ve bu çalışmada karşılaştırılan çeşitleri anlatılmış, zamansal evrişimli ağların yapısı açıklanmıştır.

2.1. Yapay Sinir Ağları

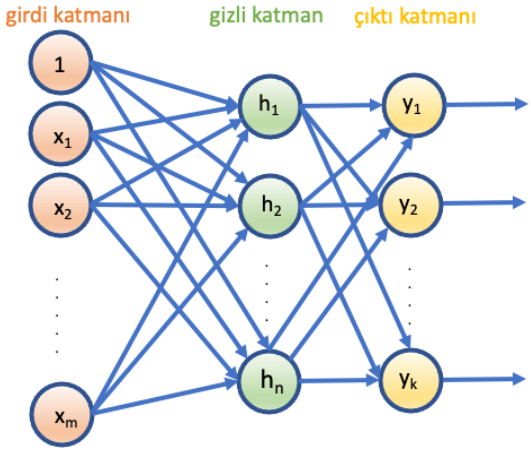
Yapay sinir ağları, tasarımı insan beyninin işleyişine benzeyen özel bir makine öğrenme modelleri kategorisidir. Sinyaller tarafından aktifleştirilen/engellenen karmaşık nöron ağları aracılığıyla bilginin işlenmesini ve iletilmesini simüle ederler (Goodfellow vd., 2016). Sinir ağı yapılarının ilk örneklerinden biri, bir çıkışa bağlı tek bir giriş katmanı içeren basit *algılayıcıdır*. Algılayıcı, Şekil 1'de gösterildiği gibi, bir aktivasyon işlevi ve bir dizi ağırlık kullanarak, nöronlardaki en basit süreçleri temsil eder. Bir algılayıcı ile makine öğrenimi, ağırlıkların her bir giriş düğümüne rastgele atanmasını ve çıktı değerini üretmek için giriş değerlerinin ağırlıklı toplamının bir etkinleştirme işlevi aracılığıyla geçişini içerir. Ağırlıklar eğitim süreci boyunca birden çok yinelemeyle ayarlanır ve eğitim sürecinin amacı çıktıdaki toplam hatayı en aza indirmektir. Hata, olması gereken çıktı ile model tarafından hesaplanan çıktı arasındaki fark olarak hesaplanır.



Şekil 1. Basit algılayıcı mimarisi

Çok katmanlı algılayıcılar (MLP), Şekil 2'de gösterildiği gibi, giriş ve çıkış katmanları arasında bir dizi gizli katman içeren ileri beslemeli sinir ağlarıdır. Şekil 2, her bir giriş düğümü her gizli düğüme ve benzer şekilde her gizli düğüm her çıkış düğümüne bağlı, tek bir gizli katmana sahip, tam bağlı bir sinir ağını göstermektedir. Tam bağlı sinir ağı daha fazla gizli katmandan oluştuğunda, gizli bir katmandaki her düğüm, bir sonraki gizli katmandaki her düğüme bağlanacaktır. Şekilde görüldüğü gibi, düğümleri birbirine bağlayan her kenar, minimum çıktı hatasını elde etmek için eğitim süreci boyunca güncellenen bir ağırlığa sahiptir. Her katmandaki gizli nöronların sayısı, giriş ve çıkış katmanı nöronlarının sayısından farklı olabilir. Ağ eğitimi, her bir yinelemede kenarların ağırlıklarını güncelleyen bir geri yayılma algoritması (Goodfellow vd., 2016) çalıştırmayı içerir. Giriş düğümlerinin sayısına giriş öznitelikleri vektörünün boyutuna göre karar verilirken, çıkış düğümlerinin sayısına öğrenme görevinin ne olduğuna (örn. çok sınıflı sınıflandırma, regresyon, ikili sınıflandırma vb.) göre karar

verilir. Bilgisayar ağlarında saldırı tespiti alanında uygulanması halinde, bu görev birkaç şekil alabilir. Örneğin, yalnızca saldırı varlığının tespiti için ikili sınıflandırma kullanılması görevinde çıktı, girdi verisinin normal ve saldırı sınıflarına ait olma olasılıkları; saldırı tiplerinin tespiti görevinde ise çıktının her bir saldırı sınıfına ve normal sınıfa ait olma olasılıkları olarak hesaplanır. Bir anomali tespiti görevinde ise modelin çıktısı, girdilerin eğitildiği sistem normalinden ne derece saptığını gösterecektir.



Şekil 2. Çok katmanlı sinir ağı mimarisi

Bir sinir ağının belirli bir girdi için doğru çıktıyı tahmin etme yeteneğini ölçmek için, *kayıp fonksiyonu* adı verilen bir fonksiyon kullanılır. Kayıp fonksiyonu, gerçek çıktı ile ağ tarafından tahmin edilen çıktı arasındaki farkı ölçer. En çok kullanılan kayıp fonksiyonlarından biri Ortalama Kare Hata (MSE) 'dır (Goodfellow vd., 2016):

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

Burada y gerçek çıktı değeri, y' model tarafından hesaplanan çıktı değeridir.

Başka bir kayıp fonksiyonu, ikili çapraz entropidir (BCE):

$$BCE(y, y') = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(y'_i) + (1 - y_i)(1 - \log(y'_i)) \quad (2)$$

Bir yapay sinir ağını eğitmenin amacı, kenarların ağırlık ve önyargı değerlerini ayarlayarak, seçilen kayıp fonksiyonunun çıktısını en aza indirmektir. Eğitim, örnek verileri girdi ve çıktı çiftleri olarak alan geri yayılım adı verilen denetimli bir eğitim algoritması kullanılarak yapılır. Eğitim algoritması, ağın ağırlıklarını rastgele seçerek başlar. Daha sonra kayıp en aza indirilene kadar iki faz, *ileri geçiş* ve *geri geçiş*, tekrarlanır.

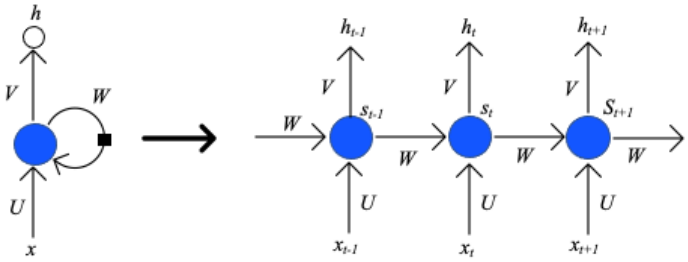
İleri geçiş aşamasında, ağın çıktısı ve kayıp fonksiyonunun değeri hesaplanır. Daha sonra geriye doğru geçiş aşamasında, ağırlıkların nasıl güncelleneceği ve böylece kaybın nasıl azaltılacağı bulunmakta ve ağırlıklar az miktarda güncellenmektedir. Kaybı azaltmak için ağırlıkların nasıl güncelleneceğini belirlemekte gradyan iniş algoritması kullanılır. Gradyan, bir ağırlık parametresi güncellendiğinde kayıp değerindeki değişiklik olarak tanımlanır. Gradyan iniş algoritması, zincir kuralı kullanarak kayıp fonksiyonuna göre her ağ parametresinin gradyanlarını hesaplar ve kayıp değerinin nasıl azaltılacağını belirlemek için gradyan yönünü kullanır.

Derin sinir ağları modelleri, yapay sinir ağlarındaki gizli katman sayısının klasik çok katmanlı algılayıcılara göre çok daha fazla olabildiği modellerdir. Tam bağlı ileri beslemeli bir derin sinir ağında (FCN), bir gizli katmandaki her nöron bir sonraki katmandaki her nörona bağlıdır ve ağın içinde döngüler (tekrarlamalar) bulunmamaktadır.

2.1.1. Tekrarlayan Sinir Ağları

İleri beslemeli sinir ağları, veri içindeki her örneğin birbirinden bağımsız olduğunu varsayar. Ancak bu varsayım her zaman doğru değildir. Doğal dil işleme, zaman serisi sınıflandırması vb. alanlarda veri noktaları arasında zaman içinde nedensel bir bağımlılık vardır. Bir siber saldırının ardışık adımlarının yarattığı ağ trafiğini veri olarak kabul ettiğimizde, yine veri noktaları arasında nedensel bir bağlılık olacaktır. Diğer birçok makine öğrenme algoritmasına benzer ileri beslemeli sinir ağları, bu tür bağımlılıkları görmezden gelir. Tekrarlayan sinir ağları (RNN), bu bağımlılıkları gözönünde bulundurmak için ileri beslemeli sinir ağlarının genişletilmiş bir versiyonudur.

RNN, sıralı veriler üzerinde çalışır ve diziyi işlerken, dizideki geçmiş öğeler hakkındaki bilgileri saklar. Bunu yapmak için, RNN, değeri dizinin geçmiş öğeleri tarafından belirlenen gizli bir durum parametresi tutar. RNN mimarisi Şekil 3'te gösterilmektedir.



Şekil 3. Tekrarlayan sinir ağları (RNN) mimarisi

RNN, veri dizisinde *t* zaman adımındaki elemanı işlerken *t-1* zaman adımındaki gizli durumu kullanır. Bir RNN'nin çıktısını hesaplamak için şu formül kullanılır:

$$s_t = \phi(Ux_t + W s_{t-1} + b_s)$$
$$h_t = \phi(Vx_t + b_h) \quad (3)$$

Bu formülde ϕ aktivasyon fonksiyonu, x_t, t anındaki girdi ve s_{t-1} gizli durumun önceki adımdaki girdisidir. Ağın t zamanındaki durumu ve çıktısı bu değerler kullanılarak hesaplanır. *U*, ileri besleme ağlarına benzer şekilde çıktı üzerinde o zaman adımındaki girdinin önemini belirleyen ağırlık parametreleridir. *W* ve *V*, geçmiş verilerin çıktı için önemini belirleyen ek ağırlık parametreleri, *b* önyargı parametresidir. Tekrarlayan sinir ağları için eğitim algoritması, tam bağlı sinir ağlarında olduğu gibi bu ağırlık parametrelerinin optimal değerlerini hesaplar.

Bir RNN, onu ileri beslemeli bir sinir ağına dönüştürmek için zamansal eksende Şekil 3'te sağ tarafta gösterildiği gibi açılabilir. Açma, her zaman adımı için RNN'nin bir kopyası oluşturularak yapılır (Goodfellow vd., 2016). Normal bir ileri beslemeli sinir ağı ile bir açılmış tekrarlayan sinir ağı arasındaki fark, açılmış ağın, ağırlıkların model boyunca paylaşılması gerektiğine dair ek kısıtlamaya sahip olmasıdır. Bu nedenle, tekrarlayan sinir ağları, geri yayılım algoritmasının Zaman Boyunca Geri Yayılım (BPTT) adı verilen bir varyantını kullanır (Williams ve Zipser, 1995).

BPTT'de, zincir kuralı aracılığıyla ağın her ağırlığına göre maliyet fonksiyonunun türevlerini hesaplamak için gradyanların tüm ağ boyunca geri yayılması gerekir. Ağın derinliği arttıkça, gradyanlar, zincir kuralını uygulamak için gereken birçok çarpma nedeniyle aşırı derecede büyük veya küçük hale gelebilir. Gradyanlar büyüdüğünde ağ, çok büyük ağırlık güncellemelerinin ağın dengesizliği ve ağırlıkların taşması gibi sorunlara neden olduğu *patlayan gradyan* probleminden muzdariptir. Öte yandan, çok küçük ağırlıklar *kaybolan gradyan* sorununa neden olur ve bu da ağın öğrenmesini sonlandırabilmek için çok küçük ağırlık güncellemelerine neden olur (Goodfellow vd., 2016). Kaybolan ve patlayan gradyanlar nedeniyle, RNN yalnızca kısa diziler üzerinde eğitilebilir.

Uzun kısa-süreli bellek ağları (LSTM), Hochreiter ve Schmidhuber (1997) tarafından RNN'nin kaybolan ve patlayan gradyan problemiyle mücadele etmek için önerilen bir RNN varyantıdır. LSTM, gizli durumdaki belirli bir öğenin saklanıp saklanmayacağını belirlemek için geçitler kullanır, bu da daha uzun bağımlılıkları hatırlamaya izin verir. Geçitler, çıktıları 0 ile 1 arasında olan sigmoid fonksiyonları kullanılarak gerçekleştirilir. Bir geçidin çıktısı 0'a yakınsa, LSTM, girdinin geçmesine izin vermez. LSTM klasik RNN'ye göre daha uzun dizileri modelleyebilmesinden dolayı birçok zamansal öğrenme probleminde başarılı sonuçlar elde eden bir model olmuştur. BPTT uzun zaman serilerinde yüksek bellek isterlerine sahiptir. LSTM'nin uzun zaman serilerine uygulanabilmesi için Sutskever (2013) *kesilmiş zamanda geri yayılım* (truncated BPTT) algoritmasını önermiştir. Bu çalışmada da LSTM modeli için tBPTT algoritması kullanılmıştır.

2.1.2. Zamansal Evrişimli Ağlar

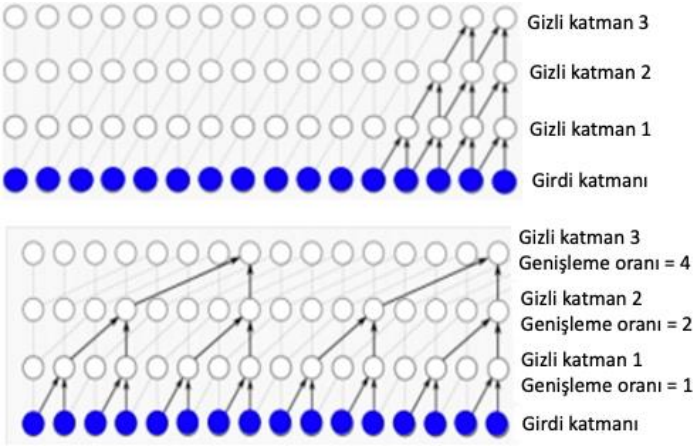
RNN'ler zaman ve bellek bakımından yüksek isterli olduğundan, bunların yerini alması için çeşitli mimariler geliştirilmiştir. Böyle bir çözüm, Zamansal Evrişimli Ağlardır (TCN). TCN, RNN'ye benzer, aynı uzunlukta bir çıktı üretmek için herhangi bir uzunlukta bir girdi dizisi alabilen bir mimaridir.

Evrişimli Sinir Ağları (CNN), girdinin ızgara benzeri bir yapıda düzenlendiği bir ileri beslemeli ağ türüdür ve özellikle bilgisayarlı görüde nesne tanıma alanında yüksek başarım elde etmiş bir modeldir. Bir CNN, evrişimli katmanlardan ve havuz katmanlarından oluşur. Evrişimli katman, bir dizi filtreden oluşur. İleri geçişte filtre, girdinin ızgara yapısı boyunca hareket ettirilir ve çıktıyı üretmek için girdi değeri ile filtre arasındaki nokta çarpımı hesaplanır. Filtrenin boyutu çıktının boyutundan daha küçüktür, bu da uzamsal özelliklerin çıkarılmasına izin verir. Her bir girdi öğesi için ağırlık yerine, evrişimli katmanların filtreleri, eğitim yoluyla optimize edilen ağ parametreleridir (Vinayakumar vd., 2017). Standart CNN diziler üzerinde çalışmaz. TCN, dizileri işlemek için CNN üzerine inşa edilmiş bir mimaridir.

Bir diziyi işleyebilmek için TCN, evrişimli katmanlar ve ardından sıfır doldurma katmanları kullanır. Evrişimli bir katmanın çıktısının boyutu, girdi boyutundan daha küçüktür. Böylece, girdi ve çıktının boyutunu aynı tutmak için sıfır dolgu katmanları kullanılır ve çıktı boyutu evrişimlerden sonra değişmez (Lea vd., 2016).

TCN'nin bir başka özelliği, evrişimli katmanda, t zamanındaki çıktının yalnızca önceki katmanda t zamanı ve daha önceki öğeler kullanılarak hesaplanmasıdır. Bu tür bir evrişime "nedensel evrişim" denir. Nedensel evrişimler, gelecekteki verilerin sonuçları etkilemesini engellediği için yalnızca geçmiş ve güncel bilgileri kullanarak dizi sınıflandırmasına izin verir.

TCN ayrıca verilere daha geriden itibaren bakabilmek için genişletilmiş evrişimler kullanır. Yalnızca nedensel evrişimler kullanıldığında, TCN ağ derinliğine doğrusal derinlikte bir geçmişe bakabilir. Genişletilmiş evrişimler, ağın geçmişi ağ derinliğine göre üssel derinlikte tutmasına izin verir. Şekil 4, genişletilmiş olan (altta) ve olmayan (üstte) nedensel evrişimleri göstermektedir.



Şekil 4. TCN'de nedensel evrişimli katmanlar

TCN, katmanlar arasında filtreleri paylaşır ve geri yayılım algoritmasının derinliği, yalnızca ağın derinliğine bağlıdır. Bu nedenle, TCN, tekrarlayan hücrelerinin tüm kapıları için kısmi sonuçları depolamak için belleğe ihtiyaç duyan tekrarlayan sinir ağlarına kıyasla düşük bellek gereksinimlerine sahiptir. Buna ek olarak, TCN ağının evrişimleri paralel olarak yapılabilirken, tekrarlayan mimariler paralelleştirilemez (TCN algoritmasının detaylı bir anlatımına Lea vd.'nin (2016) orijinal çalışmasında yer verilmiştir). Nesne tanıma alanında kullanılan evrişimli katmanlar iki boyutluyken, saldırı tespiti alanında tek boyutlu evrişimli sinir ağları (1D CNN) kullanılmaktadır (Li vd., 2019).

2.1.3. Otomatik Kodlayıcılar

Otomatik kodlayıcılar (autoencoders), kendi girdisini çıktı olarak yeniden oluşturabilmek üzere eğitilen bir yapay sinir ağı türüdür. Bir otomatik kodlayıcı iki bölümden oluşur:

• x girdisini h = f (x) kullanarak eşleyen bir kodlayıcı

• Kodlayıcı h çıkışından girdiyi yeniden yapılandıran bir kod çözücü.

Bir otomatik kodlayıcıyı eğitmenin amacı, modelin girişini olabildiğince iyi çıkarmasına olanak tanıyan en uygun parametreleri bulmaktır (Goodfellow vd., 2016).

Otomatik kodlayıcılar, giriş veri kümesinin önemli özelliklerini çıkarmak için kullanılabilir. Bir veri kümesinden yararlı özellikleri öğrenmenin bir yolu, x'e kıyasla daha küçük bir boyuta sahip bir otomatik kodlayıcı kullanmaktır. Bu, kodlayıcıyı girdinin tanımlayıcı özelliklerini çıkarmaya zorlar, böylece kod çözücü bunları kullanarak girdiyi yeniden yapılandırabilir. Kodlayıcının çıktı alanı girdi boyutundan daha büyükse, kodlayıcı girdiyi öğrenmeden belleğe alır. Bu tür bir otomatik kodlayıcı, eksik tamamlanmamış otomatik kodlayıcı olarak adlandırılır.

Otomatik kodlayıcılarda hem ileri beslemeli hem de tekrarlayan derin sinir ağı mimarileri kodlayıcı/kod çözücü olarak kullanılabilir. Bir otomatik kodlayıcı, özellikleri hem ayrı girdi noktalarından hem de dizilerden çıkarmak için kullanılabilir.

Anomali tespitinde bir otomatik kodlayıcı kullanırken, modelin veri kümesindeki normal temelin ne olduğunu öğrenmesi için, eğitim kümesinin sadece anormal olmayan örneklerden oluşturulması gerekir. Daha sonra, örneklerin anormal olarak değerlendirildiği bir yeniden yapılandırma hatası eşiği bulunmalıdır. Alanyazında bu eşiği bulmak için kullanılan farklı yöntemler vardır. Bu yöntemler genellikle sırasıyla yalnızca normal veriler ve karışık verilerden oluşan iki doğrulama veri kümesi kullanır. Eşikler, genellikle ortalama, standart sapma, vb. istatistiksel özellikleri kullanılarak normal doğrulama verilerinin yeniden yapılandırma hataları kullanılarak seçilir. Seçilen eşikler daha sonra ikinci bir doğrulama veri kümesinde test edilir.

Modeller normal veriler üzerinde eğitildikten sonra, tüm dizi için bir anormallik skor eşiği hesaplanır. Bir dizinin puanı kararlaştırılan eşiğin üzerindeyse, dizi bir anormallik olarak kabul edilir. LSTM modeliyle oluşturulan bir otomatik kodlayıcı için pencere boyutu, kodlayıcı ve kod çözücüdeki LSTM katmanlarının sayısı, LSTM katmanlarının boyutları, normalleştirme, öğrenme hızı vb. ağ parametrelerine ek olarak anormallik skor eşiği dahil olmak üzere hiperparametrelerin ayarlanması gerekir. TCN hiperparametrelerinin de benzer şekilde ayarlanması gerekir.

Bu çalışmada FCN, LSTM ve TCN tabanlı otomatik kodlayıcılar geliştirilmiş, geliştirilen bu modeller KDD Cup'99 verisi üzerinde anomali tespiti için çalıştırılmıştır. LSTM ve TCN otomatik kodlayıcılar için yeniden yapılandırma tek tek örnekler için değil, diziler için yapılmıştır. Bu otomatik kodlayıcılar için sabit boyutlu diziler, kayan bir pencere kullanılarak tüm veri kümesinden çıkarılmıştır. Her pencerenin etiketine, pencerede anormal örnekler olup olmadığına göre karar verilmiştir. Anormalliğin konumuna bakılmaksızın bir pencere içinde anormal veriler mevcutsa, otomatik kodlayıcının bir pencere sırasını yeniden yapılandıramaması beklenir. Yeniden yapılandırma hataları hesaplanırken, dizinin tamamı kullanılmıştır.

2.2. KDD Cup'99 Veri Kümesi

KDD Cup'99 veri kümesi (KDD Cup 1999), Üçüncü Uluslararası Bilgi Keşfi ve Veri Madenciliği Araçları Yarışması için oluşturulmuş, etiketli bir saldırı tespit veri kümesidir. Veriler, ABD Hava Kuvvetleri ağının simülasyonundan toplanmıştır. Simülasyon sırasında, model eğitimi için 7 haftalık TCP dökümleri kaydedilmiş, ayrıca test için 2 haftalık ağ trafiği verileri kaydedilmiştir. Veri kümesinin oluşturulmasında kaydedilen ağ trafiğinden elde edilen 41 tane öznitelik kullanılmıştır. Bu öznitelikler içinde bir bağlantının başlangıcı ve bitişi arasında geçen süre, kullanılan iletişim protokolü, iletilen toplam veri miktarı gibi özelliklerin yanı sıra geçersiz giriş denemesi sayısı, sunucu hata oranı gibi özellikler yer almaktadır. Veri kümesinde toplam 4 ana kategoride 24 çeşit ağ saldırısına yer verilmiştir. Bu kategoriler şu şekildedir:

- Hizmet Reddi (DoS): Amacı ağın hizmetlerine erişimi engelletmek olan saldırılardır.
- Araştırma (Probe): Amacı ağ hakkında bilgi toplamak olan saldırılardır. Bu saldırılar, hedefin güvenlik açıklarını tespit etmek için bağlantı noktalarını veya IP adreslerini taramak gibi yöntemler kullanır.
- Uzaktan yerele (R2L): R2L saldırıları, saldırganın ağ üzerinden hedefe paket gönderebildiği, ancak oturum açma haklarına sahip olmadığı saldırılardır. Saldırının amacı, makineye kullanıcı erişimi sağlamak için bir zafiyetten yararlanmaktır.
- Kullanıcıdan köke (U2R): U2R saldırıları, saldırganın hedefe normal kullanıcı erişimine sahip olmasına karşın kök erişimi sağlamaya çalıştığı türden saldırılardır. U2R saldırıları, erişim sağlamak için parola izleme gibi yöntemler kullanır.

KDD Cup'99 yaklaşık 20 yıllık bir veri kümesi olmasına rağmen, içerdiği saldırı çeşitliliği, toplam veri sayısının büyüklüğü ve zaman yayılımı gibi nedenlerle halen saldırı tespiti algoritmalarının başarımının sınanmasında temel kıyaslama veri kümelerinden biri olarak kullanılmaktadır. KDD Cup'99, 38 sürekli/ikili ve 3 kategorik özellik içermektedir. Kategorik özellikler şunlardır: 3 kategoriye (tcp, udp, icmp) sahip "Protokol türü", 70 kategoriye sahip (ftp, telnet, http vb.) "hizmet" ve 11 kategori içeren "İşaret". Yapay sinir ağları sayısal girdiler gerektirir; bu nedenle bu çalışmada kategorik özellikler tek sıcak kodlama kullanılarak sayısal değerlere dönüştürülmüştür. Numerik özellikler [0, 1] aralığında olacak şekilde normalize edilmiştir.

KDD Cup'99 veri kümesinin temel bir sorunu eğitim kümesindeki sınıfların dağılımının dengeli olmamasıdır. Eğitim veri kümesi %19,69 normal örnek, %79,24 DoS saldırıları, %0,83 Probe saldırıları, %0,23 R2L saldırıları ve %0,01 R2L saldırıları olacak şekilde toplam 494.019 örnek içermektedir. Bunun oluşturacağı yanlılığı önlemek için sınıflandırma modelleri eğitilirken örnekleme yöntemi kullanılmıştır. Modeller eğitim kümesinin içinden zamansal bağlantılar korunarak her bir sınıf için 2000 örnek seçilerek eğitilmiştir. 2000'den az örnek sayısına sahip olan R2L ve U2R sınıfları için mevcut örneklerin tamamı eğitim kümesine dahil edilmiştir.

3. Araştırma Sonuçları ve Tartışma

TCN modelinin KDD Cup'99 veri kümesi üzerinde üç tip saldırı tespiti görevi için başarımının değerlendirilmesi yapılmıştır. Bu görevler, (1) denetimli öğrenmeyle eğitilen modelle ağ trafiğinin normal ya da saldırı şeklinde ikili sınıflandırılması, (2) ağ trafiğinde değişik saldırı tiplerini tespit, (3) ağ trafiğinde anomalileri tespit etmedir. Sonuçlar aynı veri kümesi üzerinde LSTM ve FCN modellerinin başarımıyla karşılaştırılmıştır. Yukarıda da belirtildiği gibi, FCN modeli veri noktalarını birbirinden bağımsız olarak sınıflandırırken, LSTM ve TCN noktalar arasındaki zamansal bağlantıları dikkate alarak içinde saldırı noktaları bulunan dizileri saldırı olarak sınıflandırmıştır.

Sonuçların raporlanmasında saldırı tespiti alanyazınında sıklıkla kullanılan doğruluk (accuracy), kesinlik (precision), duyarlılık (recall) ve F-1 skoru metrikleri kullanılmıştır. Bu metrikler şu şekilde açıklanabilir:

- Gerçek pozitif (GP): Gerçekte saldırı olup model tarafından da saldırı olarak sınıflandırılan veri sayısı

- Yalancı negatif (YN): Gerçekte saldırı olup model tarafından normal olarak sınıflandırılan veri sayısı
- Yalancı pozitif (YP): Gerçekte normal olup model tarafından saldırı olarak sınıflandırılan veri sayısı
- Gerçek negative (GN): Gerçekte normal olup model tarafından da normal olarak sınıflandırılan veri sayısı

$$Doğruluk = (GP + GN)/(GP+GN+YP+YN)$$

$$Kesinlik = GP/(GP+YP)$$

$$Duyarlılık = (GP)/(GP+YN)$$

$$F1 = (2 * Kesinlik * Duyarlılık)/(Kesinlik+Duyarlılık)$$

3.1. İkili Sınıflandırma

İkili sınıflandırıcılar eğitim kümesini normal ve saldırı şeklinde iki sınıfa ayırıp, öğrenilen modelle test kümesindeki verileri bu iki sınıfta kategorize etmek için kullanılmıştır. Model eğitiminde en yüksek başarımı sağlayacak parametreleri keşfetmek için her model için hiperparametre eniyileştirmesi uygulanmıştır.

Tam bağlı ileri beslemeli sinir ağları için denenmiş olan hiperparametre değerleri Tablo 1'de verilmiştir. En iyi sonuçlar 0,001 öğrenme hızı, sırasıyla 32-64-32 nörondan oluşan 3 gizli katmanla, Adam (Kingma ve Ba, 2015) en iyileştiricisiyle elde edilmiştir, katmanlar arası ReLU aktivasyon fonksiyonu kullanılmıştır.

Tablo 1. FCN için hiperparametre değerleri

Hiperparametre	Değerler
Öğrenme hızı	0,00001; 0,0001; 0,001; 0,1; 0,5
Gizli katman sayısı	1-5
Gizli katmandaki nöron sayısı	2^3 – 2^8
Eniyileştirici	Adam, SGD

LSTM için denenmiş olan hiperparametre değerleri Tablo 2'de verilmiştir. En iyi sonuçlar 0,001 öğrenme hızı, sırasıyla 32-20 nörondan oluşan 2 gizli katmanla, Adam en iyileştiricisi ve 64 tBPTT adım sayısıyla elde edilmiştir.

Tablo 2. LSTM için hiperparametre değerleri

Hiperparametre	Değerler
Öğrenme hızı	0,00001; 0,0001; 0,001; 0,1
Gizli katman sayısı	1-4
Gizli katmandaki nöron sayısı	32-256
Eniyileştirici	Adam, SGD
tBPTT adım sayısı	32-512

TCN için denenmiş olan hiperparametre değerleri ve en iyi performansı elde eden hiperparametre değerleri Tablo 3'te verilmiştir.

Tablo 3. TCN için hiperparametre değerleri

Hiperparametre	Değerler	En iyi değerler
Öğrenme hızı	0,00001-0,1	0,0001 – 0,001
Evrişimli katmanlarda filtre sayısı	16; 32; 64	64
Evrişimli katman çekirdek sayısı	2; 4; 8; 16	4
Katmanlar arası bırakma oranı	0,1; 0,3; 0,5	0,1
Artık blok sayısı	1, 2	1
Eniyileştirici	Adam, SGD	Adam
Evrişimli katmanlar için aktivasyon fonksiyonu	Linear, ReLU	ReLU
Dizi uzunluğu	32-256	64

Tablo 4 ikili sınıflandırma deney sonuçlarını özetlemektedir. TCN modeli F1 skoru açısından FCN modeline göre daha yüksek başarım sağlamış, LSTM'le de aynı başarıyı elde etmiştir. Kesinlik metriği açısından da durum benzerdir. Genel olarak LSTM modeliyle benzer performansa sahip olduğu gözlenmektedir. Eğitim süresi ve kaynak gereksinimlerinin LSTM modeline kıyasla daha az olması, ikili sınıflandırma içeren saldırı tespitinde tercih nedenidir.

Tablo 4. İkili sınıflandırma deney sonuçları

Model	Doğruluk	Kesinlik	Duyarlılık	F1
FCN	0.935	0.927	**0.992**	0.958
LSTM	**0.942**	**0.939**	0.989	**0.963**
TCN	0.941	**0.939**	0.988	**0.963**

3.2. Saldırı Tipine Göre Tespit Performansı

Saldırı tiplerinin tespiti deneylerinde ağ trafiğindeki saldırıları tespit etmenin yanı sıra, modellerin bu saldırıların tiplerini de tespit etmedeki başarımı sınanmıştır. Tablo 5 modellerin her bir saldırı tipini tespit oranı açısından başarımını göstermektedir. Sonuçlara göre FCN modelinin Probe, R2L, U2R saldırılarını tespit etmede diğer modellere göre daha başarılı olduğu gözlenmiştir. LSTM ve TCN birbirine benzer sonuçlar elde etmiştir. TCN modelinin veri kümesinde çok nadir sayıda bulunan R2L ve U2R saldırılarının tespitinde çok başarılı olmadığı gözlenmiştir. Her iki modelin de R2L ve U2R saldırılarını yüksek başarımlı tespiti için eğitim kümesinde daha fazla örneğe ihtiyaç olduğu değerlendirilmektedir.

Tablo 5. Saldırı tipi tespit oranları

Model	DoS	Probe	R2L	U2R
FCN	0.940	**0.958**	**0.737**	**0.828**
LSTM	**0.970**	0.892	0.517	0.7
TCN	0.961	0.874	0.518	0.714

3.3. Otomatik Kodlayıcı Modelleri

Bu deneylerde otomatik kodlayıcı tabanlı denetimsiz yöntemler, ağın normal davranışını öğrenmek ve normalden sapan örnekleri bulmaya çalışmak için yalnızca normal eğitim kümesinin tamamıyla eğitilmiştir. Test verisi normal davranışla uyumluluğuna göre sınıflandırılmış, eşik değerin üzerinde farklılıklar anomali olarak tespit edilmiştir. Otomatik kodlayıcı mimarisi olarak üç katmanlı bir kodlayıcıyı takiben üç katmanlı kod çözücü kullanılmıştır. Girdiden sonraki ilk katmanda 300 nöron, ikinci katmanda 200 nöron, üçüncü katmanda 75 nöron bulunmaktadır. Kodlayıcı çıktısı 10 nöronla temsil edilmektedir. FCN ve TCN modelleri için katmanlarda ReLU aktivasyon işlevi kullanılmış, LSTM için tanh işlevi kullanılmıştır. LSTM ve TCN modellerinde dizi uzunluğu önceki deneylerde olduğu gibi 64 olarak belirlenmiştir. Modellerin diğer parametreleri önceki deneylerdeki gibi belirlenmiştir.

Tablo 6 bahsedilen modellerle geliştirilen otomatik kodlayıcıların performanslarını göstermektedir. Tabloda görüldüğü gibi modellerin performansları genel olarak birbirine yakındır. Bunun yanında TCN'in LSTM'den duyarlılık metriği dışında iyi sonuç elde etmesi anomali tespiti problemlerinde tercih edilmesini sağlayacak bir faktördür.

Tablo 6. Otomatik kodlayıcı deney sonuçları

Model	Doğruluk	Kesinlik	Duyarlılık	F1
FCN-AE	**0.953**	**0.982**	0.959	**0.970**
LSTM-AE	0.916	0.940	**0.971**	0.955
TCN-AE	0.930	0.979	0.948	0.963

3.4. Diğer Makine Öğrenme Modelleri

KDD Cup'99 veri kümesinin üzerinde makine öğrenme yöntemleriyle sınıflandırma için yapılmış birçok çalışma bulunmaktadır. Tablo 7 alanyazındaki klasik makine öğrenme modelleri ve TCN'nin KDD Cup'99 tüm test veri kümesi üzerinde sınıflandırma performanslarını göstermektedir. TCN dışındaki modellerin sonuçları Özgür ve Erdem'in (2017) KDD Cup'99 üzerindeki analizinden alıntılanmıştır.

Model	**Doğruluk**	**F1**
AdaBoost M1	0.915	0.945
BayesNet	0.916	0.945
Karar tabloları	0.947	0.966
J48	0.934	0.957
Lojistik regresyon	0.815	0.871
MLP	0.918	0.947
Naïve Bayes	0.914	0.944
OneR	0.907	0.939
Rastgele Ormanlar	0.924	0.950
RBF	0.852	0.900
SGD	0.922	0.949
SMO	0.919	0.947
TCN	0.942	0.962

Tabloda görüldüğü gibi TCN KDD Cup'99 veri kümesi üzerinde karar tabloları haricinde klasik makine öğrenme yöntemlerine kıyasla da başarılı sonuçlar elde etmiştir.

4. Sonuç

Bu çalışmada yakın zamanda özellikle bilgisayarlı görü alanında yüksek başarı sağlamış bir derin öğrenme yöntemi olan zamansal evrişimli ağların bilgisayar ağlarında saldırı tespiti için etkinliği değerlendirilmiştir. Yöntemin başarımı saldırı tespiti alanyazınındaki temel kıyaslama veri kümelerinden biri olan KDD Cup'99 üzerinde ikili sınıflandırma, saldırı tipi tespiti ve anomali tespiti görevleri için yine zamansal bir derin öğrenme yöntemi olan LSTM ve tam bağlı ileri beslemeli sinir ağlarının başarımıyla karşılaştırılmıştır. TCN'nin özellikle ağ trafiğini normal ve saldırı şeklinde kategorize eden ikili sınıflandırmada en az LSTM kadar başarılı olduğu görülmüştür. Ayrıca birçok klasik makine öğrenme modelinden yüksek başarım elde ettiği de gözlenmiştir.

Çalışmada elde edilen sonuçlar doğrultusunda TCN'nin saldırı tespiti problemleri için ümit vaat eden bir model olduğunu söylemek mümkündür. LSTM gibi tekrarlayan sinir ağları modellerine kıyasla sağladığı performans avantajları, önümüzdeki yıllarda TCN'yi siber saldırı tespiti için tercih edilen bir yöntem haline getirebilecektir. TCN KDD Cup'99 üzerindeki performans analizinde yüksek başarım elde etmiş olsa da karşılaştırma yapılan modellerin bir kısmının performansının TCN performansından çok aşağıda olmadığı görülmüştür. Değerlendirmede kullanılan KDD Cup'99 veri kümesi sınırlı sayıda saldırı kategorisi içermektedir. Siber saldırıların basamaklarını belirgin şekilde farklı data noktalarıyla modelleyen ya da çok basamaklı saldırılar içeren veri kümelerinin oluşturulması ve değerlendirmede kullanılması durumunda TCN'nin etkinliğinin daha net bir şekilde gözlemlenmesi mümkün olabilecektir. Gelecek çalışmalarımızda TCN'nin farklı siber saldırı veri kümeleri üzerindeki etkinliği de değerlendirilerek, kullanımının özellikle hangi durumlarda yüksek performans sağladığı ortaya konulacaktır.

Kaynakça

Bai, S., Kolter, J. Z., & Koltun, V. (2018). Convolutional sequence modeling revisited. In *International Conference on Learning Representations (ICLR) Workshop*. Vancouver, BC, Canada.

Behera, S., Pradhan, A., & Dash, R. (2018). Deep neural network architecture for anomaly based intrusion detection system. In *5th International Conference on Signal Processing and Integrated Networks (SPIN 2018)* (270-274). Noida, India. DOI: 10.1109/SPIN.2018.8474162.

Chuan-long, Y., Yue-fei, Z., Jin-long, F., & Xin-zheng, H. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access, 5,* 21954-21961. DOI: 10.1109/ACCESS.2017.2762418.

Eldem, A. (2020). An Application of Deep Neural Network for Classification of Wheat Seeds. *Avrupa Bilim ve Teknoloji Dergisi*, (19), 213-220. DOI: 10.31590/ejosat.719048.

Erduman, A., Yüzer, E., Durusu, A., Yıldız, F. (2020). An Educational Kit to Promote Teaching of Photovoltaic Systems. *Avrupa Bilim ve Teknoloji Dergisi*, (19), 916-922. DOI: 10.31590/ejosat.745109.

Gao, N., Gao, L., Gao, Q., & Wang, H. (2014). An intrusion detection model based on deep belief networks. In *IEEE International Conference on Advanced Cloud and Big Data* (247-252). Huangshan, China. DOI: 10.1109/CBD.2014.41.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. ISBN: 978-0262035613.

Graves, A. (2012). *Supervised sequence labeling with recurrent neural networks.* Springer. ISBN: 978-3-642-24797-2.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation, 9*(8), 1735-1780. DOI: 10.1162/neco.1997.9.8.1735.

KDD Cup (1999). [Data file and codebook]. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

Khan, F. A., Gumaei, A., Derhab, A., & Hussain, A. (2019). TSDL: A twostage deep learning model for efficient network intrusion detection. *IEEE Access, 7,* 30373–30385. DOI: 10.1109/ACCESS.2019.2899721.

Kim, J., & Kim, H. (2016). Applying recurrent neural network to intrusion detection with hessian free optimization. In H. Kim, D. Choi (Eds.), *Information Security Applications. WISA 2015. Lecture Notes in Computer Science* (1-14). Springer. DOI: 10.1007/978-3-319-31875-2_30.

Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations (ICLR)* (1-15). San Diego, CA, USA.

Lea C., Vidal R., Reiter A., & Hager G. D. (2016). Temporal convolutional networks: A unified approach to action segmentation. In G. Hua & H. Jégou (Eds.), *Computer Vision – ECCV 2016 Workshops. ECCV 2016. Lecture Notes in Computer Science* (47-54). Springer.

Li, Z., Qin, Z., Huang, Z., Yang, X., & Ye, S. (2017). Intrusion detection using convolutional neural networks for representation learning. In D. Liu, S. Xie, Y. Li, D. Zhao, & E. M. El-Alfy (Eds.), *Neural Information Processing. ICONIP 2017. Lecture Notes in Computer Science.* (858–866). Springer. DOI: 10.1007/978-3-319-70139-4_87.

Li, Z., Rios, A. L. G., Xu, G., & Trajkovic, L. (2019). Machine learning techniques for classifying network anomalies and intrusions. In *IEEE International Symposium on Circuits and Systems (ISCAS)* (1–5). Sapporo, Japan. DOI: 10.1109/ISCAS.2019.8702583.

Li, Z., Qin, Z., Shen, P., & Jiang, L. (2019) Intrusion Detection Using Temporal Convolutional Networks. In: Gedeon T., Wong K., Lee M. (eds) *Neural Information Processing*. ICONIP 2019. Communications in Computer and Information Science, vol 1142. Springer, Cham. https://doi.org/10.1007/978-3-030-36808-1_19.

Lopez-Martin, M., Carro, C., Sanchez-Esguevillas, A., & Lloret, J. (2017). Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in IoT. *Sensors, 17*(9), 1967. DOI: 10.3390/s17091967.

Ma, T., Wang, F., Cheng, J., Yu, Y., & Chen, X. (2016). A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks. *Sensors, 16*(10), 1701. DOI: 10.3390/s16101701.

Özgür, A. & Erdem, H. (2017). The impact of using large training data set KDD99 on classification accuracy. *PeerJ Preprints* 5 e2838v1. DOI: 10.7287/peerj.preprints.2838v1

Pascanu, R., Stokes, J. W., Sanossian, H., Marinescu, M., & Thomas, A. (2015). Malware classification with recurrent networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing* (1916-1920).

Sandhiya, S., & Palani, U. (2020). An effective disease prediction system using incremental feature selection and temporal convolutional neural network. *Journal of Ambient Intelligence and Humanized Computing, 11,* 5547–5560. DOI: 10.1007/s12652-020-01910-6.

Staudemeyer, R. C. (2015). Applying long short-term memory recurrent neural networks to intrusion detection. *South African Computer Journal*, 56, 136–154. DOI: 10.18489/SACJ.V56I1.248.

Su, T., Sun, H., Zhu, J., Wang, S., & Li, Y. (2020). BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset. *IEEE Access, 8,* 29575–29585. DOI: 10.1109/ACCESS.2020.2972627.

Sutskever, I. (2013). Training Recurrent Neural Networks. *PhD thesis.* University of Toronto, Ontario, Canada.

Thapa, N., Liu, Z., KC, D. B., Gokaraju, B., & Roy, K. (2020). Comparison of machine learning and deep learning models for network intrusion detection systems. *Future Internet, 12*(10), 167. DOI: 10.3390/fi12100167.

Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. *IEEE Access, 7,* 41525–41550. DOI: 10.1109/ACCESS.2017.2762418.

Vinayakumar, R., Soman, K., & Poornachandran, P. (2017). Applying convolutional neural network for network intrusion detection. In *IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (1222–1228). Udupi, India.

Williams, R. J., & Zipser, D. (1995). Gradient-based learning algorithms for recurrent networks and their computational complexity. In Y. Chauvin & D. E. Rumelhart (Eds.) *Backpropagation: Theory, Architectures, and Applications* (433-486). L. Erlbaum Associates Inc. Hillsdale, NJ, USA.

Yan, J., Chen, X., Chen, Y., & Liang, D. (2020). Multistep prediction of land cover from dense time series remote sensing images with temporal convolutional networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *13*, 5149-5161.

Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access, 5,* 21954–21961. DOI: 10.1109/ACCESS.2017.2762418.

You, J., Wang, Y., Pal, A., Eksombatchai, P., Rosenburg, C., & Leskovec, J. (2019). Hierarchical temporal convolutional networks for dynamic recommender systems. In *The World Wide Web Conference* (2236-2246). Association for Computing Machinery. DOI: 10.1145/3308558.3313747.