Keystroke Biyometrik Verilerinin Kimlik Kontrolü İçin Kullanımı

Yiğit Kürşad Ersoy
Erciyes Üniversitesi Bilgisayar Mühendisliği
1030516444
yigit@yigit.kim

04.05.2019

Özet

Günümüzde birçok büyük uygulama, firma ve kuruluş sahip oldukları kullanıcı verilerini korumakta başarısız olmuşlardır. Yeni şifreleme yöntemlerine rağmen halen veri sızıntısı olmaktadır. Sızdırılan veriler daha sonra kullanıcıların ilgili uygulama/sistemdeki hesaplarına erişim sağlamak için kullanılmakta. Keystroke dinamik korumalı sistemler, kullanıcıların tuşlama ritimlerini tanıyarak giriş yapmaya çalışan kullanıcının "genuine" (gerçek) veya "imposter" (taklitçi) olup olmadığının kararını verebilmektedir.

Böylece; çalınmış bilişimsel kaynaklar (veri) ile dahi ilgili kullanıcının bizzat kendisi hariç kimsenin hesaba erişememesi sağlanmış oluyor. Bu araştırma kapsamında Keystroke dinamiklerinin makine öğrenmesi ve derin öğrenme yöntemleri ile uygulanışı incelenecek, literatürdeki diğer yaklaşımlar ile karşılaştırılacaktır.

Anahtar Kelimeler
Bilgi güvenliği, kullanıcı giriş kontrolü, tuşlama ritmi, tuş vuruş dinamikleri, tuşlama biyometrik verileri.

1. Giriş

Geleneksel kimlik doğrulama yöntemleri; biyometrik sistemlerin daha karşılanabilir, implementasyon süreci kısa ve gün geçtikçe artan doğruluk oranları karşısında yerini biyometrik kimlik doğrulamaya devretmeye başladı [1]. Bunun başlıca sebepleri arasında veri sızıntıları, yaygınlaşan siber saldırılar ve kaçınılmaz olan konvansiyonel bilgi hırsızlığı bulunmaktadır. Durmaksızın kendini güncelleyen siber saldırı yöntemlerine karşın güvenlik sistemlerinin de kendilerini güncelleme gereksinimleri kaçınılmazdır.

Biyometrik kimlik doğrulama yöntemlerini bünyesine dahil etmeyen güvenlik sistemleri, üçüncü parti kişilerin kimlik verilerini elde etmesi durumuna karşı savunmasızdır. Çünkü dahili veya harici olarak herhangi bir "genuineness verification" (gerçeklik doğrulama) modülüne sahip değildir, bu durumda da kimlik bilgisine sahip olan herhangi bir birey (veya otomasyon), sisteme erişim sağlayabilir. Keystroke dinamiği biyometriği bu konuda ekstra bir güvenlik katmanı sağlamakta.

Üçüncü parti birey kimlik bilgilerine sahip olsa dahi ilgili kullanıcı ile aynı tuşlama ritmine sahip olmadığı için giriş yapması mümkün olmayacaktır. Keystroke dinamiği implementasyonu bununla sınırlı değil, diğer uygulama tipi [2] ise şifresiz bir şekilde, yalnızca verilen paragrafın tuşlanma ritmine göre kontrol sağlanması. Diğer bir deyişle, giriş için şifreye ihtiyaç duyulmuyor, sadece (sürekli değişen) paragrafın tuşlama ritminin ilgili kullanıcının tuşlama ritmi ile karşılaştırılması ile yapılıyor. Doğal olarak bu daha uzun soluklu projelerde sağlıklı sonuçlar vermekte. Çünkü derin öğrenme modülünün kullanıcıyı şifresiz bir şekilde kimliklendirmesi için bir kaynağa göre [3] 400 farklı paragraf tuşlaması ancak makul bir öğrenme seti hazırlıyor. Belirtilen yaklaşım her ne kadar zaman ve bütçe anlamında masraflı olsa da keystroke biyometriğinin en güvenli implementasyonu olduğu söylenebilir. Ne de olsa ortada sızıntısı yapılabilecek bir kimlik verisi (şifre) yoksa "data breach" tehlikesi de bir önem arz etmemektedir. Keystroke dinamiği biyometriğini kimlik kontrolü konusunda diğer biyometrik yöntemlerden daha cazip kılan ise hiçbir ekstra çevre aygıta ihtiyaç duyulmaksızın [4] implementasyonunun mümkün olmasıdır.

2. Literatür Taraması

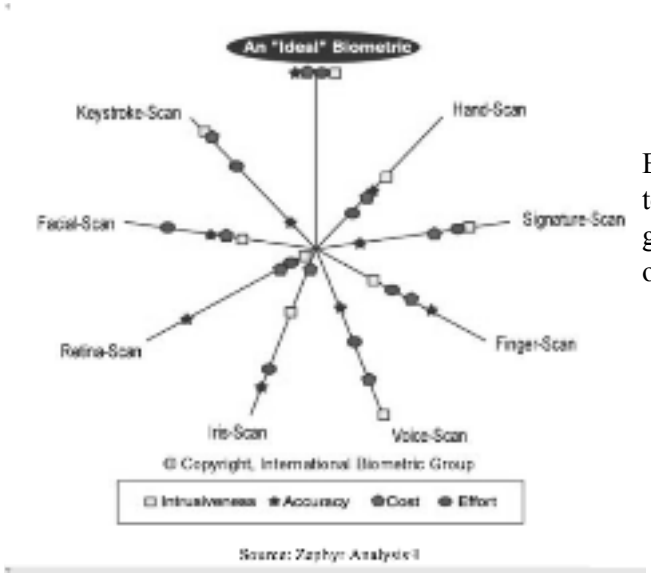
Keystroke dinamikleri, her bireyin kendine has tuşlama biçimine sahip olduğu öne süren [5] biyometrik yöntemdir. Keystroke dinamikleri temel olarak 3 temel noktadan oluşur: kullanıcı tuşlama ritimlerinin elde edilmesi, tuşlama ritimleri verileri kullanılarak (eğer öğrenme algoritması ise) yapay zekanın eğitilmesi, son olarak da devamlı olarak kimlik kontrolü. Ayrıca sistem, giriş yapmaya çalışan kullanıcının gerçek/doğru olduğuna karar verdikten sonra bu yeni tuşlama ritmini de eğitimine devamlı olarak dahil ediyor. Böylelikle, kullanıcıların ruh hali, yaşı, davranışları, tuşlama karakteristikleri değişse dahi sistem kendini güncel tutabiliyor, zaman aşımına uğramıyor.

Keystroke biyometriğini diğer biyometrik tekniklerden cazip kılan noktalar düşük masraf, düşük efor, takliti ve yanıltılma oranının ortalama olması ve entegresinin diğer tüm biyometrik tekniklere nazaran çok daha basit/kolay olması. Keystroke dinamiği karakter tabanlı kimlik kontrolünü kullanan tüm bilişim sistemlerine entegre edilebilir [4]. Diğer biyometrik teknikler bir veya birden fazla harici donanıma ihtiyaç duymakta, özetle mevcut sistem mimarisinin değiştirilmesini zorunlu kılmaktadır.

Şekil-1 incelendiği takdirde görülecektir ki Keystroke biyometriği geleneksel kimlik kontrol yöntemlerine karşı yüksek güvenlik sağlamakta, aynı zamanda masraf bakımından da düşük olmaktadır.

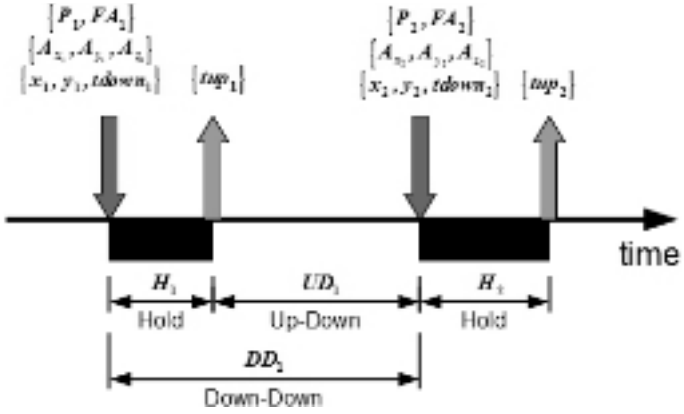
Şekil - 1 Güvenlik Mekanizmalarının Karşılaştırılması [6]

Diğer biyometrik teknikler ile karşılaştırılırsa ise daha farklı bir tablo ile karşı karşıya kalıyoruz. Şekil - 2'de belirtildiği şekilde karşılaştırma ölçütü olarak zorlamaya hassaslık, doğruluk oranı, maliyet ve implementasyonu için gerekli efor.



Şekil - 2 Zephyr Analysis, Biyometrik Tekniklerin Karşılaştırılması [7]

Literatür taramasının ilk paragrafında belirtildiği üzere, kullanıcıların tuşlama ritmi verilerinin toplanması aşamalardan biri. Veri setinin oluşturulabilmesi için dikkat edilecek nicel noktalar ise tuşa basıldığı zaman, tuşun basılı kaldığı süre, diğer tuşa geçiş süresi (diğer tuş "key down event"e kadar) . Bahsedilen parametreler Şekil - 3'de görülmektedir.



Şekil - 3 Keystroke ölçüm parametreleri [8]

Tuşlama ritminde dikkate alınan parametrelerin formülasyonu:

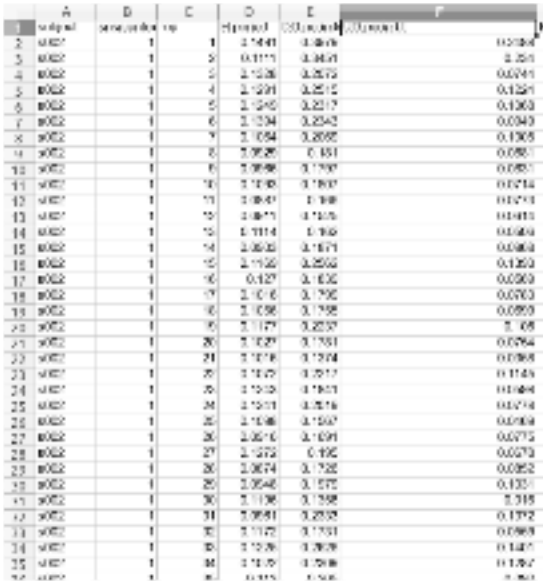
- Tek bir tuşun basılı kalma süresi (HT)
- Tuş 1 basmayı bırakma(release) ile Tuş 2 basmaya başlama arasındaki süre. (UD)
- Tuş 1 basım başlangıç süresinden Tuş 2 basımına başlangıç süresine geçen süre. (DD) Yani HT + UD = DD
- Hangi el ile basıldığının tahmini. (FA)

Bu parametreler tuşlanan her bir tuş için toplanmaktadır. Daha sonra kullanılan yönteme göre diğer tuşlama ritimleri ile karşılaştırılıp ortalaması veya öğrenme seti oluşturulur.

3. Materyal ve Metot

3.1 Veri Seti Yapısı

Veri seti olarak Carnegie Mellon University açık kaynak keystroke veri seti [9] kullanıldı. Veri seti 51 deneğin her birinin 10 karaktere sahip ".tie5Roanl" şifresini yazdığı ritim verilerine sahip. Her denek 8 farklı oturumda 50 tekrar ile tuşlama yapmış. Veri setinde denek no, oturum no, tekrar sayısı ve keystroke tuşlama parametreleri yer almaktadır. Şekil - 4 veri setinin ufak bir kısmını temsil etmektedir.



Şekil - 4 CMU Keystroke veri setinden bir kesit

3.2 Veri Setinin Anomali Karşılaştırması

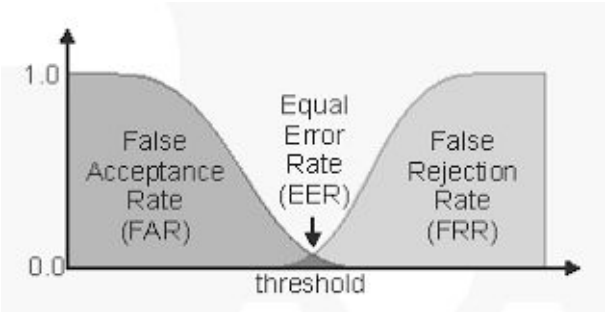
Veri setinde çeşitli anomaliler bulunmaktadır. Bu anomalileri tespit etmek, ayrıştırmak ve oranlarını bulmak için anomali tespit edici metodlar veya sınıflandırma metodları kullanılabilir. Biyometrik veri anomalileri gürültüden farklıdır. Gürültü istenmeyen veri, kirli veri olarak nitelendirilebilir iken anomali benzersiz, farklı, uç değer olarak değerlendirilebilir. Argümandan yola çıkarak anomalilerin veri setinde tespit edilmesinin taşıdığı önem fark edilebilir.

[9] kaynakta belirtilen anomali tespit metodları derlenip çalıştırıldığı takdirde elde edilen sonuçlar karşılaştırılırsa Şekil - 5 'deki gibi bir tablo karşımıza çıkar.

Detector	Average Equal-Error Rate (stddev)
Manhattan (scaled)	0.0962 (0.0694)
Nearest Neighbor (Mahalanobis)	0.0996 (0.0642)
Outlier Count (*z*-score)	0.1022 (0.0767)
SVM (one-class)	0.1025 (0.0650)
Mahalanobis	0.1101 (0.0645)
Mahalanobis (normed)	0.1101 (0.0645)
Manhattan (filter)	0.1360 (0.0828)
Manhattan	0.1529 (0.0925)
Neural Network (auto-assoc)	0.1614 (0.0797)
Euclidean	0.1706 (0.0952)
Euclidean (normed)	0.2153 (0.1187)
Fuzzy Logic	0.2213 (0.1051)
k Means	0.3722 (0.1391)
Neural Network (standard)	0.8283 (0.1483)

Şekil - 5 Anomali Tespit Metodlarının Belirtilen Veri Seti için Karşılaştırılması

"Average Equal-Error Rate" (EER) ortalama eşit hata oranı anlamına gelmekte. EER ise False Acceptance Rate ve False Rejection Rate'in ortalamasıdır(Şekil - 6). Yani hatalı kabul ve hatalı ret oranının ortalaması. Kontrol mekanizması, kimlik kontrolü yaparken hatalı ret yapabileceği gibi hatalı kabul etmesi de olasıdır.



Şekil - 6 Equal Error Rate ile FAR, FRR ilişkisi

3.3 LSTM Algoritması ile Uygulanışı

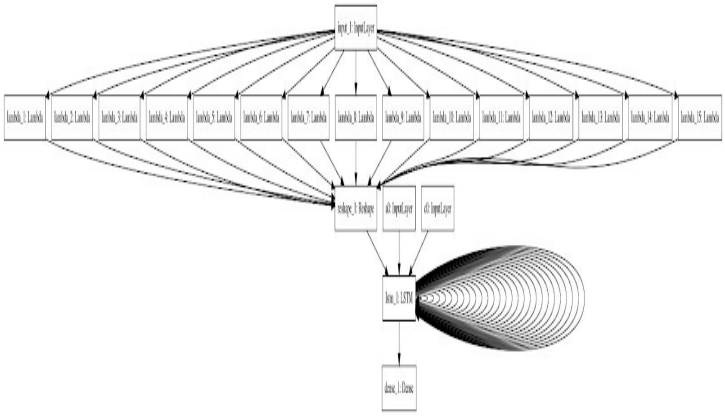
Long Short-Term Memory (LSTM), diğer bir adıyla "Uzun Kısa Vadeli Hafıza" uzun vadeli bağımlılıkları öğrenebilen özel bir RNN türüdür. Bunlar Hochreiter & Schmidhuber (1997) tarafından tanıtıldı [11]. LSTM'ler, uzun vadeli bağımlılık sorununun önüne geçmek için açıkça tasarlanmıştır. Çıkış noktası temel olarak "deep neural networkler" eğitilirken geri-yayılım (backpropagation) algoritmasının kullanımıyla ortaya çıkan "hataların katlanarak büyümesi" problemine çözüm üretmektir. RNN hücresine bir de hafıza eşlik eder. Her adımda öğrenilen hücrelerden hangilerinin hafızada tutulacağı hangilerinin atılmasına gerektiğine, hangilerinin güncelleneceğine karar verilir [12].

LSTM algoritmasının uygulanmasında takip edilen aşamalar 3 ana bölüme ayrılabilir. LSTM uygulanışı için [14] kaynaktaki kodlama yöntemi takip edilmiş olup deney, eğitim ve debugging aşamaları için "Jupyter Notebook" kullanılmıştır. Birinci aşamada veri seti gürültüden ve hatalardan temizleniyor. Bunun sebebi birçok biyometrik veri seti içerisinde bir miktar gürültü barındırmaktadır. Gürültü ise, anomalinin aksine istenmeyen veri, kullanıma elverişsiz veri, hata olarak adlandırılabilir. Gürültü ilgi çekici değildir, gelecek çalışmalara fayda sağlamamaktadır. Veri setinden gürültünün ayrıştırılması konusunda LSTM'in dahili hazır metotları mevcuttur. Birinci aşamanın devamında ise arındırılmış veri seti LSTM'in "Input Layer"larına uygun hale getirilmek için hazırlanıyor. Bu aşamadaki işlemleri özetlemek gerekirse verinin parçalara ayırılması, alt gruplar/sınıflara atanması bulunmakta. Ardından veri seti RNN'e (Recurrent Neural Network) beslenmek için hazır oluyor. Daha sonra (eğer kullanılacaksa) "Numpy" kütüphanesi dahilindeki "Numpy Frame" dönüşümü gerçekleştiriliyor.

İkinci aşamada; arındırılmış ve hazır hale getirilmiş veri seti eğitim için bölümlere ayrılıyor ve LSTM Hücrelerine yükleniyor.

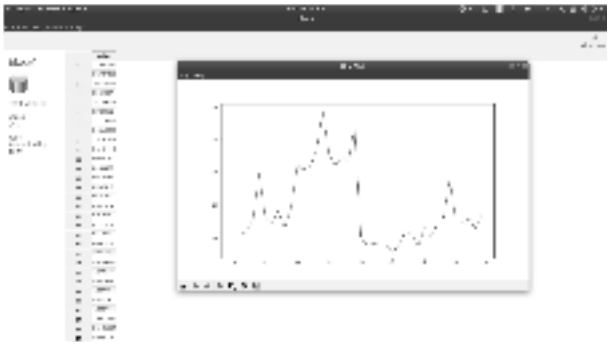
Eğitim fazının gerçekleşebilmesi için Keras kütüphanesinin bir parçası olan Lambda Layer'ları kullanmak önem arz ediyor. Lambda Katmakları birçok eğitim algoritmasında kullanılan bir yöntem. Katmanlar, daha sonra "ReShape" fonksiyonunda kullanılmak üzere oluşturuluyor. Ayrıca plot model oluşturulması için gerekli bir aşama. Ardından doğruluk ölçütü model üzerine parametre seçilerek oluşturuluyor. Artık eğitim için gerekli aşamalar tamamlandı, şimdi ise TensorFlow kütüphanesi yardımı ile "Epoch" iterasyonları yapılıyor. "Epoch" iterasyonları eğitim fazı için vazgeçilmezdir çünkü her iterasyonda, derin öğrenme belirtilen veri setini önceden belirlenmiş doğruluk ölçütüne göre daha doğru öğrenmekte. İterasyon sayısı öğrenme kalitesini ve doğruluğunu belirleyen temel bileşenlerden biridir.

Eğitim tamamlandıktan sonra üçüncü aşama olan kontrol ve sonuçlandırma aşaması mevcut. [9] kaynaktaki veri seti LSTM algoritması ile kodlandığı takdirde çıktı verilecek plot model Şekil - 7'de gösterilmiştir.



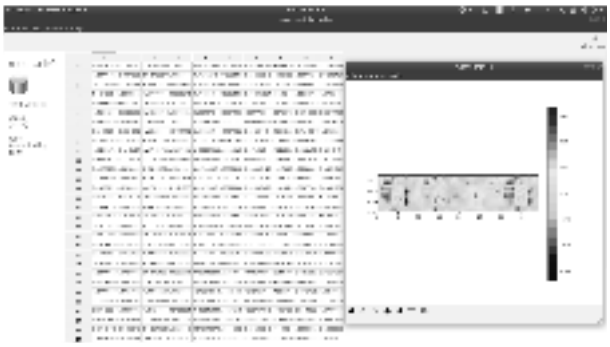
Şekil - 7 Keras'dan çıktı alınan Plot Model

Eğitim sonucu bir diğer çıktı ise deneklerin tuşlama ritmi eğilim/ağırlık grafikleridir. Çıktı verilen Format hdf5 formatında olduğu için, deney sonuçlarını inceleme işlemi "HDFCompass" adlı yazılım aracılığı ile yapılmakta. Çıktı sonuçlarında Bias grafiği, her bir denek için doğru kabul edilen 4 farklı tuşlama ve her deneğin yapmış olduğu tüm tuşlamaların ağırlık matrisi. Sonuçlar Şekil - 8 , Şekil - 9, Şekil - 10'da incelenmiştir.



Şekil - 8 Bias Line Plot grafiği

Şekil - 9 4 Her bir deneğin doğru kabul edilen tuşlama ağırlık matrisi



Şekil - 10 Tüm deneklerin tuşlama ritimlerini ağırlık matrisi

3.4 SVM Algoritmasının ile Uygulanışı

Sınıflandırma konusunda kullanılan oldukça etkili ve basit yöntemlerden birisidir. Sınıflandırma için bir düzlemde bulunan iki grup arasında bir sınır çizilerek iki grubu ayırmak mümkündür. Bu sınırın çizileceği yer ise iki grubun da üyelerine en uzak olan yer olmalıdır [13]. İşte SVM bu sınırın nasıl çizileceğini belirler.

SVM (Support Vector Machine) Keystroke dinamiklarinde anomali tespiti için kullanılabilecek yöntemlerden birisidir. Temel olarak makine öğrenmesine dayanır, doğal olarak da eğitim seti, eğitim ve kontrol aşamaları bulunur.

İmplementasyonu için [9] açık kaynak CMU Keystroke veri seti ve [14] açık kaynak temel anomali tespit algoritmaları kullanılmıştır. SVM'de, LSTM'de olduğu gibi tekrar veri temizleniyor, hazırlanıyor daha sonra hazır SVM eğitim metoduna gönderiliyor. Daha sonra eğitim sonuçları doğrultusunda her bir deneğin yapmış olduğu tuşlama ritmi doğrultusunda doğru/gerçek (genuine) kabul edilen ve taklit (imposter) tuşlama ritimleri sonuçlanıyor. Deney çıktısı olarak elde edilen EER değeri (3.2'de anlatıldığı gibi) terminal arayüzü ile bildiriliyor (Şekil - 11).

```
yke@emma-g:~/Downloads/keystroke-git/11$ python K
SVM icin ortalama EER:
0.12054244703221502
yke@emma-g:~/Downloads/keystroke-git/11$
```

Şekil - 11 SVM Algoritması İçin ortalama EER Değeri

EER değeri, bir algoritmanın hata oranı olarak kabul edilebilir. Bu doğrultuda SVM algoritmasının kullanılan veri seti için doğruluk oranı 100-0.12=99,18 olarak hesaplanır.

Aynı veri seti Manhattan Algoritması [14] için uygulanırsa alınacak sonuç ise Şekil - 12'de görülmektedir.

```
File  Edit  View  Search  Terminal  Help
yke@emma-g:~/Downloads/keystroke-git/11$ p
SVM icin ortalama EER:
0.12054244703221502
yke@emma-g:~/Downloads/keystroke-git/11$ p
Manhattan Algoritmasi icin ortalama EER:
0.18065765645731371
yke@emma-g:~/Downloads/keystroke-git/11$
```

Şekil - 12 Manhattan Algoritması İçin ortalama EER

4. Tartışma ve Sonuç

Bu çalışma kapsamında LSTM derin öğrenme algoritmasının ve SVM algoritmasının Keystroke dinamikleri biyometriği için uygulanmıştır. Uygulama yöntemi Python dili ve temel yapay zeka kütüphanelerini içermektedir. (Numpy, TensorFlow, Keras…) LSTM algoritmasının uygulama amacı tuş ritmi veri seti kullanılarak referans(doğru/gerçek) olarak kabul edilecek tuşlama ritimlerinin derin öğrenme aracılığı ile tespit edilmesi, daha sonra ritim ağırlık matrisi oluşturulması. Deney sonuçları incelendiği takdirde görülecektir ki her bireyin kendine has tuşlama ritmi bulunmakta, ve bu ritimler kişiden kişiye farklılık göstermektedir. Keystroke biyometrik verileri diğer biyometrik verilere karşın benzersizlik oranı her ne kadar düşük olsa da implementasyonu için gerekli bütçe ve efor göz önüne alındığında [7] fiyat-performans bakımından en verimli biyometrik sinyal olduğunu söylemek mümkün. Uygulaması için ekstra bir aşama, harici donanım gerektirmemesi de bir başka üstünlüğüdür. Algoritmalar veri setine bağlı olarak yüksek performans gösterebilirler. Yapay Sinir Ağları, Destek Vektör Makineleri gibi diğer makine öğrenmesi algoritmaları ile çalışma tekrarlanarak performans karşılaştırılması yapılabilir. Ayrıca, bu çalışmada önerilen model farklı veri setlerinde de sınanmalıdır.Uygulamalar için sabit uzunluktaki bir şifrenin seçilmesinin yanı sıra değişken uzunluktaki şifrelere göre de analizlerin yapılması ilerideki çalışmalarda incelenebilir. SVM algoritması uygulama deneyindeki sonuçtan çıkarabileceğimiz anlam ise derin öğrenme algoritması olmadan dahi tuşlama ritmi farklılıklarını gözetmek, ayrıştırmak mümkün. Derin öğrenme kapsamında olmayan algoritmalar her ne kadar kısa vadede daha iyi sonuçlar verse dahi, deney sonuçları ve literatür taraması incelendiğinde görülecektir ki uzun vadede en doğru sonucu verecek algoritmalar her zaman nöral ağ tabanlı sistemlerdir. Çünkü bu sistemler anlama/eğitim üzerine çalıştığı için kullanıcı ritminin değişimine ve zaman aşımına dayanıklıdır.

Çalışma çıktıları sonucunda giriş kısmındaki önerme doğrulanmış olup keystroke dinamiği biyometriğinin bilişim sistemlerindeki bilgi güvenliğine pozitif katkı sağladığı, birçok veri sızıntısı problemini önemsiz kılma özelliğine sahip olduğu görülebilmektedir.

Kaynakça

[1] Future of biometrics Bihttps://www.statista.com/chart/11122/the-future-of-mobile-biometrics/

[2] Future Generation Computer Systems 16 (2000) 351–359, Keystroke dynamics as a biometric for authentication.

[3] Young, Jay Richards, "Keystroke Dynamics: Utilizing Keyprint Biometrics to Identify Users in Online Courses" (2018).All Theses andDissertations. 6690. https://scholarsarchive.byu.edu/etd/6690

[4] International Journal of Computer Applications (0975 –8887)Volume 144 –No.9, June 2016 Keystroke Dynamics for User Authentication and Identification by using Typing Rhythm

[5] J. Ilonen, "Keystroke dynamics", Adv. Top. Inf. Process., ss. 03–04, 2003.

[6] 2015 International Conference on Communication, Information & Computing Technology (ICCICT), Jan. 16-17, Mumbai, India
978-1-4799-5522-0/15/$31.00 ©2015 IEEE
Password Authentication using Keystroke Biometrics

[7] The Body As A Password:Considerations, Uses, And Concerns Of Biometric Technologies by Michelle C. Frye, B.A.

[8] Antal M., Nemes L. (2016) The MOBIKEY Keystroke Dynamics Password Database: Benchmark Results. In: Silhavy R., Senkerik R., Oplatkova Z., Silhavy P., Prokopova Z. (eds) Software Engineering Perspectives and Application in Intelligent Systems. CSOC 2016. Advances in Intelligent Systems and Computing, vol 465. Springer, Cham

[9] Kevin S. Killourhy and Roy A. Maxion. "Comparing Anomaly Detectors for Keystroke Dynamics," in Proceedings of the 39th Annual International Conference on Dependable Systems and Networks (DSN-2009), pages 125-134, Estoril, Lisbon, Portugal, June 29-July 2, 2009. IEEE Computer Society Press, Los Alamitos, California, 2009.

[10] Teh PS, Teoh AB, Yue S. A survey of keystroke dynamics biometrics. ScientificWorldJournal. 2013;2013:408280. Published 2013 Nov 3. doi:10.1155/2013/408280

[11] https://medium.com/@ishakdolek/lstm-d2c281b92aac

[12] Yapay Sinir Ağları, Kelime Vektörleri Ve Derin Öğrenme Uygulamaları, Yrd. Doç. Dr. Ebubekir KOÇ.

[13] http://bilgisayarkavramlari.sadievrenseker.com/2008/12/01/svm-support-vector-machine-destekci-vektor-makinesi/

[14] https://github.com/kuntojirohan/key-stroke-dynamics

[15] Makale IEEE ISEC 2011 Two Column Formatına göre hazırlanmıştır. ewh.ieee.org/conf/stem/restored/ISEC_format.doc