SİBER GÜVENLİKTE VERİ MADENCİLİĞİ

ÖDEV-3 RAPOR

Büşra Kizilaslan

Hasan Emir Kara

Kyoto 2006+ veri kümesi, başlangıçta 2006 Kasım'dan 2009 Ağustos'a kadar gerçek trafik verilerinin üç yılının kullanıldığı bir siber güvenlik veri kümesidir. Daha sonra 2006 Kasım'dan 2015 Aralık'a kadar ek verilerle güncellenmiştir. Bu veri kümesi genellikle İzinsiz Giriş Tespit Sistemleri (IDS) ve ağ güvenliği alanındaki araştırma Impact of PDS Based kNN Classifiers on Kyoto Dataset (PDS Tabanlı kNN Sınıflandırıcılarının Kyoto Veri Kümesine Etkisi) Yazarlar Kailasam Swathi,Bobba Baseveswara Rao tarafından International Journal of Rough Sets and Data Analysis dergisinde yayınlanmıştır. Bu makalenin de Ağ Saldırı Tespit Sistemleri (NIDS) için Kyoto 2006+ veri kümesi ile Kısmı Mesafe Arama (PDS) tabanlı kNN sınıflandırıcısı performans açısından karşılaştırılmaktadır. PDS özellik indekslemesine göre adlandırılmaktadır.

Bunlar:

i) Basit PDS kNN, özellikler indekslenmemiş (SPDS),

ii) Varyans indekslemeye dayalı kNN (VIPDS), özellikler, özelliklerin varyansına göre indekslenir ve

iii) Korelasyon katsayısı indekslemeye dayalı kNN (CIPDS)), özellikler bir sınıf etiketiyle özelliklerin korelasyon katsayısına göre indekslenir. (Swathi, Kailasam, and Bobba Basaveswara Rao. "Impact of PDS based kNN classifiers on Kyoto dataset." International Journal of Rough Sets and Data Analysis (IJRSDA) 6.2 (2019): 61-72.).

Karşılaştırma hesaplama süresi ve doğruluğu performans ölçüsü olarak kabul edilmiştir. Yapılan deney sonucunda hesaplama süresi açısında en iyi performansa sahip olan CIPDS, doğruluk açısından bakıldığında burada da VIPDS ön planda ancak CIPDS ile karşılaştırıldığında aradaki farkın önemli olmadığı öngörülmüştür.

Literatür taramasından elde ettiğimiz yukarıdaki makaleyi ele aldığımızda Kyoto2006+ datasetimiz için en uygun algoritmanın K-Nearest Neighbor algoritması olduğuna karar verdik.

Peki neden K-Nearest Neighbor:

KNN algoritması, basitliği nedeniyle en çok kullanılan öğrenme algoritmalarından biridir. en yakın komşulara göre veri noktalarını sınıflandıran bir makine öğrenimi yöntemidir.

KNN algoritması, hem sınıflandırma hem de regresyon problemlerinde kullanılabilir. Sınıflandırmada, k veri noktasının sınıflarına göre oylama yapılır. Regresyonda, k veri noktasının değerlerinin ortalaması alınır. KNN algoritması, basit, esnek ve sezgisel bir yöntemdir. Ancak, k parametresinin belirlenmesi, verilerin ölçeklenmesi, verilerin yüksek boyutluluğu ve gürültüsü gibi bazı zorluklar da vardır.

Basitliğine rağmen KNN, diğer güçlü sınıflandırıcılardan çok daha iyi çalışır ve ekonomik tahmin ve veri sıkıştırma, Video Tanıma, Görüntü Tanıma, El Yazısı Algılama ve Konuşma Tanıma gibi yerlerde kullanılır.

KNN algoritmasının siber güvenlik açısından önemi:

Siber güvenlik açısından önemi, verileri analiz etmek, saldırıları tespit etmek, zararlı yazılımları sınıflandırmak ve güvenli iletişim kurmak gibi çeşitli uygulamalarda kullanılabilmesidir. yöntemidir. Siber güvenlik açısından önemi, verileri analiz etmek, saldırıları tespit etmek, zararlı yazılımları sınıflandırmak ve güvenli iletişim kurmak gibi çeşitli uygulamalarda kullanılabilmesidir.

KNN algoritmasının çalışma mantığı şu şekildedir:

- Öncelikle,komşu sayısını belirleyen bir k parametresi seçilir.
- Daha sonra, sınıfı belirlenmek istenen veri noktasına en yakın olan k veri noktası bulunur. Bu noktaların uzaklıkları, Öklid, Manhattan, Minkowski gibi farklı metriklerle hesaplanabilir.
- Ardından, k veri noktasının hangi sınıflara ait olduğu sayılır ve en çok sayıda olan sınıf, yeni veri noktasının sınıfı olarak atanır.

KNN algoritması, hem sınıflandırma hem de regresyon problemlerinde kullanılabilir. Sınıflandırmada, k veri noktasının sınıflarına göre oylama yapılır. Regresyonda, k veri noktasının değerlerinin ortalaması alınır. KNN algoritması, basit, esnek ve sezgisel bir yöntemdir. Ancak, k parametresinin belirlenmesi, verilerin ölçeklenmesi, verilerin yüksek boyutluluğu ve gürültüsü gibi bazı zorluklar da vardır.

KNN’nin avantajları:

- Hızlı hesaplama
- Basit algoritma-yorumlamak için
- Çok yönlü-sınıflandrıma ve regresyon için kullanışlıdır

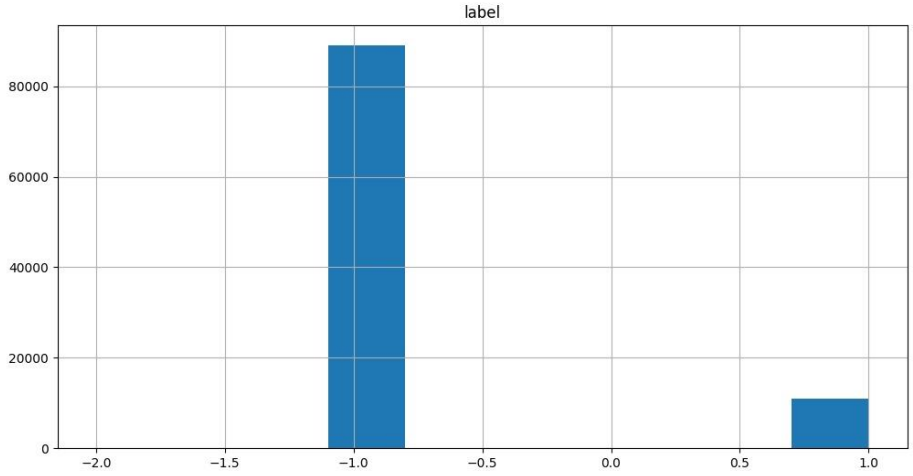
- Yüksek doğruluk
- Veriler hakkında varsayım yok -ek varsayımlar yapmaya veya bir model oluşturmaya gerek yok.

KNN'nin dezavantajları:

- Doğruluk,verilerin kalitesine bağlıdır.
- Tahmin,büyük verilerle yavaşlar
- Büyük veri kümeleri için uygun değil
- Tüm eğitim verilerini saklama ihtiyacı,bu nedenle yüksek bellek gerektirir.
- Tüm eğitmi sakladığı için hesaplama açısından pahalı olabilir.

```
df[['label']].hist(bins=10, figsize=(12, 6))
```

label görsel grafiği:




```
# modeller için gerekli kütüphaneler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris
```

```
# datayı yüklemek
irisData = load_iris()

#  feature ve target array oluşumu
df = irisData.data
df_test = irisData.target

knn = KNeighborsClassifier(n_neighbors=7)

knn.fit(df, df_test)

# dataset üzerinde daha önce görülmemiş modeli tahmin etmek
print(knn.predict(df))

# Modelin doğruluğunun hesaplanması
print(knn.score(df, df_test))

neighbors = np.arange(1, 9)
train_accuracy = np.empty(len(neighbors))
test_accuracy = np.empty(len(neighbors))

# Loop over K values
for i, k in enumerate(neighbors):
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(df, df_test)

 # Training ve test datasının doğrulugunun hesaplanması
  train_accuracy[i] = knn.score(df, df_test)
```

Generate plot - plt.plot(neighbors, test_accuracy, label = 'Testing dataset Accuracy')

plt.plot(neighbors, train_accuracy, label = 'Training dataset Accuracy')

plt.legend()

plt.xlabel('n_neighbors')

plt.ylabel('Accuracy')

plt.show()

Training dataset Accuracy