**Ujian Tengah Semester - Data Jaringan**

Dosen Pengampu : Vinna Rahmayanti Setyaning Nastiti S.Si., M.Si



Disusun oleh :

Thoriq Hermansya Rahman 202010370311492

**UNIVERSITAS MUHAMMADIYAH MALANG**

**FAKULTAS TEKNIK**

**INFORMATIKA**

**2023**

1. **Explanation about extraction of dataset. Why and what data that you use from PCAP dataset.**

: Tentang dataset. Dataset network attack yang digunakan adalah dataset yang dikhususkan untuk kelas Big Data Analyst 7A. Dataset ini diambil dan dikumpulkan memberi informasi tentang keamanan siber. Proses ekstraksi ini penting atas beberapa alasan, dan data yang diekstraksi dari dataset serangan jaringan umumnya mencakup hal berikut :

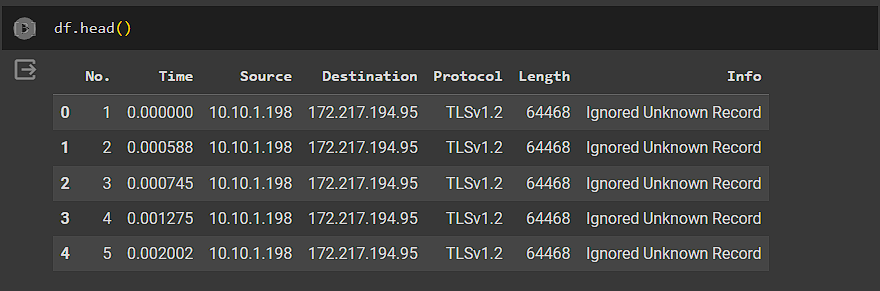
* Pola dan Tanda Serangan : Ekstraksi data terkait pola dan tanda serangan yang dikenal membantu mengidentifikasi karakteristik unik dari berbagai jenis serangan jaringan. Informasi ini penting untuk mengembangkan mekanisme deteksi dan pencegahan yang efektif.
* Log Kejadian : Ekstraksi data dari log kejadian menyediakan catatan kronologis tentang insiden keamanan dan rinciannya. Ini mencakup informasi tentang sifat serangan, sistem yang terpengaruh, dan metode yang digunakan oleh penyerang.
* Sampel Malware : Ekstraksi data terkait malware, termasuk potongan kode dan pola perilaku, membantu memahami cara kerja malware. Informasi ini penting untuk mengembangkan tanda tangan antivirus dan meningkatkan kemampuan deteksi malware.
* Log Aktivitas Pengguna : Ekstraksi data tentang aktivitas pengguna dalam jaringan memberikan wawasan tentang perilaku normal. Setiap deviasi dari pola ini dapat menjadi indikator potensi pelanggaran keamanan, memungkinkan deteksi dan respons yang cepat.

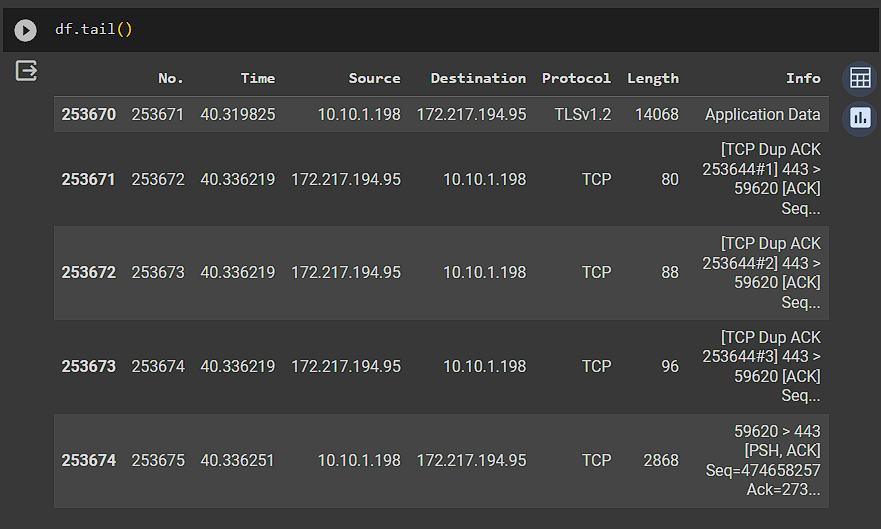
Ekstraksi dataset dari sumber serangan jaringan penting untuk meningkatkan postur keamanan siber suatu organisasi. Ini memungkinkan pendekatan proaktif terhadap deteksi ancaman, respon terhadap insiden, dan perbaikan berkelanjutan terhadap langkah-langkah keamanan untuk tetap berada di depan ancaman siber yang terus berkembang.

1. **Your dataset sample (head or tail dataset)**

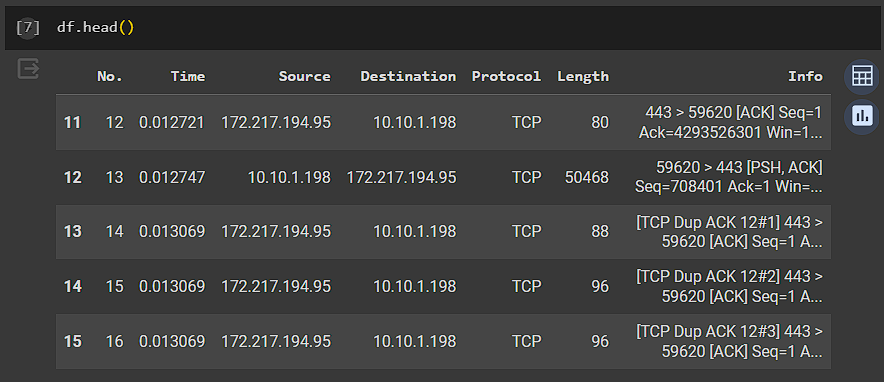
Kami menampilkan dataset yang belum dan setelah dilakukan preprocessing

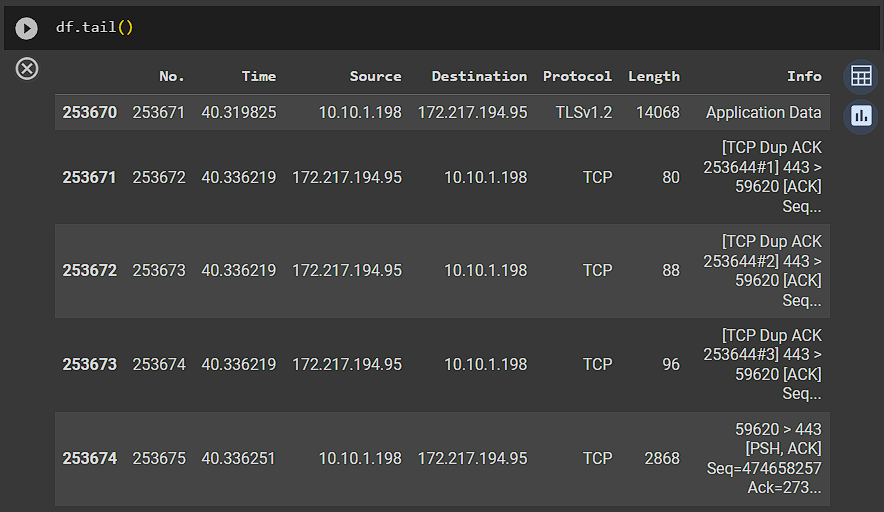
* 1. Head and Tail before Preprocessing





* 1. Head and Tail after Preprocessing





Deskripsi fitur :

1. No : Nomor data
2. Time : Waktu pengambilan sampel
3. Source : Alamat IP sumber yang mengirimkan paket yang dapat berasal dari pihak yang bersalah
4. Destination : Alamat IP tujuan yang menerima paket dari pengirim
5. Protocol : Protokol jaringan yang terserang oleh pihak yang bersalah
6. Length : Panjang paket
7. Info : Informasi aktivitas yang terjadi
8. **Explanation about modeling (preprocessing until evaluation)**
   1. **Preprocessing**
      1. Menghapus nilai nan pada seluruh fitur / atribut

|  |
| --- |
| df = df.dropna() |

* + 1. Membuat data yang dikhususkan untuk berisi hanya yang memiliki nilai pada fitur ‘Info’ (diasumsikan dalam fitur ‘Info’ yang bernilai ‘Ignored Unknown Record’ berarti nilainya kosong)

|  |
| --- |
| df = df.loc[df['Info'] != 'Ignored Unknown Record'] |

* + 1. Melakukan Label Encoding setelah feature selection

|  |
| --- |
| label\_encoder = LabelEncoder()  X['Source'] = label\_encoder.fit\_transform(X['Source'])  X['Destination'] = label\_encoder.fit\_transform(X['Destination'])  X['Protocol'] = label\_encoder.fit\_transform(X['Protocol'])  y = label\_encoder.fit\_transform(y) |

* 1. **Modeling**

Pada tugas kali ini kami menggunakan XGBoost (eXtreme Gradient Boosting) karena algoritma ini kuat dan populer dalam machine learning yang dapat digunakan untuk tugas regresi, klasifikasi, dan perankingan. XGBoost Regressor sering digunakan untuk prediksi regresi karena kombinasi kecepatan, kualitas prediksi yang baik, dan kemampuan mengatasi berbagai tantangan yang mungkin dihadapi dalam data regresi.

* + 1. Inputing Library

|  |
| --- |
| from sklearn.metrics import r2\_score  from sklearn.metrics import mean\_squared\_error  from sklearn.metrics import mean\_absolute\_error  from sklearn.metrics import mean\_squared\_error  import math  from xgboost import XGBRegressor |

* + 1. Pembuatan model

|  |
| --- |
| def score(model):  modell=model()  print('Model:',model)  print('-'\*20)  modell.fit(X\_train,y\_train)  preds=modell.predict(X\_test)  # print("RMSE :",np.sqrt(mean\_squared\_error(y\_test,preds)))  # print('r2\_score:',r2\_score(y\_test,preds))  print("Mean absolute error (MAE): %f"%mean\_absolute\_error(y\_test, preds))  print("Mean Squared error (MSE): %f"%mean\_squared\_error(y\_test, preds))  print("Roots Mean Squared error (RMSE): %f"%math.sqrt(mean\_squared\_error(y\_test, preds)))  print("R2 Score: %f"%r2\_score(y\_test, preds))  plt.figure(figsize=(7,6))  plt.scatter(x=y\_test,y=preds,color='Blue')  plt.title('Actuals vs Predictions',fontsize=14)  plt.xlabel('Actuals',fontsize=14)  plt.ylabel('Predictions',fontsize=14) |

* + 1. Hasil dari model

|  |
| --- |
| score(XGBRegressor) |
| --------------------  Mean absolute error (MAE): 47458.282851  Mean Squared error (MSE): 3167971699.801758  Roots Mean Squared error (RMSE): 56284.737716  R2 Score: 0.353743 |
|  |

dari gambar diatas menunjukkan Hasil evaluasi model XGB Regressor dalam konteks regresi. MAE, MSE, dan RMSE memberikan informasi tentang seberapa besar kesalahan prediksi secara keseluruhan, sementara R2 Score memberikan gambaran tentang sejauh mana model dapat menjelaskan variasi dalam data. Dalam kasus ini, evaluasi menunjukkan bahwa model dapat melakukan prediksi, tetapi ada ruang untuk perbaikan karena nilai-nilai kesalahan yang cukup tinggi.

1. **Conclusion and recommendation**

Dari hasil yang didapat kami melakukan analisa terhadap setiap record aktivitas jaringan yg masuk. adapun hasil MAE sebesar 47458.28%, MSE sebesar 3167971699.80%, RMSE sebesar 56284.73% dan r2 score sebesar 35%. Untuk skor evaluasinya cukup tinggi dikarenakan pada proses training model untuk penskalaan nya tidak sama rata.

1. **Source code**

Link github : [thoriqrhmn/UTSBigData (github.com)](https://github.com/thoriqrhmn/UTSBigData)

Link drive : <https://drive.google.com/drive/folders/17tV1nhPsi_dlfeCEfGSIJ0s-8NL5lwzV?usp=drive_link>

**JOURNAL REFERENCE that you used (link journal and screenshot the title, author, and abstract)**

* 1. Link Paper : <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11746/117460V/Machine-learning-raw-network-traffic-detection/10.1117/12.2586114.short?SSO=1>
  2. Title : Machine Learning for Raw Network Traffic Detection
  3. Author : Michael J. De Lucia , Paul Maxwell , Nathaniel D. Bastian , Ananthram Swami , Brian Jalaian , Nandi Leslie
  4. Abstract :

Increasingly cyber-attacks are sophisticated and occur rapidly, necessitating the use of machine learning techniques for detection at machine speed. However, the use of traditional machine learning techniques in cyber security requires the extraction of features from the raw network traffic. Thus, subject matter expertise is essential to analyze the network traffic and extract optimum features to detect a cyber-attack. Consequently, we propose a novel machine learning algorithm for malicious network traffic detection using only the bytes of the raw network traffic. The feature vector in our machine learning method is a structure containing the headers and a variable number of payload bytes. We propose a 1D-Convolutional Neural Network (1D-CNN) and Feed Forward Network for detection of malicious packets using raw network bytes. Additionally, we compare the performance of the proposed deep learning models with both a non-linear and linear Support Vector Machine and Multinomial Naïve Bayes machine learning models. We leverage the UNSW-NB15 dataset for evaluation of our novel classifier using raw network traffic. Subsequently, the UNSW-NB15 packet captures are labeled based on correlation between a hash of the standard 5-tuple and time stamp in the CSV file containing the labels (viz., malicious or benign). Our novel 1D-CNN classifier achieves an accuracy and F1 score of 98.99%. Thus, our method demonstrates the utility of using raw network traffic coupled with machine learning and reduces the need for a subject matter expert to perform feature engineering.