Data Preparation + ML Research

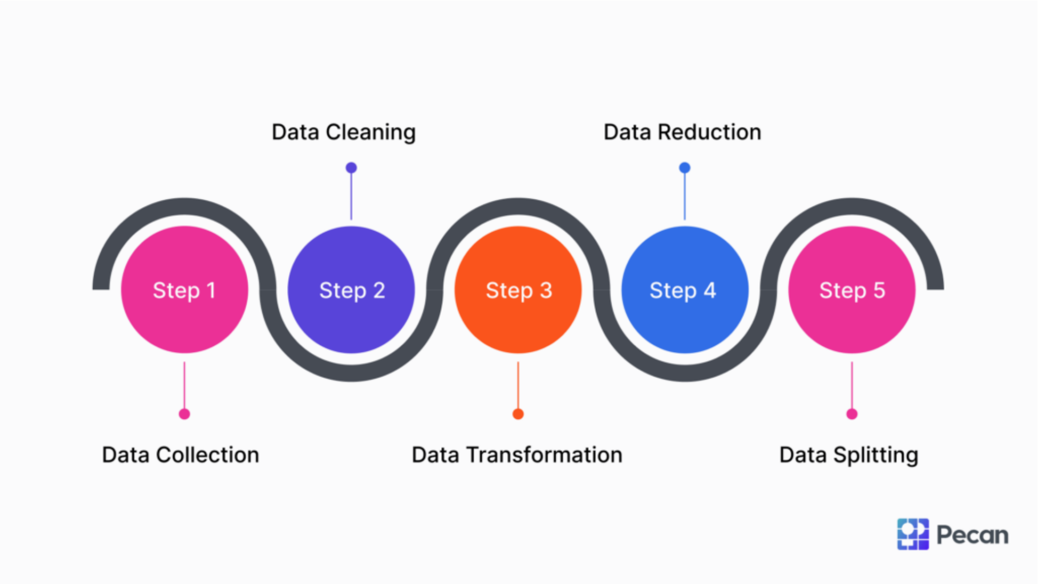
Conversion to CVS file

- CSV = structured table format, ready for data manipulation
    - pyshark, scapy or dpkt library

Cleaning Up data

- Used to clean up redundant data for the ML algorithm to understand and offer accurate results
- On-going process, might need refinement over time
- Cleaned up data saved as another CSV file
    - pandas and numpy libraries

Pipeline:



Data Collection:

- User uploads PCAP file
- PCAP file gets converted into CSV format

Data Cleaning:

1. Missing values (zero values) -> e.g. no TCP flags on UDP packets
2. Outliers (unusual values) -> e.g. very high values could indicate malicious traffic
3. Inconsistencies -> e.g. impossible values

Data Transformation:

- Feature scaling (e.g. min-max scaling/standardisation) -> evens out x/y axis, separates data into classes

    - sklearn -> StandardScaler, QuantileTransformer, LogisticRegression and PolynomialFeatures via pipeline
  - Encoding (transforming non-numeric values to numeric representation e.g. protocols ‘TCP’, ‘UDP’, flags ‘SYN’, ‘ACK’ etc.)
    - sklearn -> OneHotEncoder

Data Reduction:

- Simplify data to spot patterns -> speeds up ML
- Remove irrelevant columns (e.g. timestamps, raw IPs)
- Dimensionality reduction

Data Splitting:

- Split data into different sets, not needed for unsupervised ML

ML

- Model is trained ONCE, saved to a .pkl or .joblib file and then used when user uploads PCAP file
- The PCAP file still needs to run through cleaning pipeline, then cleaned features get fed into the model which outputs predictions

Types of ML:

- Supervised ML: Model is trained using labeled data, such as normal/malicious
  - Pros: accurate in predicting exact attacks
  - Cons: must have labeled attack PCAPs, won’t detect new attacks
- Unsupervised ML: Model is trained only on normal traffic
  - Pros: Only needs normal PCAPs, detects any traffic that looks abnormal
  - Cons: Doesn’t label specific attacks, sometimes inaccurate

➔ For our project, unsupervised ML fits the best

ML Training Process:

1. Load cleaned data from CSV file
2. Pick ML algorithm e.g. RandomForestClassifier
3. Train the model on the dataset
4. Evaluate by silhouette score, visualisation or manually
5. Store model

References

https://github.com/AssemblyAI/youtube-tutorials/blob/main/Data%20preparation%20and%20model%20training.ipynb

https://www.pecan.ai/blog/data-preparation-for-machine-learning/

https://youtu.be/0B5eIE_1vpU?list=PL99XjA3eOWgC9dhqaU4U6QroFjDXquXQL