Cyber Physical Systems Security 12th Project

Participants:

- Mohamed Mostafa Sayed Saber Ali **221027679**
- Mohamed Hany El Kordy **221027552**
- Basel Hany Abbas **221027635**
- Mohamed Basem **221027617**
- Mohamed Mostafa Abd El Baset **221027785**
- Omar Hesham **221027739**

Table of Contents:

Project Objectives

- **The main goal of this project** is to **design** and **implement** an **AI-based Intrusion Detection System (IDS) for IoT networks. IoT devices** and **Cyber-Physical Systems (CPS)** face unique **security challenges** due to their **limited computational resources**, **heterogeneous devices**, and **large attack surface**. **Traditional rule-based security mechanisms** often **fail to detect sophisticated or evolving attacks.** Therefore, **this project focuses on developing an intelligent and adaptive IDS using machine learning and deep learning techniques.**

- The specific objectives of this project are:
    - **Understand Security Threats in IoT-based CPS Environments**
        - Identify **common vulnerabilities** in **IoT devices** and **networks**.
        - Analyze the types of **cyber-attacks** that **IoT systems** are **susceptible to**, such as **Denial-of-Service (DoS), Man-in-the-Middle (MITM), Eavesdropping, and Data Injection Attacks.**
    - **Study AI-based Intrusion Detection Techniques**
        - Explore machine learning and deep learning approaches for detecting anomalies and attacks in network traffic.

        - Evaluate the suitability of different algorithms for real-time IoT network security.
    - **Design and Evaluate an AI-based IDS Model**
        - **Develop** a **deep learning model** that can classify **network traffic** as **normal** or **malicious**.
        - **Preprocess** and **scale IoT network datasets** for **model training**.
        - **Evaluate** the **effectiveness** of the **model** using **metrics** such as **accuracy**, **precision**, **recall**, and **F1-score**.
        - Analyze the **performance**, **limitations**, and **potential improvements** for **real-world deployment.**

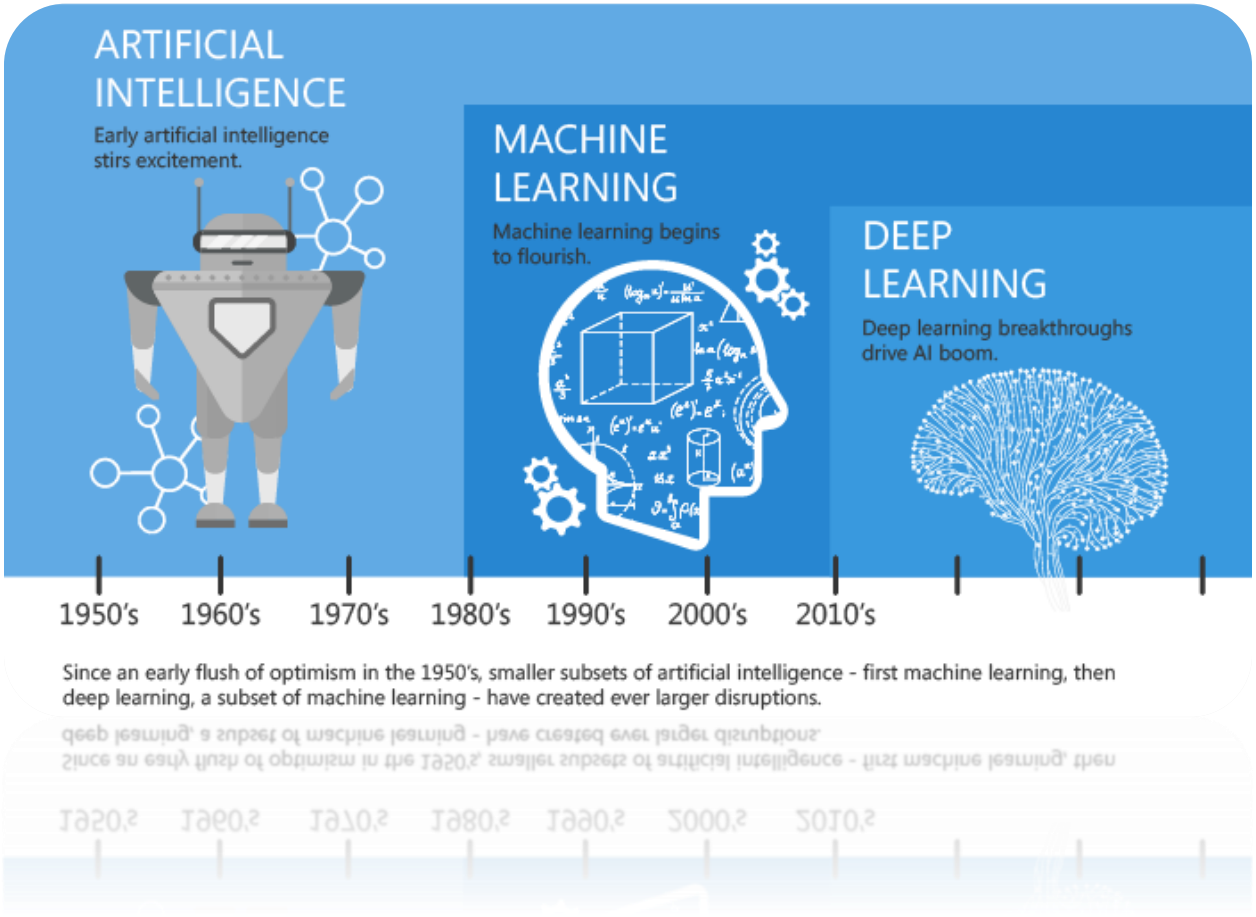
Literature Review

- **Intrusion Detection Systems (IDS)** play a critical role in **securing IoT networks**, which are increasingly targeted by **cyber-attacks** due to their widespread deployment and limited device resources. This section summarizes existing research and approaches relevant to AI-based IDS for IoT networks.

- **IDS in IoT Networks:**
  - **IoT networks** consist of **heterogeneous devices communicating** over **various protocols**, which introduces **unique security challenges.**
  - **Traditional IDS approaches**, such as **signature-based** or **rule-based systems**, **often fail** to **detect new** or **evolving attacks**.
  - **AI-based IDS** leverage intelligent algorithms to analyze network traffic and detect anomalies or malicious behavior in real-time.

- **Machine Learning and Deep Learning for IDS:**
  - **Machine Learning (ML) techniques**, such as **Decision Trees**, **Random Forests**, and **Support Vector Machines**, can classify network traffic based on statistical patterns.
  - **Deep Learning (DL) approaches**, such as **Artificial Neural Networks (ANNs), Convolutional Neural Networks (CNNs),** and **Recurrent Neural Networks (RNNs),** can **automatically extract features** from **complex data** and **achieve higher accuracy** in **detecting unknown attacks.**

- AI-based IDS models can adapt to changing network conditions and evolving attack patterns, which is essential for IoT security.



- **Common IoT Attacks:**
  - **IoT networks** are **susceptible** to **several types** of **cyber-attacks:**
    - **Denial-of-Service (DoS):** Flooding the network or devices with excessive requests, causing them to crash or become unresponsive.
    - **Man-in-the-Middle (MITM):** Intercepting and potentially altering communications between IoT devices.
    - Eavesdropping: Unauthorized monitoring of data transmissions.
    - **Data Injection Attacks:** Injecting malicious or false data into IoT systems to manipulate or disrupt operations.
    - **Malware-based attacks (e.g., Mirai):** Infecting IoT devices to create botnets or launch coordinated attacks.

OWASP TOP 10 INTERNET OF THINGS 2018

1 **Weak, Guessable, or Hardcoded Passwords**
Use of easily bruteforced, publicly available, or unchangeable credentials, including backdoors in firmware or client software that grants unauthorized access to deployed systems.

2 **Insecure Network Services**
Unneeded or insecure network services running on the device itself, especially those exposed to the internet, that compromise the confidentiality, integrity/authenticity, or availability of information or allow unauthorized remote control...

3 **Insecure Ecosystem Interfaces**
Insecure web, backend API, cloud, or mobile interfaces in the ecosystem outside of the device that allows compromise of the device or its related components. Common issues include a lack of authentication/authorization, lacking or weak encryption, and a lack of input and output filtering.

4 **Lack of Secure Update Mechanism**
Lack of ability to securely update the device. This includes lack of firmware validation on device, lack of secure delivery (un-encrypted in transit), lack of anti-rollback mechanisms, and lack of notifications of security changes due to updates.

5 **Use of Insecure or Outdated Components**
Use of deprecated or insecure software components/libraries that could allow the device to be compromised. This includes insecure customization of operating system platforms, and the use of third-party software or hardware components from a compromised supply chain.

6 **Insufficient Privacy Protection**
User's personal information stored on the device or in the ecosystem that is used insecurely, improperly, or without permission.

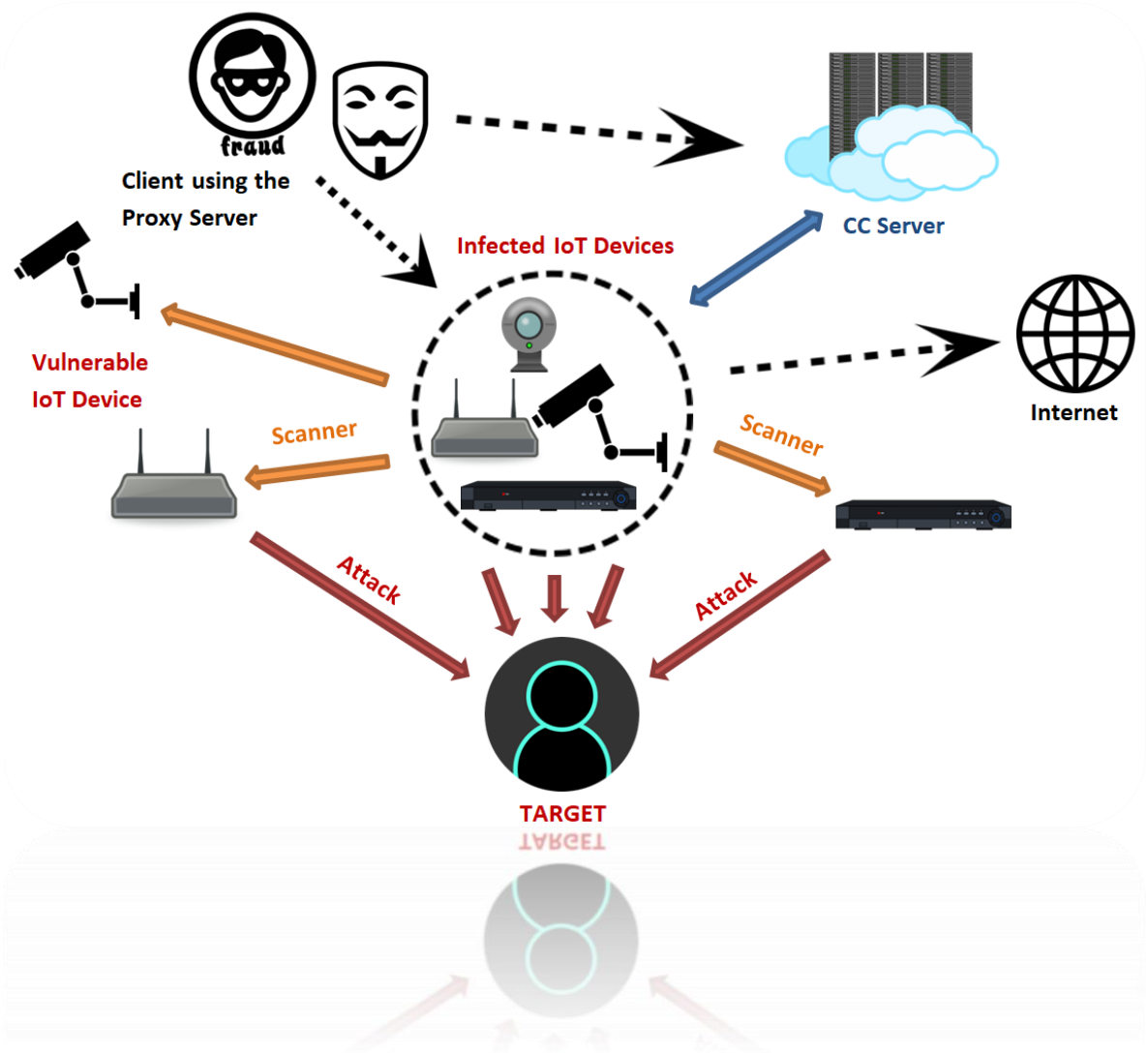
7 **Insecure Data Transfer and Storage**
Lack of encryption or access control of sensitive data anywhere within the ecosystem, including at rest, in transit, or during processing.

8 **Lack of Device Management**
Lack of security support on devices deployed in production, including asset management, update management, secure decommissioning, systems monitoring, and response capabilities.

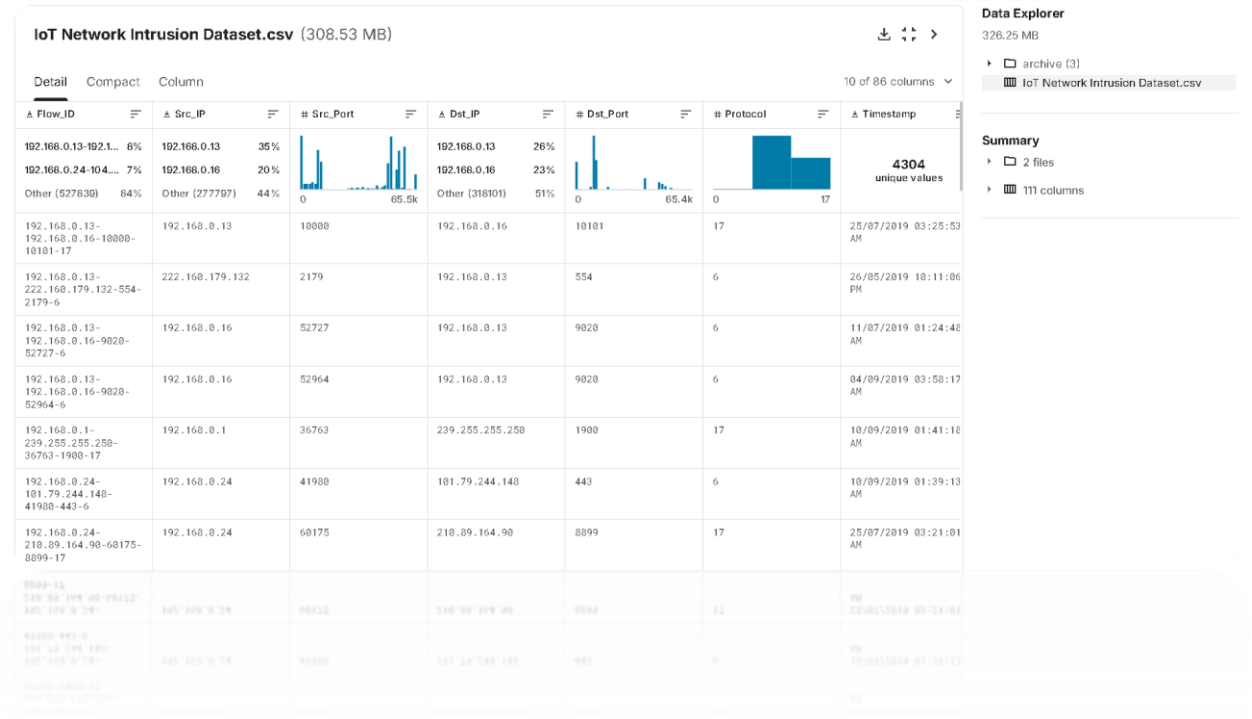
9 **Insecure Default Settings**
Devices or systems shipped with insecure default settings or lack the ability to make the system more secure by restricting operators from modifying configurations.

10 **Lack of Physical Hardening**
Lack of physical hardening measures, allowing potential attackers to gain sensitive information that can help in a future remote attack or take local control of the device.

- **Mirai Botnet: Mirai is one of the most infamous IoT malware strains.** It infects **IoT devices** by **exploiting default** or **weak passwords**, turning them into a **botnet**. **Once infected**, these devices can be used to **launch large-scale Distributed Denial-of-Service (DDoS) attacks, disrupting websites**, **networks,** and **online services**. **Variants of Mirai**, such as **Mirai-Ackflooding** or **Mirai-Hostbruteforce**, target different IoT protocols and services, making it a persistent threat in IoT security.



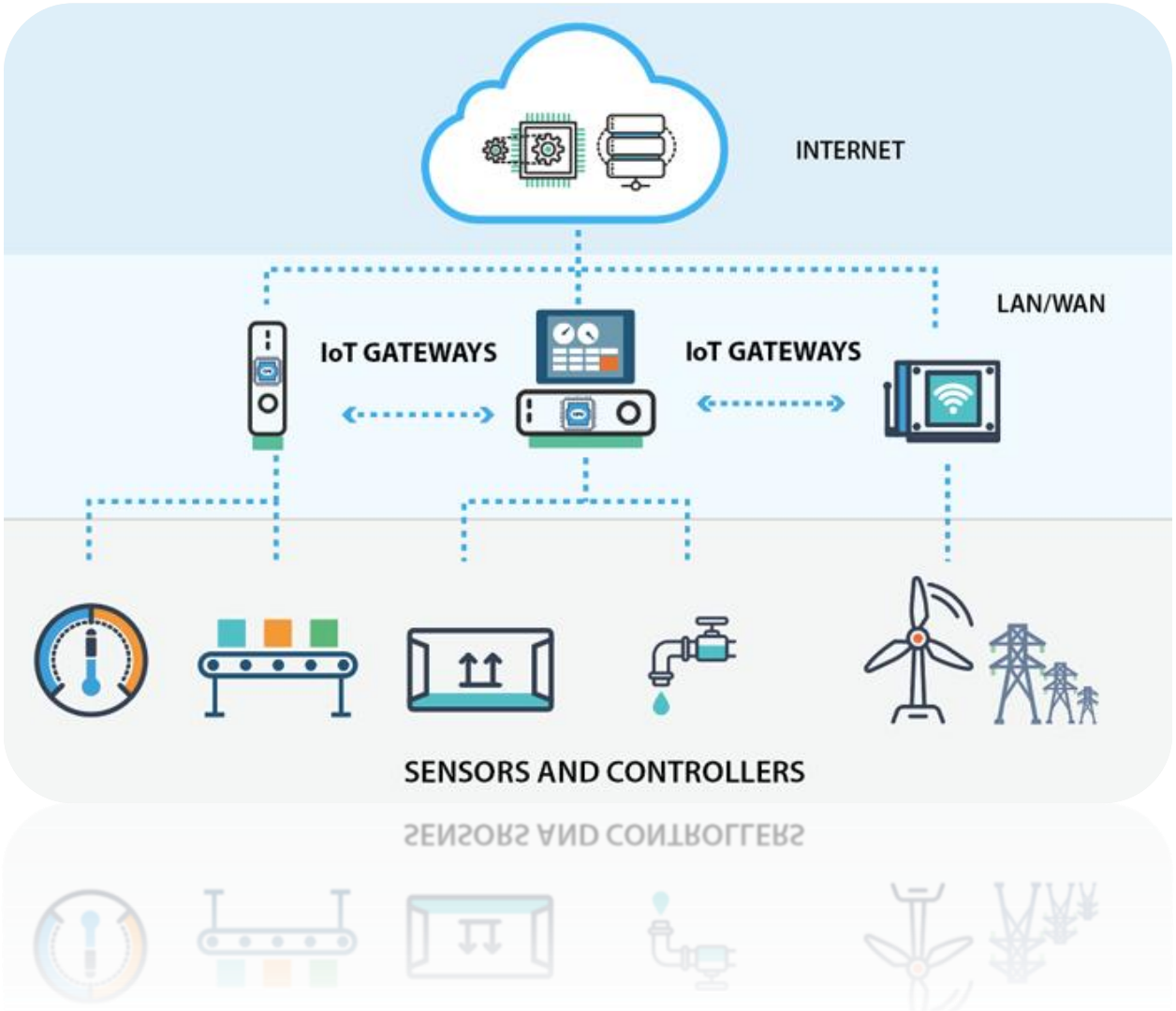
- **Public IDS Datasets:**
  - **For this project**, we used the **IoTID20 dataset** from **Kaggle**. **This dataset** contains **both normal** and **attack traffic** in **IoT networks**, **including attacks** such as **DoS**, **MITM**, and **scanning**. It is publicly available and can be accessed, just press here.



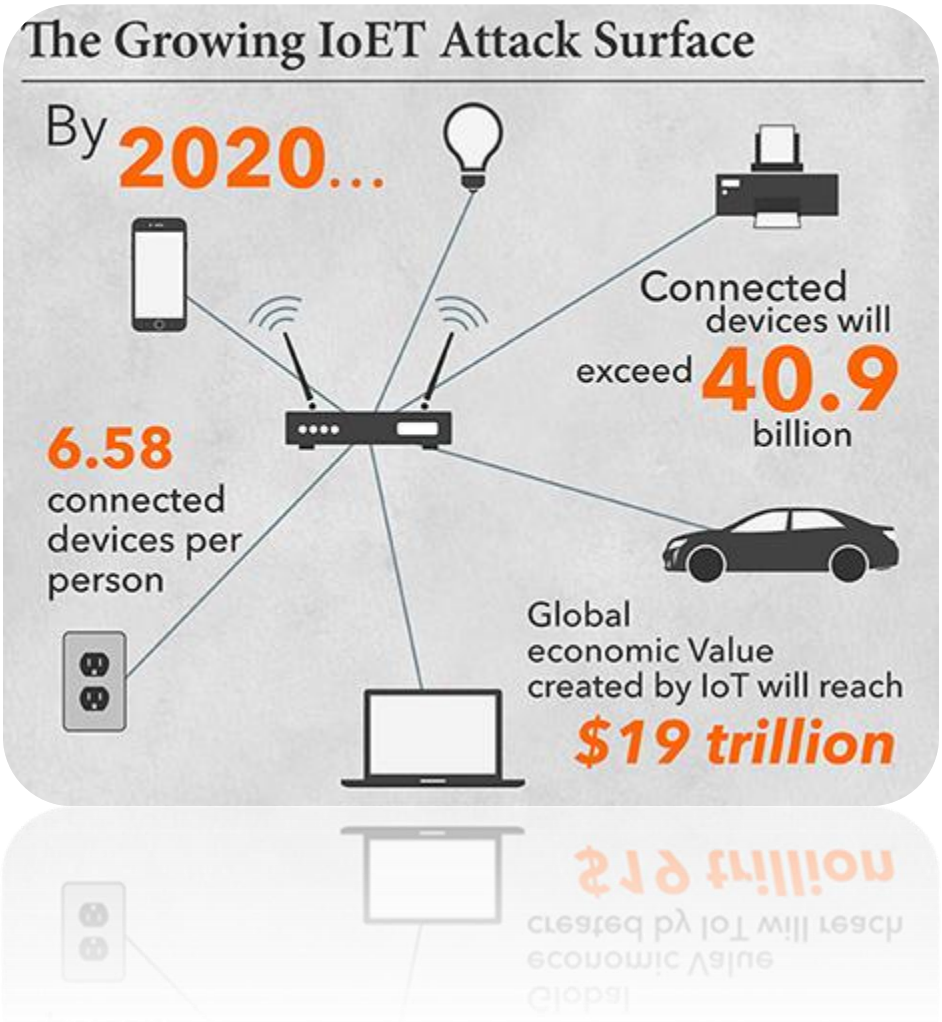
  - **Dataset details:**
    - **File:** IoT Network Intrusion Dataset.csv (308.53 MB)
    - **Usability score:** 2.94
    - **License:** Unknown
    - **Update frequency:** Not specified
  - This dataset allowed us to train and **evaluate** the **AI-based IDS** for **detecting malicious activity** in **IoT environments.**

Task Description / Implementation

- The IoT system modeled in this project consists of a variety of connected devices such as sensors, cameras, smart appliances, and gateways. These devices communicate with a central network gateway or cloud service. The architecture can be represented as:
    - **IoT Devices:** Generate network traffic and collect data.
    - **Edge Gateway / Router:** Aggregates device traffic and forwards it to cloud or monitoring systems.
    - **Cloud/Server:** Performs data storage, analytics, and intrusion detection.

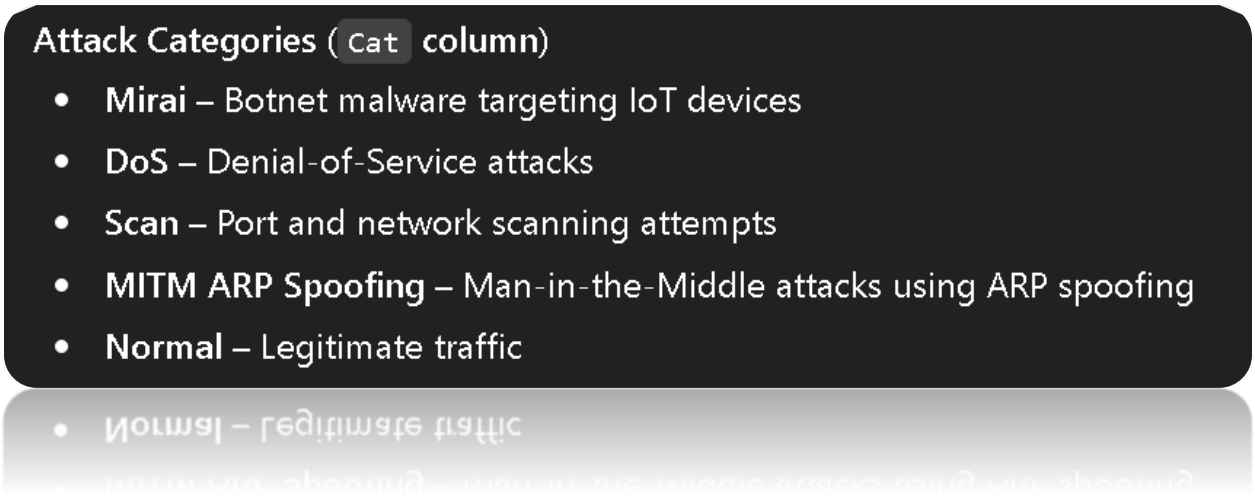


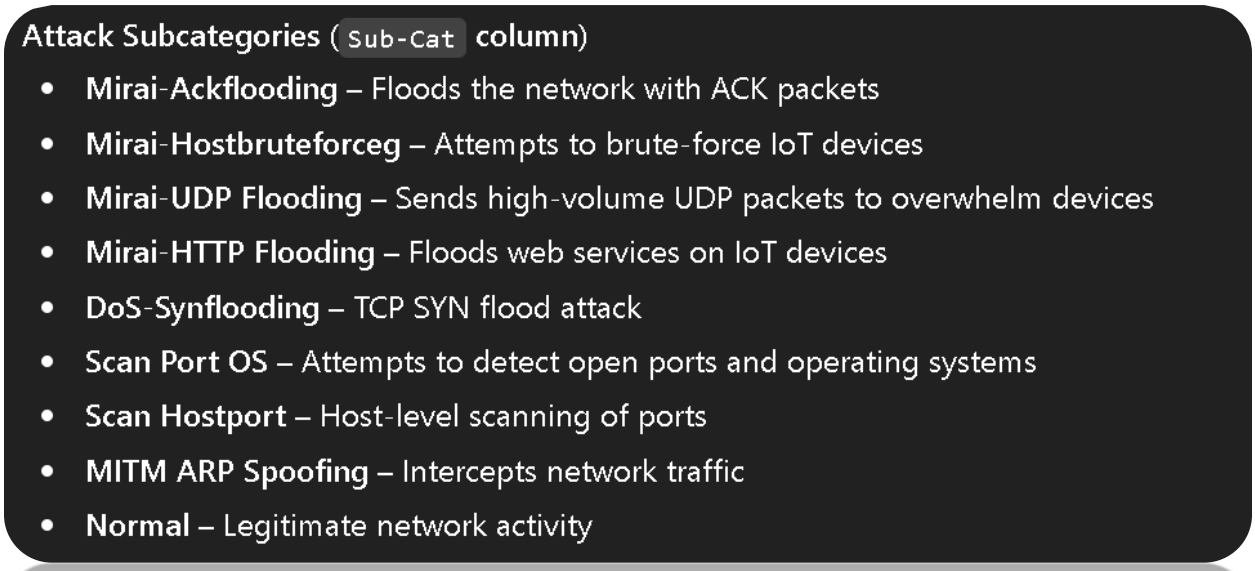
- **Potential Attack Points:**
  - **Device Interfaces: Weak authentication** and **default credentials** make **devices vulnerable** to **malware infection**, including botnets like **Mirai.**
  - **Communication Channels:** Traffic between **devices** and **gateways** may be intercepted or tampered with, allowing **Man-in-the-Middle (MITM) attacks.**
  - **Network Layer:** Overloading devices or gateways can trigger **Denial-of-Service (DoS) attacks.**
  - **Cloud or Server:** Compromised cloud servers can manipulate or inject malicious data into the IoT network.



- Identifying these attack points is critical to designing a robust Intrusion Detection System (IDS) that can monitor and classify abnormal traffic patterns.

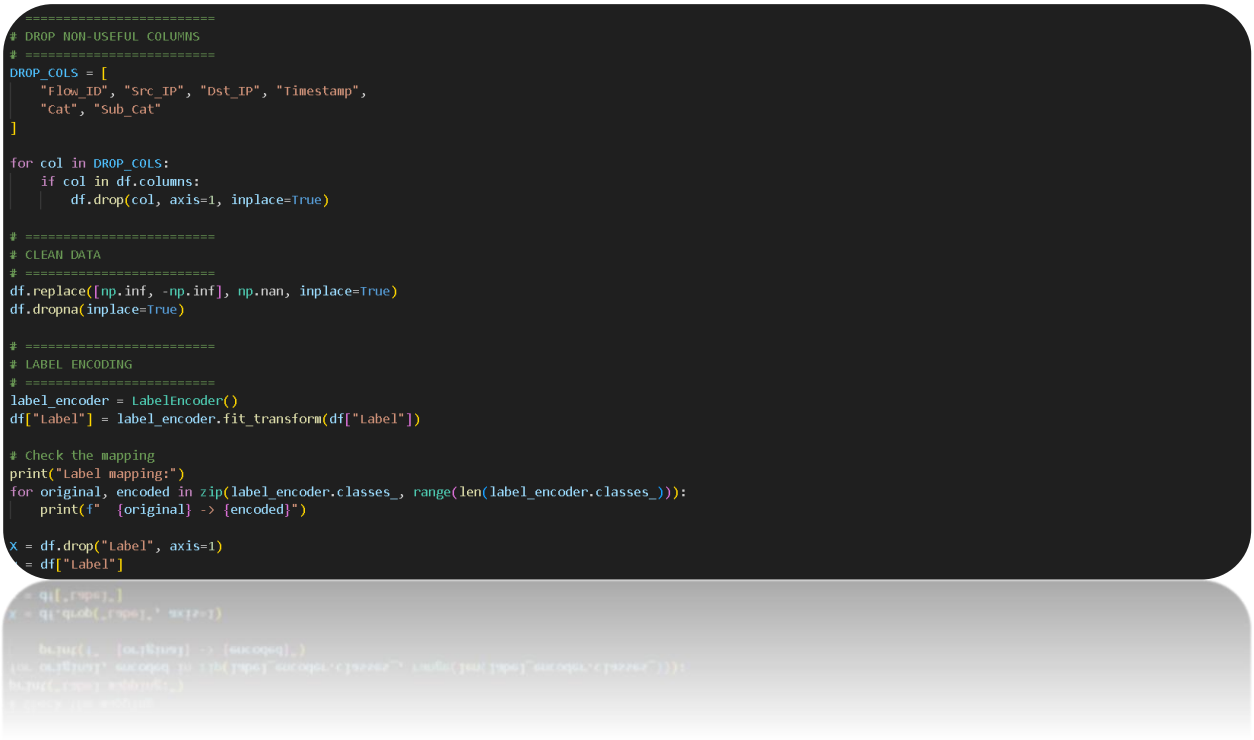
- **Analysis of Cyber-Attacks:**
  - This project focuses on detecting multiple attack types commonly observed in IoT networks.
  - The **IoTID20 dataset** used contains labeled examples for the following **Attack Categories (Cat Column)** and **Attack Subcategories (Sub-Cat Column)** respectively:





  - The **IDS** is trained to classify traffic into **normal** vs. **anomalous**, while also identifying **attack categories** and **subcategories.**

- **AI-Based IDS Design & Detection Workflow:**
    - The **IDS** was implemented using **deep learning**, leveraging a **neural network** built with **TensorFlow/Keras.** The workflow includes:
        - **Data Preprocessing:** Before feeding the data into the **neural network**, **irrelevant columns** such as **IP addresses**, **timestamps,** and **categorical labels** that do not contribute to detection were **removed**. **Missing values** and **infinite values** were handled to ensure **data integrity**. Additionally, the **target labels** were **encoded numerically**, with **Normal mapped** to **1** and **Anomaly mapped** to **0**, enabling the network to perform **binary classification**.



```
# ========================
# DROP NON-USEFUL COLUMNS
# ========================
DROP_COLS = [
    "Flow_ID", "Src_IP", "Dst_IP", "Timestamp",
    "Cat", "Sub_Cat"
]

for col in DROP_COLS:
    if col in df.columns:
        df.drop(col, axis=1, inplace=True)

# ========================
# CLEAN DATA
# ========================
df.replace([np.inf, -np.inf], np.nan, inplace=True)
df.dropna(inplace=True)

# ========================
# LABEL ENCODING
# ========================
label_encoder = LabelEncoder()
df["Label"] = label_encoder.fit_transform(df["Label"])

# Check the mapping
print("Label mapping:")
for original, encoded in zip(label_encoder.classes_, range(len(label_encoder.classes_))):
    print(f"  {original} -> {encoded}")

X = df.drop("Label", axis=1)
= df["Label"]
```

- **Feature Scaling:** To prepare the data for **neural network training**, **all numerical features** were **normalized** using **StandardScaler**. This **standardization** ensures that **each feature has a similar magnitude,** which helps the **network converge faster** and **improves overall performance.**

```
# ==========================
# SCALING
# ==========================
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# SAVE SCALER AND LABEL ENCODER
joblib.dump(scaler, "scaler.save")
joblib.dump(label_encoder, "label_encoder.save")  # Save label encoder too!
```

- **Neural Network Architecture:** The **IDS model** was implemented as a **deep neural network** using **TensorFlow/Keras**. The input layer accepts **all preprocessed features**. **Two hidden layers** are employed: the **first contains 128 neurons** with **ReLU activation** and a **dropout rate** of **0.3** to **reduce overfitting**, while the **second** has **64 neurons** with **ReLU activation** and the **same dropout rate**. The **output layer** consists of a **single neuron** with a **Sigmoid activation function**, **producing** a **probability** indicating whether a **network sample is normal or represents an attack.**

```
# ========================
# IDS MODEL (DEEP LEARNING)
# ========================
model = Sequential([
    Dense(128, activation="relu", input_shape=(X_train.shape[1],)),
    Dropout(0.3),
    Dense(64, activation="relu"),
    Dropout(0.3),
    Dense(1, activation="sigmoid")
])

model.compile(
    optimizer="adam",
    loss="binary_crossentropy",
    metrics=["accuracy"]
)

early_stop = EarlyStopping(
    monitor="val_loss",
    patience=3,
    restore_best_weights=True
)

model.fit(
    X_train, y_train,
    epochs=15,
    batch_size=1024,
    validation_split=0.2,
    callbacks=[early_stop],
    verbose=1
)
```

- **Training Process:** The **dataset** was split into **training** and **testing sets**, with **80%** of the data used for **training** and **20%** reserved for **testing**. The network was trained using the **Adam optimizer** with **binary cross-entropy** as the **loss function**. **Early stopping** was applied to **prevent overfitting** by **monitoring the validation loss** and **restoring the best model weights. Training** was conducted over **15 epochs** with a **batch size** of **1024** to balance **convergence speed** and **stability.**

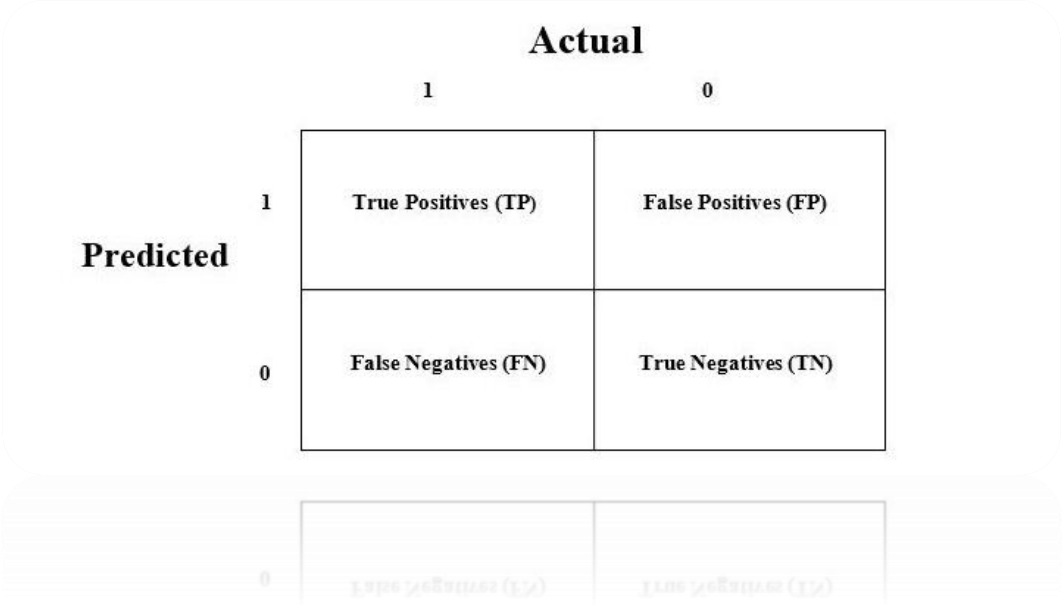
```
# ==========================
# TRAIN / TEST SPLIT
# ==========================
X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.2,
    random_state=42,
    stratify=y
)
```

```
model.compile(
    optimizer="adam",
    loss="binary_crossentropy",
    metrics=["accuracy"]
)

early_stop = EarlyStopping(
    monitor="val_loss",
    patience=3,
    restore_best_weights=True
)

model.fit(
    X_train, y_train,
    epochs=15,
    batch_size=1024,
    validation_split=0.2,
    callbacks=[early_stop],
    verbose=1
)
```

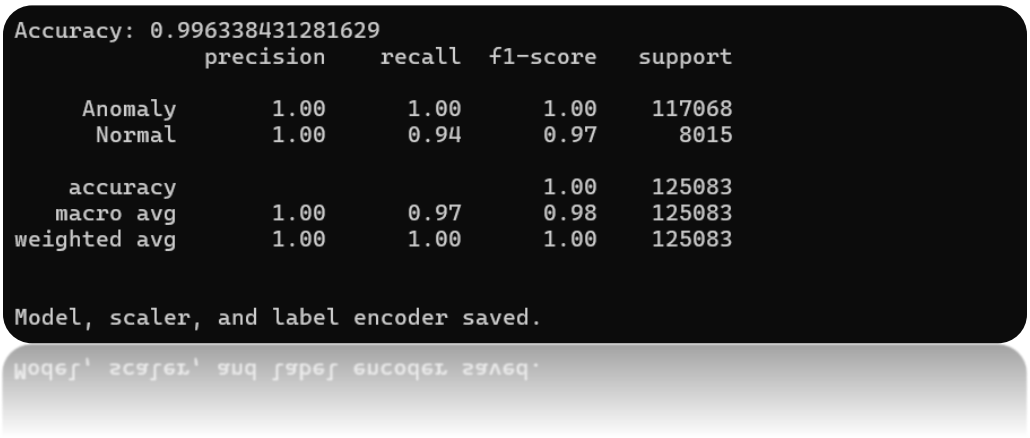
- **Detection Workflow:** During inference, **new network traffic samples** are **preprocessed and scaled using the same procedures** as for the **training data**. The **neural network predicts** the **probability** of a **sample being normal** or **malicious.** A **threshold** of **0.5** is used for **classification**: **probabilities above 0.5** indicate **normal traffic**, while **probabilities below 0.5** indicate an **attack**. **When an attack is detected,** the system also **outputs** the **corresponding attack category** and **subcategory**, **enabling detailed analysis of the type of intrusion.**

```python
# ========================
# IDS DETECTION FUNCTION
# ========================
def detect_intrusion(sample, cat, sub_cat):
    sample = np.array(sample).reshape(1, -1)
    sample = scaler.transform(sample)
    prediction = model.predict(sample, verbose=0)[0][0]

    # Debug: print prediction probability
    print(f"Prediction: {prediction:.4f} (1=Normal, 0=Anomaly)", end=" - ")

    if prediction > 0.5:
        return f"NORMAL TRAFFIC "
    else:
        return f"ATTACK DETECTED {cat} ({sub_cat})"
```

- **Evaluation Metrics:**
  - To **evaluate IDS performance**, the **following metrics were used:**
    - **Accuracy:** Measures the proportion of correct predictions out of all samples.
    - **Precision:** Fraction of correctly predicted attacks out of all predicted attacks.
    - **Recall (Detection Rate):** Fraction of correctly detected attacks out of all actual attacks.
    - **F1-Score:** Harmonic mean of precision and recall, balancing false positives and false negatives.
    - **Confusion Matrix:** Shows the distribution of true vs. predicted labels:



- **Results:**
  - **Achieved ~99.6% accuracy** on the **test set**
  - **Minimal false positives (normal traffic misclassified as attack)**



- **Advantages and Limitations:**
  - **Advantages:**
    - **High accuracy** in detecting attacks in **IoT networks**
    - **Scalable** to **large-scale IoT environments**
  - **Limitations:**
    - **Requires labeled datasets** for **supervised learning**
    - May not detect extremely **rare** or **new attack types**
    - **Training** can be **resource-intensive** for **large networks**

Results / Simulation

- To **evaluate** the **performance** of the **AI-based IDS**, we tested the **trained model** on a **mixture** of **normal** and **attack traffic samples** from the **IoTID20 dataset**. For each sample, the **model predicts** the **probability** of the **traffic being normal** or an **attack**. Using a **classification threshold** of **0.5**, **the model correctly identified all types of attacks,** including **Mirai-based attacks**, **Denial-of-Service (DoS) attempts**, and **network scans**, as well as **normal traffic.**
- For example, **normal traffic samples** were consistently classified as **NORMAL TRAFFIC**, while **attack samples** were correctly labeled with their corresponding **category** and **subcategory**, such as **ATTACK DETECTED → Mirai (Mirai-Ackflooding)** or **ATTACK DETECTED →**

DoS (DoS-Synflooding). The predictions also include the **confidence level,** allowing network administrators to assess the **certainty of each detection.**

```
============================================================
Testing IDS on mixed samples:
============================================================

Testing 5 Normal samples and 5 Attack samples:

Prediction: 0.9635 (1=Normal, 0=Anomaly) - Sample 1: NORMAL TRAFFIC →  [True: Normal] ✓ CORRECT
Prediction: 0.9962 (1=Normal, 0=Anomaly) - Sample 2: NORMAL TRAFFIC →  [True: Normal] ✓ CORRECT
Prediction: 0.9998 (1=Normal, 0=Anomaly) - Sample 3: NORMAL TRAFFIC →  [True: Normal] ✓ CORRECT
Prediction: 1.0000 (1=Normal, 0=Anomaly) - Sample 4: NORMAL TRAFFIC →  [True: Normal] ✓ CORRECT
Prediction: 0.8945 (1=Normal, 0=Anomaly) - Sample 5: NORMAL TRAFFIC →  [True: Normal] ✓ CORRECT
Prediction: 0.0000 (1=Normal, 0=Anomaly) - Sample 6: ATTACK DETECTED → Mirai (Mirai-Ackflooding) [True: Attack] ✓ CORRECT
Prediction: 0.0000 (1=Normal, 0=Anomaly) - Sample 7: ATTACK DETECTED → DoS (DoS-Synflooding) [True: Attack] ✓ CORRECT
Prediction: 0.0012 (1=Normal, 0=Anomaly) - Sample 8: ATTACK DETECTED → Scan (Scan Port OS) [True: Attack] ✓ CORRECT
Prediction: 0.0006 (1=Normal, 0=Anomaly) - Sample 9: ATTACK DETECTED → Mirai (Mirai-Hostbruteforceg) [True: Attack] ✓ CORRECT
Prediction: 0.0008 (1=Normal, 0=Anomaly) - Sample 10: ATTACK DETECTED → Mirai (Mirai-Hostbruteforceg) [True: Attack] ✓ CORRECT

============================================================
Accuracy: 10/10 = 100.0%
============================================================
```

- A **bulk testing evaluation** on a **larger set** of **1,000 randomly selected samples** demonstrated **high accuracy**, with **very few false positives.** The **confusion matrix highlighted** that **most normal** and **attack samples** were **correctly classified**, and **only a minimal number of normal samples** were **mistakenly flagged** as **attacks**. This confirms the **effectiveness** of the **IDS** in **accurately detecting various cyber-attacks in IoT networks.**

```
Bulk Testing Accuracy:
============================================================
Accuracy on 1000 random samples: 0.9980

Confusion Matrix:
              Predicted
              Anomaly  Normal
Actual Anomaly     951       0
Actual Normal        2      47

              precision    recall  f1-score   support

     Anomaly       1.00      1.00      1.00       951
      Normal       1.00      0.96      0.98        49

    accuracy                           1.00      1000
   macro avg       1.00      0.98      0.99      1000
weighted avg       1.00      1.00      1.00      1000

WARNING: 2 normal samples are being classified as attacks!
```

Conclusion

- The **AI-based IDS** demonstrated **high accuracy** in **detecting both normal** and **malicious IoT network traffic. Sample testing** and **bulk evaluation** confirmed its ability to **correctly classify various attack types,** including **Mirai botnet attacks, DoS**, **Scan**, and **MITM attacks.** The results validate the **effectiveness** of using **deep learning** for **intrusion detection in IoT networks.**
- **Future Improvements:**
    - **Implement real-time IDS deployment**
    - **Integrate unsupervised learning** to **detect unknown attacks**

Thank You