

أساسيات تعلم الآلة (Machine Learning) للأمن السيبراني

(MSCS640) الذكاء الاصطناعي والأمن السيبراني

د. المعتر بالله قرباب



أكاديمية جوعان بن جاسم للدراسات الدفاعية
Joaan Bin Jassim Academy for Defence Studies



أكاديمية جوعان بن جاسم للدراسات الدفاعية
Joaan Bin Jassim Academy for Defence Studies

8 أكتوبر 2026

المحتويات

- 1 الذكاء الاصطناعي (AI) وتعلم الآلة (ML) والتعلم العميق (DL)
- 2 نماذج تعلم الآلة (ML) الأساسية
- 3 عائلات نماذج تعلم الآلة (ML) الرئيسية
- 4 مقاييس التقييم للأمن
- 5 تحديات تعلم الآلة (ML) في الأمن السيبراني
- 6 مشروع عملي



أهداف الفصل

- يوفر هذا الفصل المعرفة الأساسية حول تعلم الآلة (Machine Learning (ML) لمحترفي الأمن السيبراني.
- التمييز بين الذكاء الاصطناعي (Artificial Intelligence (AI)، وتعلم الآلة (Machine Learning (ML)، والتعلم العميق (Deep Learning (DL).
- استكشاف النماذج الأساسية: التعلم الخاضع للإشراف (Supervised)، وغير الخاضع للإشراف (Unsupervised)، وشبه الخاضع للإشراف (Semi-supervised)، والتعلم المعزز (Reinforcement Learning).
- تعلم عائلات النماذج الرئيسية: من الغابة العشوائية (Random Forest) إلى الشبكات العصبية (Neural Networks).
- إتقان الجوانب العملية: هندسة الميزات (Feature Engineering)، مقاييس التقييم (Evaluation Metrics)، وسير العمل (Workflow).
- معالجة التحديات: البيانات غير المتوازنة (Imbalanced data)، التشفير (Encryption)، والهجمات المضادة (Adversarial attacks).

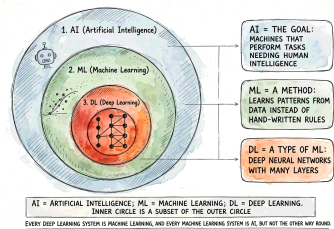


المحتويات

- 1 الذكاء الاصطناعي (AI) وتعلم الآلة (ML) والتعلم العميق (DL)
- 2 نماذج تعلم الآلة (ML) الأساسية
- 3 عائلات نماذج تعلم الآلة (ML) الرئيسية
- 4 مقاييس التقييم للأمن
- 5 تحديات تعلم الآلة (ML) في الأمن السيبراني
- 6 مشروع عملي



تعريفات الذكاء الاصطناعي (AI) وتعلم الآلة (ML) والتعلم العميق (DL)



● الذكاء الاصطناعي (Artificial Intelligence (AI):

المجال الأوسع. إنشاء آلات تؤدي مهام تتطلب ذكاءً بشرياً (مثل اتخاذ القرار، الإدراك البصري).

● تعلم الآلة (Machine Learning (ML):

حقل فرعي من الذكاء الاصطناعي (AI). يمنح أجهزة الكمبيوتر القدرة على التعلم من البيانات دون أن تتم برمجتها بشكل صريح. تجد الخوارزميات الأنماط لعمل تنبؤات.

● التعلم العميق (Deep Learning (DL):

حقل فرعي متخصص من تعلم الآلة (ML). يستخدم شبكات عصبية اصطناعية (*artificial neural networks*) ذات طبقات عديدة ("عميقة") للعثور على أنماط معقدة في البيانات غير المهيكلة (*unstructured data*) مثل الصور والنصوص.

شكل: العلاقة الهرمية: التعلم العميق (DL) جزء من تعلم الآلة (ML)، وتعلم الآلة (ML) جزء من الذكاء الاصطناعي (AI).



القسم 1: تقييم (1/3)

السؤال 1

أي مما يلي يصف العلاقة بين الذكاء الاصطناعي (AI) وتعلم الآلة (ML) والتعلم العميق (DL) بشكل أفضل؟

أ. هي ثلاثة مجالات منفصلة وغير مرتبطة.

ب. الذكاء الاصطناعي (AI) هو مجموعة فرعية من تعلم الآلة (Machine Learning).

ج. التعلم العميق (Deep Learning) هو مجموعة فرعية من تعلم الآلة (Machine Learning)، والذي هو مجموعة فرعية من الذكاء الاصطناعي (AI).

د. يستخدم تعلم الآلة (Machine Learning) الشبكات العصبية، بينما يستخدم الذكاء الاصطناعي (AI) المنطق.



القسم 1: تقييم (1/3) - الإجابة

السؤال 1

أي مما يلي يصف العلاقة بين الذكاء الاصطناعي (AI) وتعلم الآلة (ML) والتعلم العميق (DL) بشكل أفضل؟

أ. هي ثلاثة مجالات منفصلة وغير مرتبطة.

ب. الذكاء الاصطناعي (AI) هو مجموعة فرعية من تعلم الآلة (Machine Learning).

ج. التعلم العميق (Deep Learning) هو مجموعة فرعية من تعلم الآلة (Machine Learning)، والذي هو مجموعة فرعية من الذكاء الاصطناعي (AI). (✓)

د. يستخدم تعلم الآلة (Machine Learning) الشبكات العصبية، بينما يستخدم الذكاء الاصطناعي (AI) المنطق.

الشرح

كما هو موضح في مخطط فن (Venn diagram)، الذكاء الاصطناعي (AI) هو المظلة الواسعة. تعلم الآلة (ML) هي طريقة التعلم من البيانات داخل الذكاء الاصطناعي (AI). والتعلم العميق (DL) هي تقنية متخصصة تستخدم الشبكات العصبية داخل تعلم الآلة (ML).



القسم 1: تقييم (2/3)

السؤال 2

ما هي الخاصية الأساسية التي تعرف تعلم الآلة (Machine Learning)؟

- أ. يتطلب برمجة صريحة لكل سيناريو.
- ب. يتعلم الأنماط من البيانات لعمل تنبؤات على بيانات غير مرئية.
- ج. يعمل فقط مع الصور والصوت.
- د. يحاكي الدماغ البشري باستخدام الخلايا العصبية.



القسم 1: تقييم (2/3) - الإجابة

السؤال 2

ما هي الخاصية الأساسية التي تعرف تعلم الآلة (Machine Learning)؟

- أ. يتطلب برمجة صريحة لكل سيناريو.
- ب. يتعلم الأنماط من البيانات لعمل تنبؤات على بيانات غير مرئية. (✓)
- ج. يعمل فقط مع الصور والصوت.
- د. يحاكي الدماغ البشري باستخدام الخلايا العصبية.

الشرح

يُعرف تعلم الآلة (ML) بقدرته على تعلم القواعد من البيانات بدلاً من وجود قواعد مشفرة بشكل صارم بواسطة مبرمج.



القسم 1: تقييم (3/3)

السؤال 3

التعلم العميق (Deep Learning) فعال بشكل خاص لأي نوع من البيانات؟

- أ. جداول البيانات البسيطة.
- ب. البيانات غير المهيكلة (unstructured data) مثل الصور والصوت والنص.
- ج. مجموعات البيانات الصغيرة ذات الميزات القليلة.
- د. مشاكل الانحدار الخطي (Linear regression).



القسم 1: تقييم (3/3) - الإجابة

السؤال 3

التعلم العميق (Deep Learning) فعال بشكل خاص لأي نوع من البيانات؟

- أ. جداول البيانات البسيطة.
- ب. البيانات غير المهيكلة (unstructured data) مثل الصور والصوت والنص. (✓)
- ج. مجموعات البيانات الصغيرة ذات الميزات القليلة.
- د. مشاكل الانحدار الخطي (Linear regression).

الشرح

يسمح الهيكل متعدد الطبقات لشبكات التعلم العميق (Deep Learning) باستخراج الميزات تلقائياً من البيانات المعقدة وعالية الأبعاد وغير المهيكلة.



المحتويات

- 1 الذكاء الاصطناعي (AI) وتعلم الآلة (ML) والتعلم العميق (DL)
- 2 نماذج تعلم الآلة (ML) الأساسية
- 3 عائلات نماذج تعلم الآلة (ML) الرئيسية
- 4 مقاييس التقييم للأمن
- 5 تحديات تعلم الآلة (ML) في الأمن السيبراني
- 6 مشروع عملي



التعلم الخاضع للإشراف (Supervised) مقابل غير الخاضع للإشراف (Unsupervised)

التعلم غير الخاضع للإشراف (Unsupervised Learning) (بالملاحظة)

- المفهوم: تغذية البيانات الخام بدون تسميات. يقوم الذكاء الاصطناعي (AI) بتجميع أوجه التشابه وإبراز القيم المتطرفة (outliers).
- الأفضل لـ: التهديدات الداخلية (insider threats) والبحث عن هجمات جديدة غير معروفة.

التعلم الخاضع للإشراف (Supervised Learning) (بالمثال)

- المفهوم: التدريس باستخدام التسميات (labels) ("هذا برنامج ضار"، "هذا آمن").
- الأفضل لـ: اكتشاف البرامج الضارة (Malware) وتصنيف الهجمات المعروفة.



خط أنابيب إعداد البيانات I (Data Preparation Pipeline)

"مدخلات سيئة تعني مخرجات سيئة". يجب تنظيف البيانات قبل التدريب.

1 التطبيع (Normalization): تحويل لغات الأدوات المختلفة (Microsoft) مقابل (Cisco) إلى تنسيق قياسي واحد (مثل JSON).

● الأداة A تقول: "User: logged Admin in."

● الأداة B تقول: "Account: Admin, Action: Authentication Success."

2 مزامنة الوقت (Time Synchronization): يجب أن تستخدم جميع الأجهزة التوقيت العالمي المنسق (UTC). إذا اختلفت الساعات، يتم تدمير الجدول الزمني للسبب والنتيجة للهجوم.



خط أنابيب إعداد البيانات (Data Preparation Pipeline) II

استخراج الميزات (Feature Extraction)

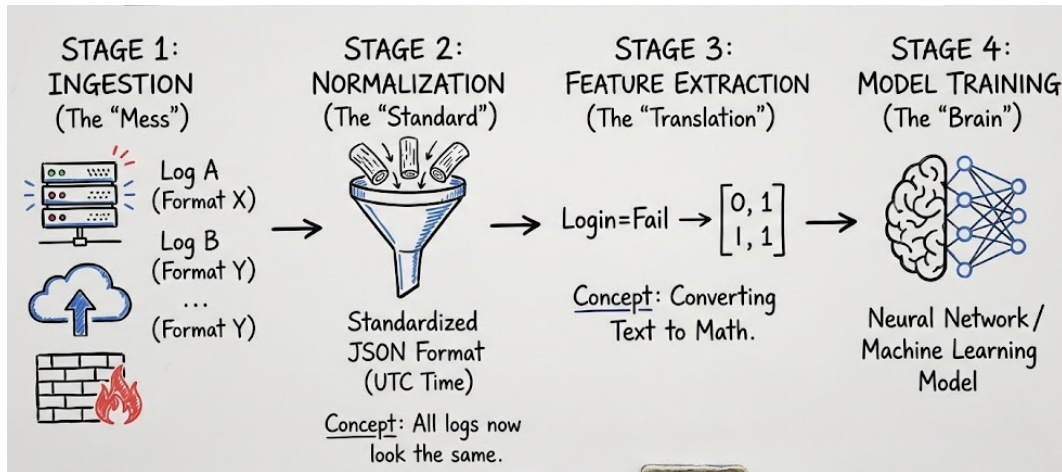
نماذج الذكاء الاصطناعي (AI) مثل الشبكات العصبية (Neural Networks) لا يمكنها قراءة النص. إنها تفهم الرياضيات.

● الختام: "Status=Fail" → الميزة: 1 = is_failure

● الختام: "AM 10:00" → الميزة: 10 = hour



خط أنابيب إعداد البيانات (Data Preparation Pipeline)



Python: محاكاة البيانات (1/2)

تحاكي كجلة الكود هذه تحميل بيانات السجل الخام. في سيناريو حقيقي، سيكون هذا ملف CSV كبير.

```
import pandas as pd
import io

# 1. Simulate raw log data (CSV format)
log_data = """timestamp,source_ip,status_code
2025-08-16 10:00:01,192.168.1.10,200
2025-08-16 10:00:05,192.168.1.10,404
2025-08-16 10:00:10,192.168.1.10,404
2025-08-16 10:00:15,10.0.0.5,200
"""

# 2. Load the data into a DataFrame
df = pd.read_csv(io.StringIO(log_data))
```



Python: منطق استخراج الميزات (Feature Extraction) (2/2)

هنا، نقوم بتحويل رموز "404" الختام إلى عدد رياضي. يمكن للذكاء الاصطناعي (AI) أن يتعلم أن العدد الكبير مثير للريبة.

```
# 3. FEATURE EXTRACTION
```

```
# Logic: Group by IP address, count how many '404' errors appear.
```

```
features = df.groupby('source_ip')['status_code'].apply(
    lambda x: (x == 404).sum()
).reset_index(name='error_count_feature')
```

```
print("Data ready for AI Model:")
print(features)
```

```
# Output:
```

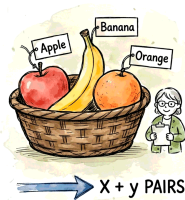
```
#   source_ip  error_count_feature
# 0    10.0.0.5                    0
# 1  192.168.1.10                  2
```



نظرة عامة على التعلم الخاضع للإشراف (Supervised Learning)

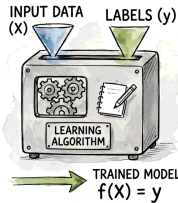
تتعلم الخوارزمية من مجموعة بيانات حيث يتم إقران المدخلات بتسميات الإخراج (labels) الصحيحة.

1. TRAINING DATA (Practice Exam with Answer Key)



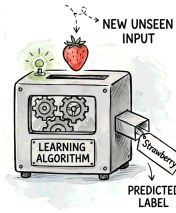
The teacher supplies both the questions and the answers.

2. MODEL TRAINING (Learning the Mapping X to y)



The student learns the rule that links each input to its label.

3. PREDICTION (Exam with Hidden Answers)



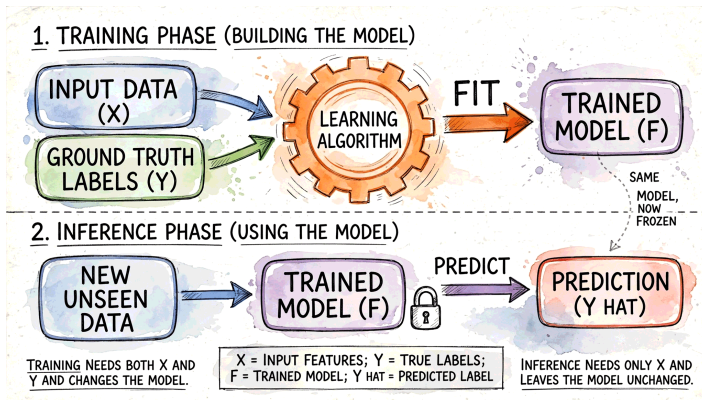
The trained model labels an input it has never seen.

X = input features; y = correct label; f = the learned mapping from X to y

شكل: تعلم الآلة الخاضع للإشراف (Supervised Machine Learning): تعلم التعيين من المدخلات (X) إلى المخرجات (y).



نظام التعلم الخاضع للإشراف (Supervised Learning)

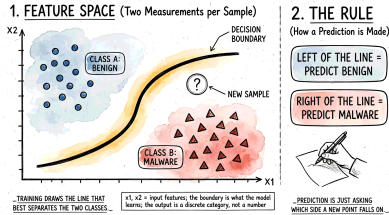


شكل: نظرة عامة على نظام تعلم الآلة الخاضع للإشراف (Supervised Machine Learning).

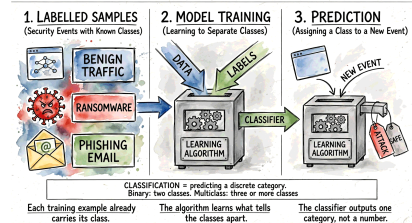


التطبيق: التصنيف (Classification)

التنبؤ بفئة منفصلة (مثل حميد مقابل براجم ضارة (malware)).



شكل: حدود التصنيف (Classification Boundary).



شكل: نظرة عامة على تصنيف تعلم الآلة (ML).

الكود: التصنيف الخاضع للإشراف (Supervised Classification) للتصيد

كاشف تصيد (phishing) بسيط باستخدام الانحدار اللوجستي (Logistic Regression). الميزات: طول الرابط (URL) وعدد الأحرف الخاصة.

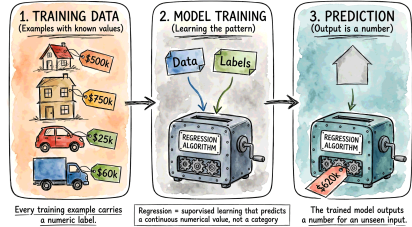
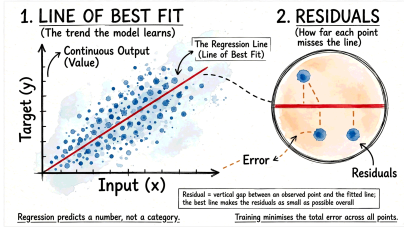
```

1 import numpy as np
2 from sklearn.linear_model import LogisticRegression
3
4 # --- CLASSIFICATION EXAMPLE: Phishing Detection ---
5 # 1. Dataset (Features: [URL Length, Number of '@' symbols])
6 X_train = np.array([
7     [15, 0], [22, 0], [18, 0], # Benign samples
8     [85, 1], [90, 2], [75, 1] # Phishing samples
9 ])
10 # Labels (0: Benign, 1: Phishing)
11 y_train = np.array([0, 0, 0, 1, 1, 1])
12
13 # 2. Train Model
14 clf = LogisticRegression()
15 clf.fit(X_train, y_train)
16
17 # 3. Predict on new data
18 new_email = np.array([[80, 1]]) # A long URL with an '@' symbol
19 prediction = clf.predict(new_email)
20 print(f"Classification Prediction: {prediction[0]} (1=Phishing, 0=Benign)")

```



التنبؤ بقيمة رقمية مستمرة (مثل درجة المخاطرة).



شكل: ملائمة خط للتنبؤ بالاتجاهات.

شكل: نظرة عامة على الانحدار.



الكود: الانحدار الخاضع للإشراف (Supervised Regression) لدرجة المخاطرة

التنبؤ بدرجة المخاطرة (risk score) بناءً على عدد الثغرات الأمنية الحرجة (critical vulnerabilities) المفتوحة.

```

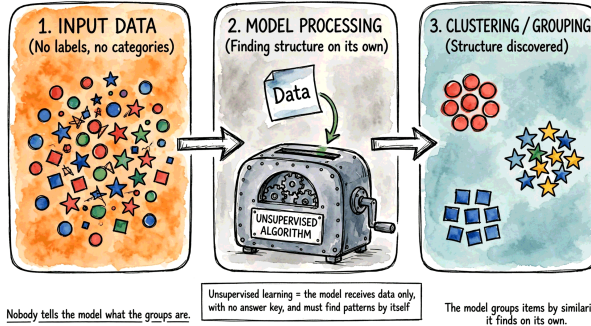
1 import numpy as np
2 from sklearn.linear_model import LinearRegression
3
4 # --- REGRESSION EXAMPLE: Risk Scoring ---
5 # 1. Dataset (Feature: [Count of Open Critical Vulnerabilities])
6 X_history = np.array([ [0], [1], [2], [5], [10] ])
7
8 # Labels (Target Risk Score: 0.0 to 100.0)
9 y_history = np.array([10.0, 25.0, 40.0, 65.0, 95.0])
10
11 # 2. Train Model
12 reg = LinearRegression()
13 reg.fit(X_history, y_history)
14
15 # 3. Predict Risk for a system with 7 open vulnerabilities
16 new_system = np.array([[7]])
17 predicted_score = reg.predict(new_system)
18
19 print(f"Regression Prediction (Risk Score): {predicted_score[0]:.2f}")

```



نظرة عامة على التعلم غير الخاضع للإشراف (Unsupervised Learning)

العمل مع البيانات التي ليس لها تسميات (labels) محددة مسبقًا. يجب أن يجد النموذج البنية بمفرده.

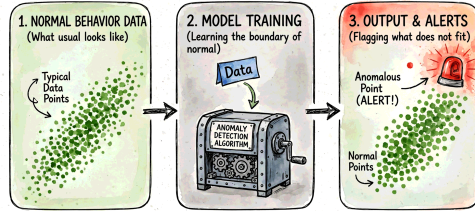


شكل: التعلم غير الخاضع للإشراف (Unsupervised Learning): العثور على بنية في البيانات غير المسماة.



التطبيق: كشف الشذوذ (Anomaly Detection)

تحديد القيم المتطرفة (outliers) التي تختلف اختلافاً كبيراً عن القاعدة (مثل التهديدات الداخلية (insider threats)).



Anomaly = a data point that lies far outside the learned normal distribution; used to catch zero-day attacks and insider threats

The model is trained on normal behaviour only.

Anything far from normal is raised as an alert.

شكل: كشف الشذوذ.



الكود: كشف الشذوذ (Anomaly Detection)

استخدام غابة العزل (*Isolation Forest*) لاكتشاف أنماط الوصول غير الطبيعية للملفات (التهديد الداخلي).

```

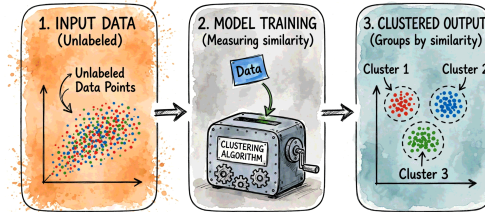
1 from sklearn.ensemble import IsolationForest
2 import numpy as np
3
4 # --- ANOMALY DETECTION: Insider Threat ---
5 # 1. Training Data (Normal User Behavior: Files per minute)
6 X_normal = np.array([[5], [6], [5], [7], [10], [8], [5]])
7
8 # 2. Train Model (contamination=0.01 tells model anomalies are rare)
9 iso = IsolationForest(contamination=0.01, random_state=42)
10 iso.fit(X_normal)
11
12 # 3. Live Data Monitoring
13 # User A (Normal), User B (Data Exfiltration)
14 X_live = np.array([[6], [1000]])
15
16 predictions = iso.predict(X_live)
17 # Output: 1 = Normal, -1 = Anomaly
18 print(f"Predictions: {predictions}")

```



التطبيق: التجميع (Clustering)

تجميع نقاط البيانات المتشابهة (مثل تصنيف عائلات البرامج الضارة (malware families)).



Clustering = grouping unlabeled points so that items in one group are more alike than items in different groups; used to sort malware families and botnet hosts

The input carries no group names at all.

The groups emerge from similarity, not from labels.

شكل: التجميع: تجميع البيانات غير المسماة.



الكود: التجميع بخوارزمية المتوسطات (K-Means)

استخدام خوارزمية المتوسطات (K-Means) لفصل حركة مرور نظام أسماء النطاقات (DNS) عن حركة مرور بروتوكول النقل الآمن (HTTPS) بدون تسميات.

```

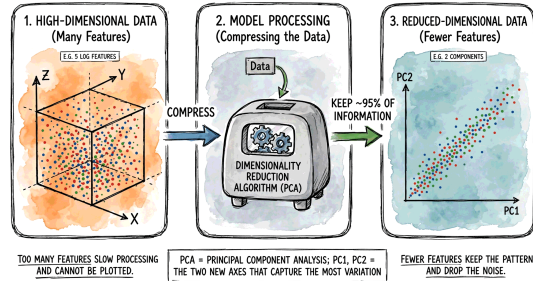
1 from sklearn.cluster import KMeans
2 import numpy as np
3
4 # --- CLUSTERING: Traffic Categorization ---
5 # 1. Unlabeled Data (Features: [Packet Size, Destination Port])
6 X_traffic = np.array([
7     [64, 53], [70, 53], [60, 53],      # Small packets on Port 53 (DNS)
8     [1500, 443], [1450, 443], [1500, 443] # Large packets on Port 443 (HTTPS)
9 ])
10
11 # 2. Train Model (Find 2 clusters)
12 kmeans = KMeans(n_clusters=2, random_state=42)
13 kmeans.fit(X_traffic)
14
15 # 3. Analyze Results
16 print(f"Cluster Labels: {kmeans.labels_}")
17 # Expected: [0, 0, 0, 1, 1, 1] - distinct groups found.

```



التطبيق: تقليل الأبعاد (Dimensionality Reduction)

ضغط البيانات المعقدة مع الاحتفاظ بالمعلومات (مثل التصور، المعالجة المسبقة).



شكل: تقليل الأبعاد.



الكود: تقليل الأبعاد بتحليل المكونات الرئيسية (PCA)

استخدام تحليل المكونات الرئيسية (PCA) لضغط بيانات السجل عالية الأبعاد.

```

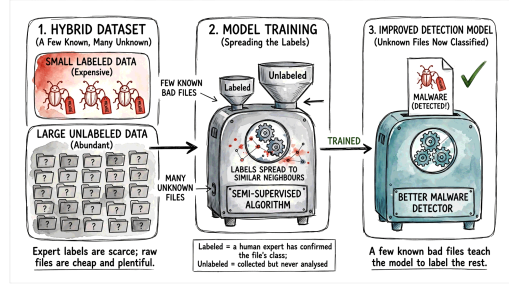
1 from sklearn.decomposition import PCA
2 import numpy as np
3
4 # --- DIMENSIONALITY REDUCTION: Log Compression ---
5 # 1. High-Dimensional Data (10 samples, 5 features each)
6 X_logs = np.random.rand(10, 5)
7
8 # 2. Apply PCA
9 # Compress 5 features down to 2 "Principal Components"
10 pca = PCA(n_components=2)
11 X_compressed = pca.fit_transform(X_logs)
12
13 # 3. Result
14 print(f"Compressed Shape: {X_compressed.shape}")
15 print(f"Information Retained: {np.sum(pca.explained_variance_ratio_):.2%}")

```



التعلم شبه الخاضع للإشراف (Semi-supervised Learning)

يستخدم كمية صغيرة من البيانات المسماة وكمية كبيرة من البيانات غير المسماة.



شكل: نشر التسميات (Label Propagation) من البيانات المعروفة إلى غير المعروفة.



الكود: التعلم شبه الخاضع للإشراف (Semi-supervised Learning)

استخدام نشر التسمية (Label Spreading) لاستنتاج حالة البرامج الضارة للملفات غير المعروفة.

```

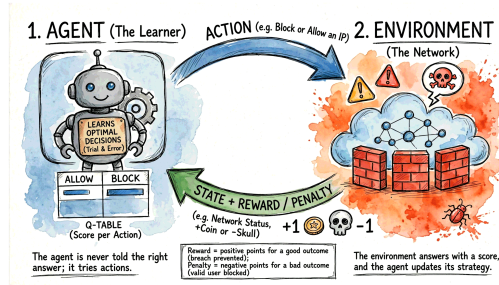
1 import numpy as np
2 from sklearn.semi_supervised import LabelSpreading
3
4 # --- SEMI-SUPERVISED EXAMPLE: Malware Label Propagation ---
5 # 1. Dataset (Feature: [File Entropy, File Size])
6 X = np.array([
7     [0.1, 100], [0.2, 120], # Cluster A (Likely Benign)
8     [0.9, 900], [0.8, 950], # Cluster B (Likely Malware)
9     [0.15, 110], [0.85, 920] # New Files (Unknown)
10 ])
11 # 2. Labels (0: Benign, 1: Malware, -1: Unknown)
12 y_mixed = np.array([0, 0, 1, 1, -1, -1])
13
14 # 3. Train Model
15 model = LabelSpreading(kernel='knn', n_neighbors=2)
16 model.fit(X, y_mixed)
17
18 # 4. Results
19 print(f"Inferred Labels: {model.transduction_}")

```



التعلم المعزز (Reinforcement Learning (RL))

يتعلم الوكيل (agent) عن طريق التجربة والخطأ (الإجراءات، المكافآت، العقوبات).



شكل: حلقة التعلم المعزز (Reinforcement Learning).

الكود: التعلم المعزز (Reinforcement Learning) (وكيل جدار الحماية)

حلقة خوارزمية تعلم كيو (Q-Learning) بسيطة لجدار حماية (firewall) مستقل.

```

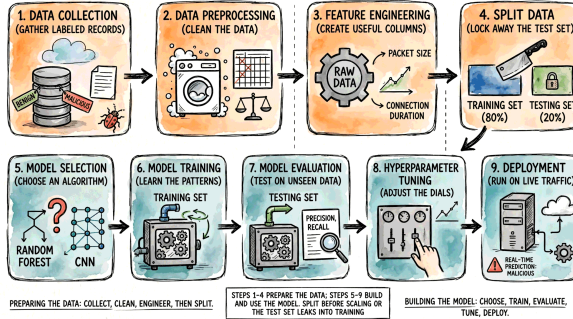
1 import numpy as np
2 import random
3 # --- RL EXAMPLE: Q-Learning Firewall Agent ---
4 # Actions: 0 = Allow, 1 = Block. State: IP '192.168.1.5'
5 q_table = { '192.168.1.5': [0.0, 0.0] }
6
7 def get_reward(action, is_malicious):
8     if is_malicious and action == 1: return +10 # Good: Blocked attack
9     if is_malicious and action == 0: return -50 # Bad: Allowed attack
10    return +5 if action == 0 else -10 # Valid user logic
11
12 # Simulation Loop
13 ip = '192.168.1.5'; is_malicious = True; learning_rate = 0.5
14 for episode in range(5):
15     # Choose Action (Explore vs Exploit)
16     action = random.choice([0, 1]) if random.random() < 0.5 else np.argmax(q_table[ip])
17
18     reward = get_reward(action, is_malicious)
19
20     # Update Q-Table
21     old_value = q_table[ip][action]
22     q_table[ip][action] = old_value + (learning_rate * reward)
23     print(f"Ep {episode+1}: Action {action} -> Reward {reward}")

```



سير عمل التعلم الخاضع للإشراف (Supervised Learning)

النهج المنظم أمر بالغ الأهمية لمنع تسرب البيانات (data leakage).



شكل: سير عمل التعلم الخاضع للإشراف (Supervised Learning).



سير العمل: 1. جمع البيانات و 2. التقسيم

قم بتقسيم البيانات (data splitting) على الفور لمنع التسرب (leakage).

```

1 import pandas as pd
2 from sklearn.model_selection import train_test_split
3
4 # 1. Data Collection (Mock Dataset)
5 data = {
6     'duration': [0.1, 5.2, 0.2, 12.0, 0.1, 0.5, 4.5, 1.2, 0.1, 3.0],
7     'src_bytes': [64, 5000, 128, 9000, 64, 256, 4500, 500, 64, 6000],
8     'dst_bytes': [1000, 0, 500, 0, 800, 1200, 0, 300, 950, 0],
9     'label': [0, 1, 0, 1, 0, 0, 1, 0, 0, 1] # 0: Benign, 1: Malicious
10 }
11 df = pd.DataFrame(data)
12
13 # 2. Data Splitting
14 X = df.drop('label', axis=1); y = df['label']
15
16 # stratify=y ensures both sets maintain the same ratio of attacks
17 X_train, X_test, y_train, y_test = train_test_split(
18     X, y, test_size=0.2, random_state=42, stratify=y
19 )

```



سير العمل: 3. الهندسة و 4. المعالجة المسبقة

هندسة الميزات (Feature Engineering) والمعالجة المسبقة (Preprocessing): ملائمة المقاييس (scalers) فقط على بيانات التدريب (training data).

```

1 from sklearn.preprocessing import StandardScaler
2
3 # 3. Feature Engineering (Apply to both)
4 X_train['bytes_per_sec'] = X_train['src_bytes'] / (X_train['duration'] + 0.01)
5 X_test['bytes_per_sec'] = X_test['src_bytes'] / (X_test['duration'] + 0.01)
6
7 # 4. Data Preprocessing (Scaling)
8 scaler = StandardScaler()
9
10 # CRITICAL: Fit scaler ONLY on Training data
11 X_train_scaled = scaler.fit_transform(X_train)
12
13 # Transform Test data using learned parameters
14 X_test_scaled = scaler.transform(X_test)

```



سير العمل: 5-7. الاختيار، الضبط، التدريب

اختيار النموذج (Model Selection) وضبط المعاملات الفائقة (Hyperparameter Tuning): استخدم التحقق المتبادل (Cross-Validation) على مجموعة التدريب (training set).

```

1 from sklearn.ensemble import RandomForestClassifier
2 from sklearn.model_selection import GridSearchCV
3
4 # 5. Model Selection
5 rf = RandomForestClassifier(random_state=42)
6
7 # 6. Hyperparameter Tuning (Cross-Validation)
8 param_grid = {'n_estimators': [10, 50], 'max_depth': [3, 5]}
9 grid_search = GridSearchCV(rf, param_grid, cv=2, scoring='f1')
10 grid_search.fit(X_train_scaled, y_train)
11
12 # 7. Model Training (Automatic via GridSearchCV)
13 best_model = grid_search.best_estimator_
14 print(f"Best Parameters: {grid_search.best_params_}")

```



سير العمل: 8. التقييم و 9. النشر

التقييم (Evaluation) والنشر (Deployment): قم بالتقييم على مجموعة الاختبار (test set) المقللة.

```

1 from sklearn.metrics import classification_report
2
3 # 8. Model Evaluation
4 y_pred = best_model.predict(X_test_scaled)
5 print(classification_report(y_test, y_pred, target_names=['Benign', 'Malicious']))
6
7 # 9. Deployment (Simulation)
8 def analyze_packet(duration, src_bytes, dst_bytes):
9     live_data = pd.DataFrame([[duration, src_bytes, dst_bytes]],
10                               columns=['duration', 'src_bytes', 'dst_bytes'])
11     live_data['bytes_per_sec'] = live_data['src_bytes'] / (live_data['duration'] + 0.01)
12     live_scaled = scaler.transform(live_data)
13     return "ALERT" if best_model.predict(live_scaled)[0] == 1 else "Benign"
14
15 print(analyze_packet(15.0, 10000, 0))

```



القسم 2: تقييم (1/3)

السؤال 1

أي نموذج من نماذج تعلم الآلة (ML) يتطلب مجموعة بيانات بأمثلة مسماة (أزواج الإدخال/الإخراج)؟

- أ. التعلم غير الخاضع للإشراف (Unsupervised Learning).
- ب. التعلم المعزز (Reinforcement Learning).
- ج. التعلم الخاضع للإشراف (Supervised Learning).
- د. التجميع (Clustering).



القسم 2: تقييم (1/3) - الإجابة

السؤال 1

- أي نموذج من نماذج تعلم الآلة (ML) يتطلب مجموعة بيانات بأمثلة مسماة (أزواج الإدخال/الإخراج)؟
- أ. التعلم غير الخاضع للإشراف (Unsupervised Learning).
 - ب. التعلم المعزز (Reinforcement Learning).
 - ج. التعلم الخاضع للإشراف (Supervised Learning). (✓)
 - د. التجميع (Clustering).

الشرح

يعمل التعلم الخاضع للإشراف (Supervised Learning) مثل نموذج المعلم/الطالب، حيث يتطلب "مفتاحاً" (تسميات) لتعلم العلاقة بين المدخلات والمخرجات.



القسم 2: تقييم (2/3)

السؤال 2

- ما هي حالة الاستخدام الأساسية للتعلم غير الخاضع للإشراف (Unsupervised Learning) في الأمن السيبراني؟
- أ. كشف التصيد باستخدام روابط سيئة معروفة.
 - ب. كشف الشذوذ (العثور على هجمات يوم الصفر (zero-day) أو القيم المتطرفة).
 - ج. التنبؤ بدرجات المخاطرة الدقيقة.
 - د. تدريب وكيل جدار الحماية عبر المكافآت.



القسم 2: تقييم (2/3) - الإجابة

السؤال 2

- ما هي حالة الاستخدام الأساسية للتعلم غير الخاضع للإشراف (Unsupervised Learning) في الأمن السيبراني؟
- أ. كشف التصيد باستخدام روابط سيئة معروفة.
 - ب. كشف الشذوذ (العثور على هجمات يوم الصفر (zero-day) أو القيم المتطرفة). (✓)
 - ج. التنبؤ بدرجات المخاطرة الدقيقة.
 - د. تدريب وكيل جدار الحماية عبر المكافآت.

الشرح

نظراً لأن التعلم غير الخاضع للإشراف (unsupervised learning) لا يحتاج إلى تسميات، فهو مثالي للعثور على البيانات التي تخوف عن "القاعدة" دون معرفة شكل الهجوم مسبقاً.



القسم 2: تقييم (3/3)

السؤال 3

في سير عمل التعلم الخاضع للإشراف (Supervised Learning)، لماذا يجب حساب معلمات القياس (scaling parameters) فقط على مجموعة التدريب؟

- أ. لتوفير وقت الحساب.
- ب. لمنع تسرب البيانات (Data Leakage) (اختلاس النظر في مجموعة الاختبار).
- ج. لأن مجموعة الاختبار لا تحتوي على أرقام.
- د. لا يهم؛ يمكنك استخدام مجموعة البيانات بأكملها.



القسم 2: تقييم (3/3) - الإجابة

السؤال 3

في سير عمل التعلم الخاضع للإشراف (Supervised Learning)، لماذا يجب حساب معلمات القياس (scaling parameters) فقط على مجموعة التدريب؟

- أ. لتوفير وقت الحساب.
- ب. لمنع تسرب البيانات (Data Leakage) (اختلاس النظر في مجموعة الاختبار). (✓)
- ج. لأن مجموعة الاختبار لا تحتوي على أرقام.
- د. لا يهم؛ يمكنك استخدام مجموعة البيانات بأكملها.

الشرح

يجب أن تحاكي مجموعة الاختبار البيانات غير المرئية. استخدامها لحساب المتوسط/التباين للقياس يسمح للنموذج "باختلاس النظر" في توزيع بيانات الاختبار، مما يبطل التقييم.



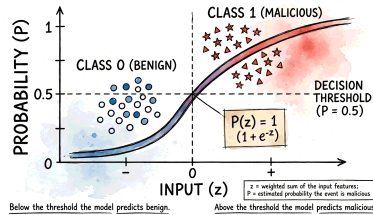
المحتويات

- 1 الذكاء الاصطناعي (AI) وتعلم الآلة (ML) والتعلم العميق (DL)
- 2 نماذج تعلم الآلة (ML) الأساسية
- 3 عائلات نماذج تعلم الآلة (ML) الرئيسية
- 4 مقاييس التقييم للأمن
- 5 تحديات تعلم الآلة (ML) في الأمن السيبراني
- 6 مشروع عملي



الانحدار اللوجستي (Logistic Regression)

خوارزمية تصنيف تقدر الاحتمالات باستخدام دالة سينية (sigmoid function).



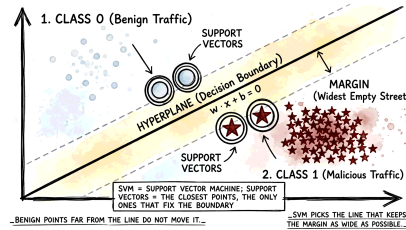
شكل: المنحنى السيني للانحدار اللوجستي (Logistic Regression Sigmoid).

```
from sklearn.linear_model import LogisticRegression
clf = LogisticRegression()
clf.fit(X_train, y_train) # Train
```



آلات المتجهات الداعمة (Support Vector Machines (SVM))

تجد مستوى فائقاً (hyperplane) يزيد الهامش (margin) بين الفئات إلى أقصى حد.



شكل: تزيد آلة المتجهات الداعمة (SVM) الهامش إلى أقصى حد.

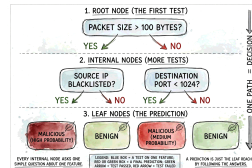
```
from sklearn.svm import SVC
clf = SVC(kernel='linear', C=1.0)
clf.fit(X_train, y_train)
```



أشجار القرار (Decision Trees) و النماذج المجمعة (Ensembles)

أشجار القرار (Decision Trees): هياكل تشبه المخطط الانسيابي.
عرضة للتخصيص المفرط (overfitting).

النماذج المجمعة (Ensembles): تجمع عدة نماذج، مثل الغابة العشوائية (Random Forest) و XGBoost، لتحقيق الاستقرار.



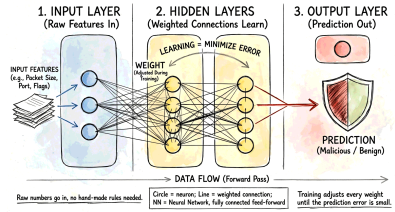
```
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
```

```
rf_model = RandomForestClassifier(n_estimators=100)
rf_model.fit(X_train, y_train)
```



الشبكات العصبية (Neural Networks (NNs))

طبقات من الخلايا العصبية (neurons) المترابطة تتعلم الأوزان (weights) لتقليل الخطأ.



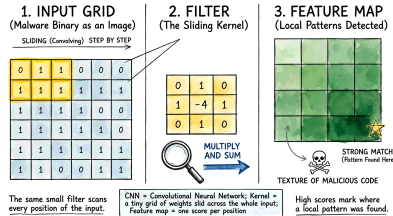
شكل: شبكة عصبية أمامية التغذية (Feed-Forward Neural Network).

```
import torch.nn as nn
# Input: 10 features -> Hidden: 64 neurons -> Output: 2 classes
model = nn.Sequential(nn.Linear(10, 64), nn.ReLU(), nn.Linear(64, 2))
```



الشبكات العصبية الالتفافية (Convolutional Neural Networks (CNNs))

تستخدم المرشحات (filters) [2] للبحث عن أنماط محلية (مثل صور البرامج الضارة أو تسلسلات البايث).



شكل: مرشحات الشبكة العصبية الالتفافية (CNN).

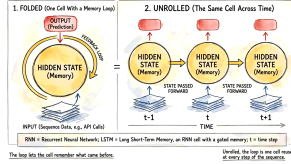
1D Conv Layer for sequence data (e.g., packet bytes)

```
conv_layer = nn.Conv1d(in_channels=1, out_channels=16, kernel_size=3)
```

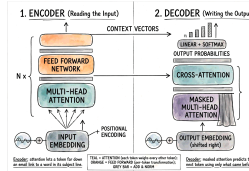


الشبكات العصبية المتكررة (RNNs) والمحولات (Transformers)

الشبكات العصبية المتكررة (RNNs/LSTMs) [1]: معالجة التسلسلات مع الذاكرة (السلاسل الزمنية، استدعاءات واجهة برمجة التطبيقات (API)).



المحولات (Transformers) [3]: تستخدم "الانتباه" (Attention) لربط أجزاء بعيدة من البيانات (مثل رسائل التصيد الإلكتروني (phishing)).



هندسة الميزات (Feature Engineering): النصوص ومعالجة اللغة الطبيعية (NLP)

تحويل النص إلى أرقام. مثال: حقيبة الكلمات (Bag-of-Words) للتصيد.

```
1 from sklearn.feature_extraction.text import CountVectorizer
2 import pandas as pd
3
4 # 1. Raw Data (Email Subjects)
5 emails = ["Urgent: Update password", "Meeting agenda", "Win free iPhone"]
6
7 # 2. Transform to Numbers
8 vectorizer = CountVectorizer(stop_words='english')
9 X = vectorizer.fit_transform(emails)
10
11 # 3. Result
12 print(pd.DataFrame(X.toarray(), columns=vectorizer.get_feature_names_out()))
13 # Columns: 'urgent', 'win', 'meeting'... Values: 1 or 0
```



هندسة الميزات (Feature Engineering): الزمني والإنتروپيا

الزمني (Temporal): استخراج 'ساعة اليوم'. الإنتروپيا (Entropy): قياس العشوائية (برامج ضارة مقابل شرعية).

```

1 # Temporal Example
2 df['timestamp'] = pd.to_datetime(df['timestamp'])
3 df['hour'] = df['timestamp'].dt.hour # Attackers often work at night
4
5 # Entropy Example (Malware Detection)
6 from scipy.stats import entropy
7 def calc_entropy(data):
8     probs = [count / len(data) for count in collections.Counter(data).values()]
9     return entropy(probs, base=2)
10 # High Entropy = Encrypted/Packed Malware

```



تطبيع البيانات (Data Normalization)

تخجيم الميزات (feature scaling) إلى نطاق مشترك (0-1) حتى لا تهيمن الأرقام الكبيرة.

```

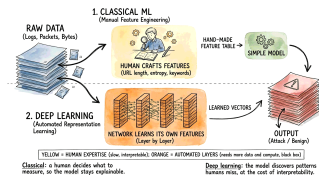
1 from sklearn.preprocessing import MinMaxScaler
2 import pandas as pd
3
4 # Unscaled: 'Bytes' (10000) dominates 'Hour' (23)
5 df = pd.DataFrame({'bytes': [10000], 'hour': [23]})
6
7 # Apply Scaling
8 scaler = MinMaxScaler()
9 df_scaled = pd.DataFrame(scaler.fit_transform(df), columns=df.columns)
10 # Result: Both features are now between 0.0 and 1.0

```



تعلم التمثيل (Representation Learning)

يقوم التعلم العميق (Deep Learning) بأتمتة استخراج الميزات باستخدام التضمينات (Embeddings).



شكل: هندسة الميزات اليدوية مقابل تعلم التمثيل (Feature Engineering vs. Representation Learning).

```
1 import torch.nn as nn
2 # Learn a vector of size 3 for every API call automatically
3 embedding_layer = nn.Embedding(num_embeddings=10, embedding_dim=3)
4 learned_features = embedding_layer(raw_sequence)
```



القسم 3: تقييم (1/3)

السؤال 1

أي خوارزمية كلاسيكية تستخدم دالة "سينية" (sigmoid) لإخراج احتمال بين 0 و 1؟

- أ. شجرة القرار (Decision Tree).
- ب. آلة المتجهات الداعمة (Support Vector Machine (SVM)).
- ج. الانحدار اللوجستي (Logistic Regression).
- د. التجميع بخوارزمية المتوسطات (K-Means Clustering).



القسم 3: تقييم (1/3) - الإجابة

السؤال 1

أي خوارزمية كلاسيكية تستخدم دالة "سينية" (sigmoid) لإخراج احتمال بين 0 و 1؟

- أ. شجرة القرار (Decision Tree).
- ب. آلة المتجهات الداعمة ((Support Vector Machine (SVM)).
- ج. الانحدار اللوجستي (Logistic Regression) (✓).
- د. التجميع بخوارزمية المتوسطات (K-Means Clustering).

الشرح

على الرغم من اسمها، فإن الانحدار اللوجستي (Logistic Regression) هو مصنف (classifier) يطابق البيانات بمنحنى سيني (على شكل S)، مما ينتج عنه مخرجات احتمالية مناسبة للتصنيف الثنائي (binary classification).



القسم 3: تقييم (2/3)

السؤال 2

ما هي الميزة الأساسية للغابة العشوائية (Random Forest) مقارنة بشجرة قرار (Decision Tree) واحدة؟

- أ. أسرع في التدريب.
- ب. أسهل في التفسير بصرياً.
- ج. تقلل من التخصيص المفرط (overfitting) وتحسن الاستقرار.
- د. لا تتطلب بيانات مسماة.



القسم 3: تقييم (2/3) - الإجابة

السؤال 2

- ما هي الميزة الأساسية للغابة العشوائية (Random Forest) مقارنة بشجرة قرار (Decision Tree) واحدة؟
- أ. أسرع في التدريب.
 - ب. أسهل في التفسير بصرياً.
 - ج. تقلل من التخصيص المفرط (overfitting) وتحسن الاستقرار. (✓)
 - د. لا تتطلب بيانات مسماة.

الشرح

غالباً ما تحفظ أشجار القرار الفردية بيانات التدريب (تخصيص مفرط). تقوم الغابة العشوائية (Random Forest) بمتوسط تنبؤات العديد من الأشجار، مما يخفف الأخطاء ويحسن التعميم (generalization).



القسم 3: تقييم (3/3)

السؤال 3

ما هو الغرض من تطبيع البيانات (Scaling / Data Normalization)؟

- أ. لتحويل النص إلى أرقام.
- ب. لضمان عدم هيمنة الميزات ذات القيم الكبيرة (مثل البايت) على الميزات ذات القيم الصغيرة (مثل الساعة).
- ج. لإزالة القيم الفارغة من مجموعة البيانات.
- د. لتشفير البيانات من أجل الخصوصية.



القسم 3: تقييم (3/3) - الإجابة

السؤال 3

ما هو الغرض من تطبيع البيانات (Scaling / Data Normalization)؟

- أ. لتحويل النص إلى أرقام.
- ب. لضمان عدم هيمنة الميزات ذات القيم الكبيرة (مثل البايت) على الميزات ذات القيم الصغيرة (مثل الساعة). (✓)
- ج. لإزالة القيم الفارغة من مجموعة البيانات.
- د. لتشفير البيانات من أجل الخصوصية.

الشرح

تعتمد خوارزميات تعلم الآلة (ML) غالباً على المسافة أو الحجم. بدون التحجيم (scaling)، ستتغلب ميزة تتراوح من 0-10000 رياضياً على ميزة تتراوح من 0-1، مما يؤدي إلى ضعف أداء النموذج.



المحتويات

- 1 الذكاء الاصطناعي (AI) وتعلم الآلة (ML) والتعلم العميق (DL)
- 2 نماذج تعلم الآلة (ML) الأساسية
- 3 عائلات نماذج تعلم الآلة (ML) الرئيسية
- 4 مقاييس التقييم للأمن
- 5 تحديات تعلم الآلة (ML) في الأمن السيبراني
- 6 مشروع عملي



مقاييس التصنيف (Classification Metrics)

الدقة (Accuracy) مضللة في الأمن بسبب مغالطة المعدل الأساسي (Base Rate Fallacy) (البيانات غير المتوازنة).

- الدقة (Precision): $TP / (TP + FP)$ (موثوقية التنبؤات).
- الاستدعاء (Recall): $TP / (TP + FN)$ (القدرة على اكتشاف الهجمات).
- مقياس إف-1 (F1-Score): المتوسط التوافقي للدقة والاستدعاء.

حقيقي: حميد	حقيقي: هجوم	
FP	TP	تنبؤ: هجوم
TN	FN	تنبؤ: حميد

جدول: مصفوفة الارتباك (Confusion Matrix)



الكود: تقييم المصنفات (Classifiers)

```

1 from sklearn.metrics import classification_report, confusion_matrix
2 import numpy as np
3
4 # 1. Ground Truth (Most are Benign/0)
5 y_true = np.array([0, 0, 0, 0, 0, 0, 0, 1, 1, 1])
6
7 # 2. Model Predictions
8 y_pred = np.array([0, 0, 0, 0, 0, 0, 1, 1, 1, 0])
9
10 # 3. Evaluation
11 print("Confusion Matrix:\n", confusion_matrix(y_true, y_pred))
12 print(classification_report(y_true, y_pred, target_names=['Benign', 'Attack']))
13 # Precision (Attack) = 0.67 (2 real out of 3 alerts)
14 # Recall (Attack) = 0.67 (Caught 2 out of 3 attacks)

```



مقاييس الانحدار والتجميع

الانحدار (Regression): يعاقب متوسط مربع الخطأ (MSE) الأخطاء الكبيرة أكثر من متوسط الخطأ المطلق (MAE).

```
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(y_true, y_pred) # High if outliers exist
```

التجميع (Clustering): يقيس معامل الظل (Silhouette Score) فصل الكَلّة (-1 إلى 1).

```
from sklearn.metrics import silhouette_score
score = silhouette_score(X_features, labels) # Closer to 1 is better
```



ضبط العتبة (Threshold Tuning)

- يؤثر ضبط حدود القرار (decision boundary) على الدقة (Precision) والاستدعاء (Recall).
- عتبة منخفضة (<2.0): استدعاء عالي (مصاب بجنون العظمة/صيد التهديدات (threat hunting)).
- عتبة عالية (<9.0): دقة عالية (الحظر الآلي).

```

1 # Scenario A: Paranoid Mode (Catch everything)
2 y_pred_paranoid = (y_proba > 0.3).astype(int)
3
4 # Scenario B: Conservative Mode (Avoid false alarms)
5 y_pred_safe = (y_proba > 0.8).astype(int)

```



القسم 4: تقييم (1/3)

السؤال 1

- في مجموعة بيانات تحتوي على 99 حزمة حميدة وحزمة واحدة ضارة، يتنبأ النموذج بـ "حميد" لكل شيء. ما هي المشكلة؟
- أ. الدقة (accuracy) هي 0%.
 - ب. الدقة هي 99%، لكن النموذج فاته الهجوم (Base Rate Fallacy).
 - ج. النموذج يعاني من التخصيص المفرط (overfitting).
 - د. الدقة (Precision) هي 100%.



القسم 4: تقييم (1/3) - الإجابة

السؤال 1

- في مجموعة بيانات تحتوي على 99 حزمة حميدة وحزمة واحدة ضارة، يتنبأ النموذج بـ "حميد" لكل شيء. ما هي المشكلة؟
- أ. الدقة (accuracy) هي 0%.
 - ب. الدقة هي 99%، لكن النموذج فاته الهجوم (Base Rate Fallacy). (✓)
 - ج. النموذج يعاني من التخصيص المفرط (overfitting).
 - د. الدقة (Precision) هي 100%.

الشرح

الدقة العالية (accuracy) لا معنى لها في بيانات الأمان غير المتوازنة إذا تم تجاهل فئة الأقلية (minority class) تماماً. سيكون الاستدعاء (Recall) 0%.



القسم 4: تقييم (2/3)

السؤال 2

أي مقياس يمثل النسبة المئوية للهجمات الفعلية التي تم اكتشافها بنجاح؟

- أ. الدقة (Precision).
- ب. الاستدعاء (Recall).
- ج. الدقة (Accuracy).
- د. متوسط مربع الخطأ (Mean Squared Error).



القسم 4: تقييم (2/3) - الإجابة

السؤال 2

أي مقياس يمثل النسبة المئوية للهجمات الفعلية التي تم اكتشافها بنجاح؟

- أ. الدقة (Precision).
- ب. الاستدعاء (Recall). (✓)
- ج. الدقة (Accuracy).
- د. متوسط مربع الخطأ (Mean Squared Error).

الشرح

يقيس الاستدعاء (Recall) أو الحساسية (sensitivity) جزء الحالات ذات الصلة (الهجمات) التي تم استردادها بواسطة النموذج.



القسم 4: تقييم (3/3)

السؤال 3

- إذا قمت بخفض عتبة القرار (decision threshold) للمصنف (على سبيل المثال، من 5.0 إلى 2.0)، فماذا يحدث عادةً؟
- أ. يزداد الاستدعاء (Recall)، ولكن تزداد الإيجابيات الخاطئة (False Positives) أيضاً.
 - ب. تزداد الدقة (Precision)، ويزداد الاستدعاء.
 - ج. ينخفض الاستدعاء، وتقل الإيجابيات الخاطئة.
 - د. يتوقف النموذج عن العمل.



القسم 4: تقييم (3/3) - الإجابة

السؤال 3

إذا قمنا بخفض عتبة القرار (decision threshold) للمصنف (على سبيل المثال، من 5.0 إلى 2.0)، فماذا يحدث عادةً؟

- أ. يزداد الاستدعاء (Recall)، ولكن تزداد الإيجابيات الخاطئة (False Positives) أيضًا. (✓)
- ب. تزداد الدقة (Precision)، ويزداد الاستدعاء.
- ج. ينخفض الاستدعاء، وتقل الإيجابيات الخاطئة.
- د. يتوقف النموذج عن العمل.

الشرح

خفض العتبة يجعل النموذج أكثر "جنون عظمة". يكتشف المزيد من الهجمات (استدعاء أعلى) ولكنه يضع علامة أيضًا على المزيد من العناصر الحميدة على أنها ضارة (المزيد من الإيجابيات الخاطئة).



المحتويات

- 1 الذكاء الاصطناعي (AI) وتعلم الآلة (ML) والتعلم العميق (DL)
- 2 نماذج تعلم الآلة (ML) الأساسية
- 3 عائلات نماذج تعلم الآلة (ML) الرئيسية
- 4 مقاييس التقييم للأمن
- 5 تحديات تعلم الآلة (ML) في الأمن السيبراني
- 6 مشروع عملي



البيانات غير المتوازنة (Imbalanced Data)

بيانات الهجوم نادرة. تصبح النماذج كسولة.

الحل المحتمل: تقنية زيادة عينات الأقلية الاصطناعية (SMOTE (Synthetic Minority Over-sampling Technique)).

```

1 from imblearn.over_sampling import SMOTE
2 # 1. Imbalanced Dataset (99 Benign, 1 Attack)
3 X = [[1.0, 2.0]] * 99 + [[9.0, 9.0]] * 1
4 y = [0] * 99 + [1] * 1
5
6 # 2. Apply SMOTE
7 smote = SMOTE(k_neighbors=1)
8 X_res, y_res = smote.fit_resample(X, y)
9 # Result: Classes are now balanced (99 vs 99)

```



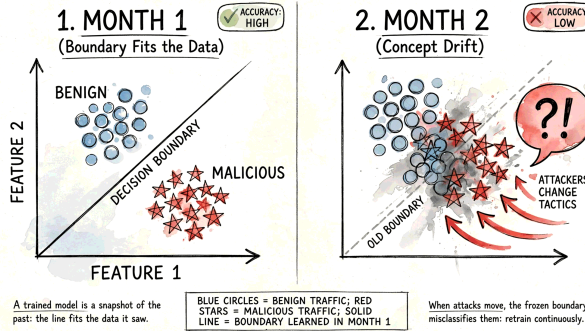
التشفير، الانجراف، والهجمات المضادة I

- التشفير (Encryption): ("التحول إلى الظلام") 90% من حركة المرور مشفرة.
- الحل المحتمل: تحليل حركة المرور المشفرة (Encrypted Traffic Analysis - ETA) باستخدام البيانات الوصفية (metadata) مثل التوقيت وأحجام الحزم بدلاً من الحمولة (payload).
- تسلسل أطوال الحزم.
- أوقات ما بين الوصول (inter-arrival times): (التوقف المؤقت بين الحزم).
- أنماط المصافحة (handshake)، أي أجنحة التشفير (cipher suites) المعروضة.



التشفير، الانجراف، والهجمات المضادة II

- انحراف المفهوم (Concept Drift): تتغير أنماط الهجوم بمرور الوقت (مثل متغيرات برامج الفدية (ransomware) الجديدة).



شكل: انحراف المفهوم، مع تغيير المهاجمين للتكتيكات، تصبح حدود القرار القديمة قديمة.



التشفير، الانجراف، والهجمات المضادة III

- الحل المحتمل: إعادة التدريب المستمر (continuous retraining).



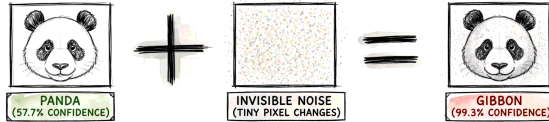
التشفير، الانجراف، والهجمات المضادة IV

- الأمثلة المضادة (Adversarial Examples): إضافة ضوضاء لخداع النموذج.

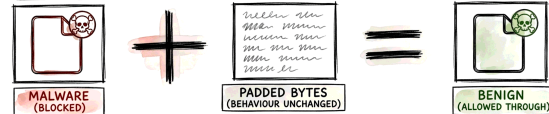


التشفير، الانجراف، والهجمات المضادة V

1. IMAGE ATTACK (The Famous Panda):



2. MALWARE ATTACK (Same Trick on a File):



ADVERSARIAL EXAMPLE = AN INPUT CRAFTED SO A TINY CHANGE PUSHES IT ACROSS THE DECISION BOUNDARY

THE CHANGE IS INVISIBLE TO A HUMAN BUT FLIPS THE MODEL'S ANSWER WITH HIGH CONFIDENCE.

THE SAME IDEA LETS MALWARE SLIP PAST A CLASSIFIER WITHOUT CHANGING WHAT IT DOES.

شكل: الهجمات المضادة (Adversarial Attacks).



الذكاء الاصطناعي القابل للتفسير (Explainable AI - XAI)

نماذج التعلم العميق (Deep Learning) هي "صناديق سوداء" (black boxes). إذا قام جدار الحماية بحظر كمبيوتر محمول للرئيس التنفيذي قائلاً "99% ضار"، يحتاج محلل الأمن إلى معرفة السبب للتحقق من التنبيه. استخدام قيم شابلي التفسيرية (SHAP) لشرح نماذج "الصندوق الأسود".

```

1 import shap
2 # Use SHAP to explain a single prediction
3 explainer = shap.Explainer(model)
4 shap_values = explainer(X_test)
5
6 # Visualization tells analyst which feature caused the alert
7 shap.plots.waterfall(shap_values[0])

```



القسم 5: تقييم (1/3)

السؤال 1

فيما تُستخدم تقنية زيادة عينات الأقلية الاصطناعية (SMOTE)؟

- أ. تشفير البيانات.
- ب. شرح تنبؤات النموذج.
- ج. موازنة مجموعات البيانات عن طريق توليد عينات أقلية اصطناعية.
- د. الكشف عن انحراف المفهوم (Concept Drift).



القسم 5: تقييم (1/3) - الإجابة

السؤال 1

فيما تُستخدم تقنية زيادة عينات الأقلية الاصطناعية (SMOTE)؟

- أ. تشفير البيانات.
- ب. شرح تنبؤات النموذج.
- ج. موازنة مجموعات البيانات عن طريق توليد عينات أقلية اصطناعية. (✓)
- د. الكشف عن انحراف المفهوم (Concept Drift).

الشرح

تقوم تقنية زيادة عينات الأقلية الاصطناعية (SMOTE (Synthetic Minority Over-sampling Technique بإنشاء أمثلة مزيفة ولكن واقعية لفئة الأقلية (مثل الهجمات) لمنع النموذج من تجاهلها.



القسم 5: تقييم (2/3)

السؤال 2

- كيف يمكن لنماذج تعلم الآلة (ML) اكتشاف الهجمات في حركة المرور المشفرة (HTTPS) إذا لم تتمكن من قراءة الحمولة (payload)؟
- أ. يكسرون التشفير باستخدام القوة الغاشمة.
 - ب. يحللون البيانات الوصفية (metadata) مثل توقيت الحزمة والأحجام وأنماط المصافحة.
 - ج. لا يمكنهم ذلك؛ تعلم الآلة (ML) عديم الفائدة ضد التشفير.
 - د. يطلبون من المستخدم كلمة المرور.



القسم 5: تقييم (2/3) - الإجابة

السؤال 2

- كيف يمكن لنماذج تعلم الآلة (ML) اكتشاف الهجمات في حركة المرور المشفرة (HTTPS) إذا لم تتمكن من قراءة الحمولة (payload)؟
- أ. يكسرون التشفير باستخدام القوة الغاشمة.
 - ب. يحللون البيانات الوصفية (metadata) مثل توقيت الحزمة والأحجام وأنماط المصافحة. (✓)
 - ج. لا يمكنهم ذلك؛ تعلم الآلة (ML) عديم الفائدة ضد التشفير.
 - د. يطلبون من المستخدم كلمة المرور.

الشرح

يعتمد تحليل حركة المرور المشفرة (ETA) على معلومات القناة الجانبية (side-channel) (البيانات الوصفية) التي تظل مرئية حتى عندما يتم تشفير المحتوى.



القسم 5: تقييم (3/3)

السؤال 3

ما هو المثال المضاد (Adversarial Example)؟

- أ. خطأ برمجي في مكتبة تعلم الآلة (ML).
- ب. إدخال بضوضاء مصممة بعناية لخداع النموذج.
- ج. عينة من البيانات تم تصنيفها بشكل غير صحيح من قبل إنسان.
- د. نموذج يتنافس ضد نموذج آخر.



القسم 5: تقييم (3/3) - الإجابة

السؤال 3

ما هو المثال المضاد (Adversarial Example)؟

أ. خطأ برمجي في مكتبة تعلم الآلة (ML).

ب. إدخال بضوضاء مصممة بعناية لخداع النموذج. (✓)

ج. عينة من البيانات تم تصنيفها بشكل غير صحيح من قبل إنسان.

د. نموذج يتنافس ضد نموذج آخر.

الشرح

تستغل الأمثلة المضادة الحدود الرياضية للنموذج، باستخدام اضطرابات (perturbations) غير مرئية لفرض تصنيف خاطئ (على سبيل المثال، جعل البرامج الضارة تبدو حميدة).



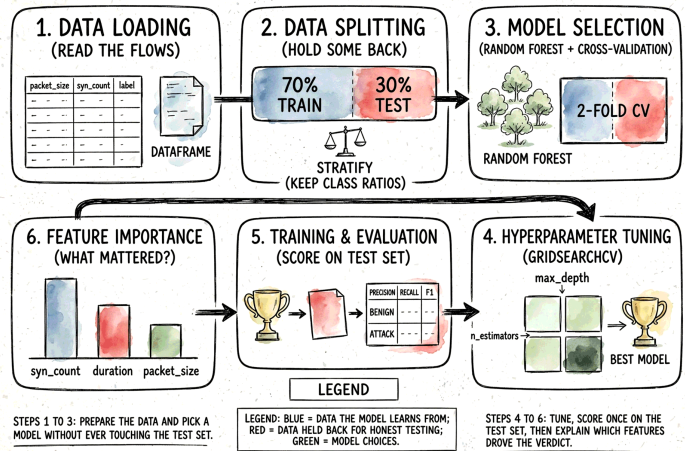
المحتويات

- 1 الذكاء الاصطناعي (AI) وتعلم الآلة (ML) والتعلم العميق (DL)
- 2 نماذج تعلم الآلة (ML) الأساسية
- 3 عائلات نماذج تعلم الآلة (ML) الرئيسية
- 4 مقاييس التقييم للأمن
- 5 تحديات تعلم الآلة (ML) في الأمن السيبراني
- 6 مشروع عملي



المشروع: سير عمل تصنيف التهديدات (Threat Classification)

سنقوم ببناء نظام بسيط لكشف التسلل للشبكة (NIDS) باستخدام الغابة العشوائية (Random Forest).



كود المشروع: التحميل والتقسيم

```

1 import pandas as pd
2 from sklearn.model_selection import train_test_split
3
4 # 1. Create Data (Features: Packet Size, SYN Count, Duration)
5 data = {
6     'packet_size': [64, 1500, 64, 200, 64, 1500, 500, 64, 1500, 64],
7     'syn_count': [0, 0, 1, 50, 0, 0, 20, 0, 0, 100],
8     'duration': [0.1, 5.2, 0.1, 1.5, 0.2, 10.0, 0.5, 0.1, 8.5, 2.0],
9     'label': [0, 0, 0, 1, 0, 0, 1, 0, 0, 1] # 1 = Attack
10 }
11 X = pd.DataFrame(data).drop('label', axis=1)
12 y = pd.DataFrame(data)['label']
13
14 # 2. Split (Stratify ensures attack ratio is preserved)
15 X_train, X_test, y_train, y_test = train_test_split(
16     X, y, test_size=0.3, random_state=42, stratify=y
17 )

```



```

1 from sklearn.ensemble import RandomForestClassifier
2 from sklearn.model_selection import GridSearchCV
3
4 # 3 & 4. Grid Search with Cross-Validation
5 param_grid = {'n_estimators': [10, 50], 'max_depth': [None, 5]}
6
7 grid_search = GridSearchCV(
8     RandomForestClassifier(random_state=42),
9     param_grid,
10    cv=2,
11    scoring='f1' # Optimize for F1
12 )
13 grid_search.fit(X_train, y_train)
14 best_model = grid_search.best_estimator_
15
16 print(f"Best Parameters: {grid_search.best_params_}")

```



كود المشروع: التقييم والتفسير

```

1 from sklearn.metrics import classification_report
2 import matplotlib.pyplot as plt
3 import numpy as np
4
5 # 5. Evaluation
6 y_pred = best_model.predict(X_test)
7 print(classification_report(y_test, y_pred, target_names=['Benign', 'Attack']))
8
9 # 6. Feature Importance
10 importances = best_model.feature_importances_
11 indices = np.argsort(importances)[::-1]
12
13 plt.figure(figsize=(8, 4))
14 plt.bar(range(X.shape[1]), importances[indices])
15 plt.xticks(range(X.shape[1]), X.columns[indices])
16 plt.title("Feature Importance"); plt.show()

```



الخلاصة

ملخص

- التسلسل الهرمي: الذكاء الاصطناعي (AI) يحتوي تعلم الآلة (ML)، الذي يحتوي التعلم العميق (DL).
- النماذج: الخاضع للإشراف (Supervised) لبيانات مسماة، وغير الخاضع للإشراف (Unsupervised) لبيانات غير مسماة، والتعلم المعزز (RL) بالمكافآت.
- سير العمل: قم بتقسيم البيانات أولاً لتجنب التسرب (leakage). تدريب، ضبط، تقييم.
- المقاييس: استخدم الدقة والاستدعاء ومقياس إف-1 (Precision/Recall/F1) للأمن، وليس فقط الدقة (Accuracy).
- التحديات: يجب التعامل مع البيانات غير المتوازنة، والتشفير، والتهديدات المضادة.

الخطوات التالية

مع هذا الأساس وسير العمل العملي، أنت مستعد لاستكشاف أنظمة الدفاع المتقدمة.



المراجع I

- [1] Sepp Hochreiter and Jürgen Schmidhuber.
Long short-term memory.
Neural computation, 9(8):1735–1780, 1997.
- [2] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner.
Gradient-based learning applied to document recognition.
Proceedings of the IEEE, 86(11):2278–2324, 1998.
- [3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin.
Attention is all you need.
Advances in neural information processing systems, 30, 2017.



المختصرات والمصطلحات I

AI الذكاء الاصطناعي **Artificial Intelligence**: محاكاة الذكاء البشري في الآلات، المبرمجة للتعلم والاستدلال وأتمتة المهام، خاصة لتحليل البيانات واتخاذ القرارات.

API واجهة برمجة التطبيقات **Application Programming Interface**: مجموعة من القواعد والأدوات التي تسمح لتطبيقات برمجية مختلفة بالتواصل مع بعضها البعض. يمكن أن تكون واجهات برمجة التطبيقات **APIs** غير الآمنة ثغرة أمنية **vulnerability** رئيسية.

ATT&CK التكتيكات والتقنيات والمعرفة العامة للخصم **Adversarial Tactics, Techniques, and Common Knowledge**: قاعدة معارف **knowledge base** وإطار عمل **framework** يمكن الوصول إليه عالمياً من **MITRE** يصنف إجراءات الخصم بناءً على ملاحظات من العالم الحقيقي.

Availability التوافر **Availability**: مبدأ أساسي في ثلاث السي أي إيه **CIA Triad** يضمن أن الأنظمة والبيانات متاحة للمستخدمين المصرح لهم عند الحاجة إليها.

C2 القيادة والتحكم **Command and Control**: قناة الاتصال التي يستخدمها المهاجم **attacker** للحفاظ على السيطرة على نظام مخترق، وإصدار الأوامر، واستخراج البيانات.



المختصرات والمصطلحات II

Checksum المجموع الاختباري **Checksum**: كلمة صغيرة من البيانات مشتقة من كلمة أكبر، تستخدم للكشف عن الأخطاء أو التغييرات غير المصرح بها في البيانات الأصلية، وبالتالي ضمان السلامة **integrity**.

Triad CIA ثلاث السي أي إيه **Triad CIA**: نموذج أمني أساسي يمثل الأهداف الأساسية الثلاثة لأمن المعلومات: السرية **Confidentiality**، والسلامة **Integrity**، والتوافر **Availability**.

Confidentiality السرية **Confidentiality**: مبدأ أساسي في ثلاث السي أي إيه **Triad CIA** يضمن أن المعلومات الحساسة متاحة فقط للأفراد المصرح لهم.

Chain Kill Cyber سلسلة القتل السيبراني **Cyber Kill Chain**: نموذج يصف التسلسل النموذجي للمراحل التي يجب على الخصم إكمالها لتنفيذ هجوم سيبراني.

Denial-of-Service (DoS) هجوم حجب الخدمة **Denial-of-Service (DoS)**: هجوم مصمم لجعل نظام أو خدمة غير متاحة للمستخدمين الشرعيين عن طريق إغراقها بحركة مرور أو طلبات.

Signature Digital التوقيع الرقمي **Digital Signature**: تقنية تشفير **cryptographic** تستخدم للتحقق من أصالة وسلامة رسالة أو ملف رقمي، مما يؤكد أنه لم يتم تغييره.



III المختصرات والمصطلحات

- EDR** كشف واستجابة النقاط النهائية **Endpoint Detection and Response**: فئة من الأدوات الأمنية التي تراقب باستمرار أجهزة المستخدمين النهائيين (مثل أجهزة الكمبيوتر المحمولة والخوادم) للكشف عن التهديدات المتقدمة والاستجابة لها.
- ELF** تنسيق قابل للتنفيذ والربط **Executable and Linkable Format**: تنسيق الملفات القياسي للبرامج والمكتبات القابلة للتنفيذ على أنظمة لينكس ومعظم أنظمة التشغيل الشبيهة بيونكس.
- Encryption** التشفير **Encryption**: عملية تحويل البيانات إلى تنسيق غير قابل للقراءة (نص مشفر **ciphertext**) لحماية سريتها **confidentiality**. يتطلب الأمر مفتاحاً لفك تشفيرها وإعادةتها إلى بيانات قابلة للقراءة.
- Positive False** نتيجة إيجابية خاطئة **False Positive**: تنبيه من نظام أمني يحدد بشكل غير صحيح نشاطاً طبيعياً حميداً على أنه ضار.
- IAM** إدارة الهوية والوصول **Identity and Access Management**: فرع أمني يضمن أن الأفراد المناسبين لديهم المستوى المناسب من الوصول إلى موارد التكنولوجيا.
- IDS** نظام كشف التسلل **Intrusion Detection System**: أداة أمنية تراقب أنشطة الشبكة أو النظام بحثاً عن أنماط ضارة أو انتهاكات للسياسات وتقوم بالإبلاغ عنها.



المختصرات والمصطلحات IV

Integrity السلامة **Integrity**: مبدأ أساسي في ثلاث السي أي إيه **CIA Triad** يركز على الحفاظ على دقة وموثوقية البيانات، مما يضمن عدم تغييرها دون إذن.

IOCs مؤشرات الاختراق **Indicators of Compromise**: أدلة جنائية أو أجزاء من البيانات (مثل تجزئات الملفات أو عناوين IP أو أسماء النطاقات) تشير إلى حدوث خرق أمني.

MFA المصادقة متعددة العوامل **Multi-Factor Authentication**: آلية أمنية تتطلب من المستخدم تقديم نوعين أو أكثر من عوامل التحقق المختلفة للوصول إلى حساب أو نظام.

ATT&CK MITRE انظر **ATT&CK**.

NIST المعهد الوطني للمعايير والتكنولوجيا **National Institute of Standards and Technology**: وكالة حكومية أمريكية تضع معايير وأطر عمل مؤثرة للأمن السيبراني.

PCAP التقاط الحزم **Packet Capture**: ملف يحتوي على البيانات الأولية للحزم الملتقطة من شبكة، مما يوفر سجلاً مفصلاً لحركة مرور الشبكة للتحليل.



المختصرات والمصطلحات V

- PE** قابل للتنفيذ المحمول **Portable Executable**: تنسيق الملفات القياسي للبرامج والمكتبات القابلة للتنفيذ المستخدمة بواسطة نظام تشغيل ويندوز.
- SAST** اختبار أمان التطبيقات الثابت **Static Application Security Testing**: عملية تحليل الكود المصدري للتطبيق بحثاً عن ثغرات أمنية دون تشغيل الكود فعلياً.
- Control Security** ضابط أمني **Security Control**: إجراء وقائي أو مضاد، مثل جدار الحماية **firewall** أو التشفير **encryption**، يتم وضعه لتجنب أو اكتشاف أو تقليل المخاطر الأمنية.
- Telemetry Security** بيانات الأمان عن بعد **Security Telemetry**: مصطلح عام لمجموعة واسعة من بيانات الأمان (مثل السجلات، تدفقات الشبكة) التي يتم جمعها من جميع أنحاء البيئة الرقمية للمراقبة والتحليل.
- SOC** مركز عمليات الأمان **Security Operations Center**: فريق مركزي، إلى جانب التكنولوجيا والعمليات المرتبطة به، مسؤول عن المراقبة المستمرة والدفاع عن منظمة ضد التهديدات السيبرانية.
- Spoofing** الانتحال **Spoofing**: هجوم ينجح فيه شخص أو برنامج في انتحال شخصية كيان آخر عن طريق تزيف البيانات، مثل مرسل بريد إلكتروني أو عنوان IP.



المختصرات والمصطلحات VI

Injection SQL

حقن إس كيو إل SQL Injection: ثغرة أمنية في الويب تسمح للمهاجم بالتدخل في الاستعلامات التي يرسلها التطبيق إلى قاعدة بياناته، غالباً لاستخراج البيانات.

STRIDE

ستريد STRIDE: إطار عمل لنمذجة التهديدات يصنف التهديدات المحتملة إلى ستة أنواع: الانتحال Spoofing، والعبث Tampering، والتنصل Repudiation، وكشف المعلومات Information Disclosure، وحجب الخدمة Denial of Service، ورفع الامتيازات Elevation of Privilege.

Sysmon

سيسمون Sysmon: أداة من مايكروسوفت توفر تسجيلاً مفصلاً وعالي الجودة لنشاط النظام على ويندوز، مثل إنشاء العمليات واتصالات الشبكة.

Tampering

العبث Tampering: التعديل غير المصرح به للبيانات، وهو تهديد مباشر لسلامة البيانات data integrity.

Intelligence Threat

استخبارات التهديدات Threat Intelligence: معلومات وبيانات حول التهديدات السيبرانية الحالية أو الناشئة، بما في ذلك تفاصيل عن المهاجمين وأساليبهم، تستخدم لإرشاد القرارات الأمنية.

Positive True

نتيجة إيجابية حقيقية True Positive: تنبيه من نظام أمني يحدد بشكل صحيح نشاطاً ضاراً حقيقياً أو تهديداً أمنياً.



المختصرات والمصطلحات VII

- TTPs** التكتيكات والتقنيات والإجراءات **Tactics, Techniques, and Procedures**: مصطلح يستخدم في استخبارات التهديدات لوصف سلوك وأساليب الخصم أثناء الهجوم.
- XSS** البرمجة النصية عبر المواقع **Cross-Site Scripting**: نوع من الثغرات الأمنية في الويب يسمح للمهاجم بحقن نصوص برمجية ضارة في صفحات الويب التي يشاهدها المستخدمون الآخرون.



شكراً لكم!

