HANDOUT 5 - WEEK 6 READING

Unsupervised Learning and Clustering

CIS 350: AI for Cybersecurity - Smart Defense for the Digital Business

Colorado State University - Computer Information Systems

How to use this reading

This handout supports Week 6. It explains unsupervised learning - finding patterns in data with NO labels - and its two workhorses for security: CLUSTERING (grouping similar traffic, users, or malware) and ANOMALY DETECTION (flagging what doesn't fit). The figures and code match Lab 5 and the lectures; the review questions map to Quiz 5 and the lab.

Estimated reading time: 20-25 minutes. No prior coding background needed.

1. Learning Without Labels

In Weeks 3-4 every model learned from LABELED examples (spam vs. ham). Unsupervised learning is different: the data has only features - no answer column. Instead of predicting a known label, we ask what STRUCTURE is already in the data. This matters because labeled data is expensive, and brand-new attacks have no label yet. Two unsupervised tools dominate security: clustering (group the similar) and anomaly detection (flag the odd).

Why unsupervised learning is powerful in security

Supervised models only catch what they were taught. Unsupervised methods learn what NORMAL looks like from raw, unlabeled data - so they can surface NEW, never-seen behavior. That is exactly what you need for zero-day attacks, insider threats, and novel malware.

2. Clustering: Grouping Similar Behavior

Clustering automatically groups records that are similar - where 'similar' means close together when you plot them by their features. Each group (cluster) has a CENTER, its typical member. In security, a cluster is a behavior group: routine web traffic, bulk transfers, or a small odd group worth investigating. The most common method is k-means.

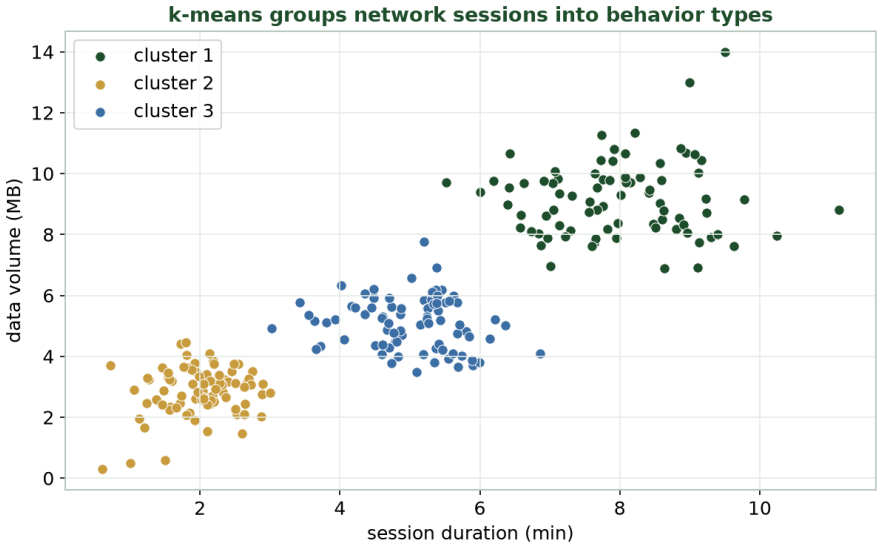

Figure 1. k-means groups network sessions into behavior types (here, three). You then give each group a plain-language name from its center.

- k-means recipe: pick k, assign each point to the nearest center, move centers, repeat until stable.
- The output is k groups + their centers; cluster NUMBERS (0,1,2) are arbitrary names, not rankings.
- Clusters describe behavior - a human decides which groups matter.

3. Two Decisions: How Many Groups, and Scaling

Clustering's value depends on two choices you make. First, HOW MANY groups (k)? Use the ELBOW

METHOD: run k-means for a range of k, measure the within-cluster spread (inertia), and pick the 'elbow' where extra groups stop helping much. Second, WHICH features and on what SCALE?

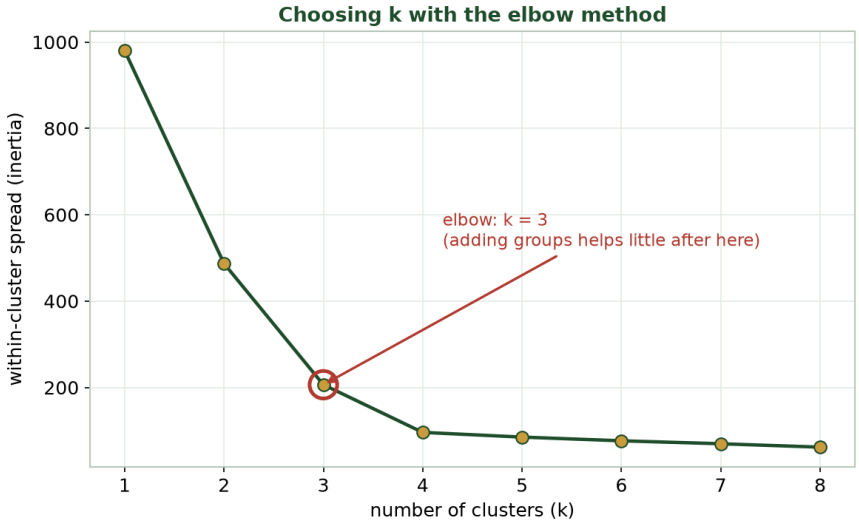

Figure 2. Inertia drops steeply then flattens; the elbow (here k=3) is a sensible choice.

Always scale first

k-means measures DISTANCE, so a feature with a huge range (bytes) drowns out a small one (duration) and the clusters become meaningless. STANDARDIZE features first so each contributes fairly. This one step often matters more than the algorithm you pick.

4. Hierarchical Clustering and Dendrograms

Hierarchical clustering starts with every record as its own cluster, then repeatedly merges the two closest, building a TREE of merges called a dendrogram. Unlike k-means, you do not choose k up front - you CUT the tree afterward at a height that gives useful groups. The height of a merge shows how different the groups were (higher = more different).

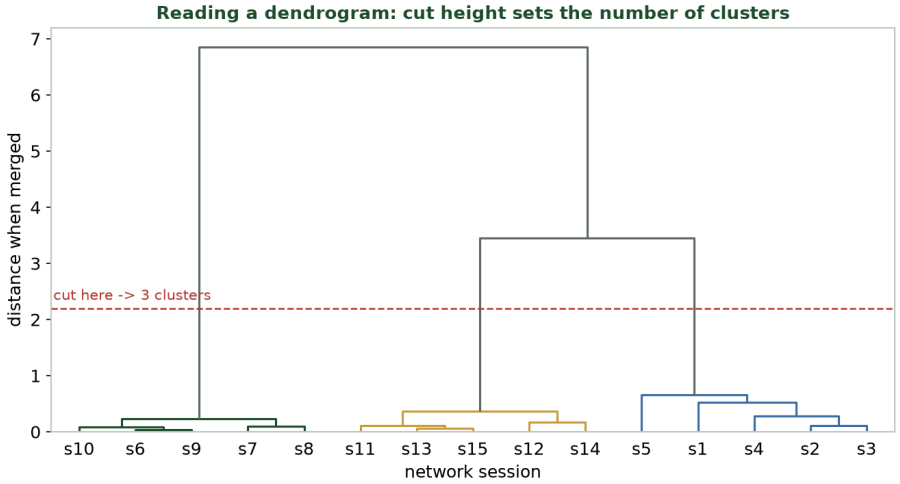

Figure 3. A dendrogram. A horizontal cut sets the number of clusters; cut just below a big vertical jump to keep meaningful groups apart.

- k-means: needs k up front, fast, scales to big data.
- Hierarchical: no k up front, shows nested structure, but heavier to compute (best for smaller data).
- Cross-check: do the elbow and the dendrogram suggest the same number of groups?

5. Anomaly Detection: Finding What Doesn't Fit

An anomaly (outlier) is a rare record unlike the normal majority - not a separate cluster, just a point that stands apart. Anomaly detection LEARNS normal, then flags deviations, with NO labels. The go-to method is the ISOLATION FOREST: it makes many random splits; anomalies are ISOLATED in only a few splits (they sit apart), while normal points take many. Every point gets a score, and you set 'contamination' - the expected fraction of anomalies.

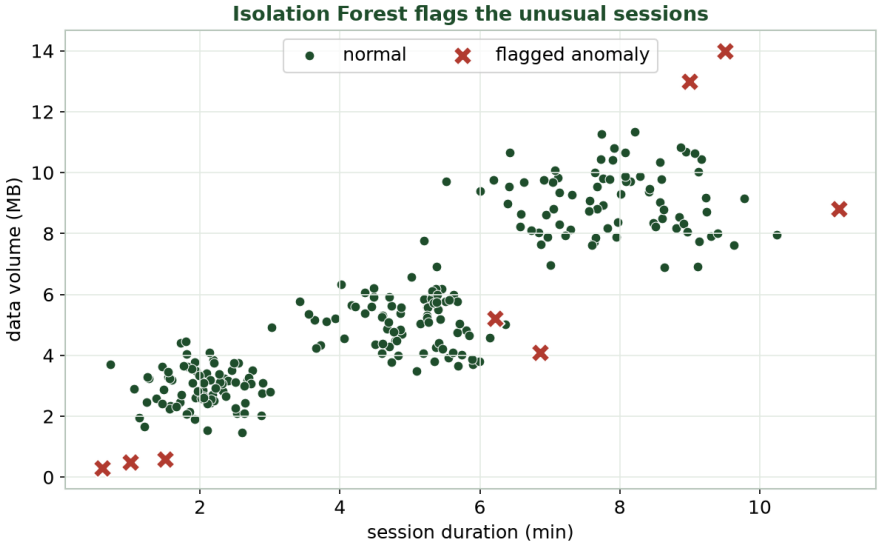

Figure 4. An Isolation Forest flags the unusual sessions (port scans, exfiltration). Some flagged points look normal in 2-D but are odd across all features.

A second method, LOCAL OUTLIER FACTOR (LOF), compares a point's crowding to its neighbors' - it catches LOCAL outliers in varying-density data. Isolation Forest is more global; LOF is more local.

Running both and comparing raises confidence. Remember: an anomaly means 'INVESTIGATE,' not 'attack' - keep a human in the loop and tune the alert volume to avoid alert fatigue.

6. The Workflow in Code

The whole week is a handful of scikit-learn lines - exactly what you run in Lab 5. Scale first, then cluster, then detect anomalies:

```
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.ensemble import IsolationForest

X  = df[['duration','data_volume','packets','failed_conns']]
Xs = StandardScaler().fit_transform(X)          # 1) SCALE first (always)

df['cluster'] = KMeans(n_clusters=3, n_init=10).fit_predict(Xs)   # 2) CLUSTER
df['anomaly'] = IsolationForest(contamination=0.03).fit_predict(Xs)  # 3) FLAG (-1 = anomaly)
```

Key Terms

- Unsupervised learning - finding patterns in data with no labels.
- Clustering - grouping similar records; cluster center = typical member.
- k-means - clustering that needs k chosen up front (use the elbow method).
- Inertia / elbow method - within-cluster spread; the bend suggests a good k.
- Scaling (standardization) - putting features on a comparable range before clustering.
- Hierarchical clustering / dendrogram - a merge tree you cut to choose clusters.
- Anomaly / outlier - a rare record that fits no normal pattern.
- Isolation Forest - flags points isolated in few random splits; scores every point.
- Local Outlier Factor (LOF) - flags points sparse relative to their neighbors.
- Contamination - the expected fraction of anomalies.

Review Questions (self-check for Quiz 5 and Lab 5)

- What makes a learning task 'unsupervised'?
- What does the elbow method help you choose, and how do you read it?
- Why must you scale features before k-means? What goes wrong if you don't?
- How is hierarchical clustering different from k-means?
- In plain words, how does an Isolation Forest decide something is an anomaly?
- Give a security use of anomaly detection and explain why it needs no labeled attacks.

Remember: AI tools are not permitted on quizzes and labs in this course.

References

- NIST, Artificial Intelligence Risk Management Framework (AI RMF 1.0), 2023.
- CISA, Detecting and Responding to Anomalous Activity (guidance).

- scikit-learn User Guide: Clustering, and Novelty & Outlier Detection.
- F. T. Liu, K. M. Ting, Z.-H. Zhou, 'Isolation Forest,' IEEE ICDM, 2008.
- M. M. Breunig et al., 'LOF: Identifying Density-Based Local Outliers,' ACM SIGMOD, 2000.
- Google Machine Learning Crash Course - Clustering (developers.google.com/machine-learning).