**1. Project Overview**

For my capstone project, I delved into network security by focusing on intrusion and anomaly detection. I utilized three datasets BETH, Cybersecurity Attacks, and UNSW NB15 to examine how various clustering and machine learning techniques can identify and analyze malicious patterns or attacks.

These datasets differ in their complexity and scope while encompassing cybersecurity scenarios. My objective was to reveal patterns within high dimensional noisy data and extract insights that could support detection systems in spotting behavior, within extensive network logs.

Every dataset presents views on cyber threat detection:

BETH Dataset; Comprises system event logs featuring columns like processId userId argsNum and returnValue. These numeric attributes monitor system behavior at the kernel level offering a level of detail for identifying malicious processes. Cybersecurity Attacks Dataset; Captures behavior with features like Packet Length Anomaly Scores and Severity Level.The Attack Type column acts as a label for supervised modeling. UNSW NB15 Dataset; Provides flow data comprising dur spkts dpkts sbytes rate sload and dload that reflect real world network patterns.

The main objective was to assess how machine learning methods both supervised and unsupervised can reveal structures within dimensional noisy data and whether recognizing patterns can aid, in detecting anomalies.

**2. Machine Learning Methods Employed**

Week 8; K Nearest Neighbors (KNN)

In my analysis of the BETH Cybersecurity and UNSW NB15 datasets I utilized KNN with both Euclidean and Manhattan distance metrics. The aim was to classify behaviors based on their proximity to labeled training examples. KNN proved effective for the dataset, with its smaller numeric feature set.

BETH Dataset;

* Accuracy: 0.87 using k=5, Euclidean distance
* Key features: returnValue, argsNum, sus
* The binary evil label was well separated in the feature space.
* Euclidean and Manhattan metrics showed minimal variance in accuracy.

Cybersecurity Dataset;

* F1 Score: 0.62 (macro average) due to class imbalance.
* Overlap in Anomaly Scores and Packet Length between attack types resulted in lower precision.

UNSW NB15;

* Accuracy: 0.69 with k=7
* Sparse regions in sbytes, sload, and rate reduced the efficacy of neighbor-based classification.

Insight; KNN demonstrated effectiveness, with low dimensional balanced data while being less reliable, on noisy or imbalanced datasets like Cybersecurity Attacks.

Week 9; Gradient Boosting

Gradient Boosting was employed to capture interactions among features. I utilized the Classifier from sklearn for all three datasets and examined feature importance plots.

In the BETH Dataset;

* AUC: 0.91 | Accuracy: 0.93
* Top Features: returnValue (importance = 0.28), argsNum (0.22), sus (0.15)

returnValue, argsNum and sus emerged as the features for predicting the label. Boosting outperformed KNN, in terms of precision especially.

In the Cybersecurity Dataset;

* Macro F1: 0.71 | AUC: 0.79
* Features: Anomaly Scores (0.34), Severity Level (0.25)
* Boosting outperformed KNN significantly by modeling nonlinear class boundaries.

Anomaly Scores and Severity Level were highlighted. Boosting revealed interactions overlooked by KNN.

For UNSW NB15;

* AUC: 0.86
* Boosting revealed synergies between sload, rate, dur, and dpkts
* Implemented early stopping on a validation fold (patience = 10)

Gradient Boosting excelled by integrating behaviors such as packet bursting (spkts, dpkts) and load dynamics (sload, dload).

Hyperparameter Tuning:

* Used GridSearchCV to explore combinations of learning\_rate = [0.05, 0.1], max\_depth = [3, 5], n\_estimators = [100, 200]

Insight: I fine tuned parameters using GridSearchCV (estimators, learning rate, depth) and found a balance. Between complexity and overfitting. Techniques like early stopping and shallow trees helped avoid fitting.

Week 10; K Means Clustering + Silhouette Scores

K means was utilized to explore structures without labels. The goal was to categorize behaviors into clusters and assess their quality, using silhouette scores.

BETH Dataset;

* Best k: 3 | Silhouette Score: 0.24
* Cluster 2 was dominated by high returnValue (>2000) and sus > 1.
* The optimal number of clusters ( K) was found to be 3. PCA plots indicated clusters, for commands. Cluster 2 showed a correlation with high returnValue and sus values.

Cybersecurity Dataset;

* Best k: 3 | Silhouette Score: 0.21
* Clusters were influenced by Anomaly Scores > 90 and moderate Packet Length
* PCA revealed overlapping centroids indicating poor separation.

K means displayed clusters with silhouette scores. The presence of Anomaly Scores and packet length influenced the formation of Cluster 0.

UNSW NB15;

* Best k: 3 | Silhouette Score: 0.27
* Cluster 0: High sload (>10^5), Cluster 1: Balanced load, Cluster 2: Short bursts with low dur

Both the elbow and silhouette methods suggested K=3. The clusters reflected traffic intensity with packet counts, loading patterns and durations aligning together.

Week 11; DBSCAN + Hierarchical Agglomerative

DBSCAN:

BETH ;

* eps=1.5, min\_samples=5: Found 2 core clusters and 41 noise points.
* Cluster 1 associated with returnValue > 3000

DBSCAN pinpointed a core cluster related to events, by analyzing features like returnValue and sus. PCA revealed cluster shapes.

Cybersecurity Attack;

* eps=1.2: Many samples labeled as noise (-1), weak cluster formation due to flat density
* Only high Anomaly Scores > 95 formed stable regions

DBSCAN faced challenges due to the lack of density contrast in Packet Length Only a fraction of attacks could be separated based on Anomaly Scores exceeding 80.

UNSW NB15;

* eps=1.5: 3 clusters, 89 noise points
* Dense traffic flows (high dload, sbytes) grouped together

It excelled in capturing packet activities with features like rate sload and dload forming concentrated areas.

HAC:

BETH

* n\_clusters=3, Ward linkage
* Dendrogram showed nested substructures in argsNum and threadId

Ward linkage identified nested groupings in behaviors. The dendrograms displayed three levels of nesting.

Cybersecurity Attack;

* Average linkage formed 3 loose clusters shaped by Severity Level

average linkage unveiled clusters. Severity Level helped differentiate attack types within branch patterns.

UNSW NB15

* Dendrogram separated high-throughput sessions from low-rate noise

HAC provided a separation between large and small flows. This was evident in the dpkts and duration metrics with clusters loosely corresponding to packet volumes.

**3. Addressing Overfitting and Evaluation Methods**

To mitigate overfitting; I employed the following strategies

* KNN and boosting techniques utilized cross validation
* Clustering assessments relied on silhouette evaluations
* Hyperparameter tuning for model adjustments was conducted through GridSearchCV for Gradient Boost and KNN (n\_neighbors, max\_depth, learning\_rate).
* Numeric ranges were normalized using the StandardScaler.

Result: Models were stable across folds. Boosting especially showed low variance between training and validation.

**4. Assessment Metrics and Parameter Optimization**

KNN; I focused on metrics like accuracy, F1 score and confusion matrix. Accuracy (BETH = 0.87), F1 (Cyber = 0.62), Grid: k=3–10.

Gradient boosting; AUC > 0.85 for all datasets; tuned via learning rate and depth

In K Means clustering; I calculated WCSS along with Silhouette Scores across different K values (optimal k=3 across datasets).

DBSCAN analysis involved visual evaluations and counting clusters. Tuned eps from 0.8–2.0; best at 1.5 (UNSW/BETH).

HAC scrutinized dendrogram heights and cluster coherence. linkage='ward' best for BETH/UNSW.

**5. Anticipated vs. Unforeseen Outcomes**

Anticipated Results; Boosting surpassing KNN performance and DBSCAN isolating noise and irregular events.

Unexpected Outcomes; KNN was more effective than expected on the BETH dataset model despite its simplicity. Clustering techniques showing effectiveness, on BETH and UNSW compared to Cybersecurity due to clearer separations.

**6. Impact of Exploratory Data Analysis (EDA)**

Distributions: sload, rate, Anomaly Scores showed extreme right skew

Correlations: sbytes and dload had strong Pearson correlation (>0.85)

Outliers: DBSCAN tuning guided by Anomaly Scores > 95 and returnValue > 2000

PCA: Used to reduce feature space and visualize K-Means/DBSCAN clusters

**8. Final Insights**

The BETH dataset proves suitable for both clustering and supervised classification tasks. Its features, (returnValue, argsNum, sus) effectively distinguish between malicious and benign entities. DBSCAN revealed dense anomaly groups, boosting produced precision = 0.9.

Conversely the Cybersecurity dataset faces challenges due to overlapping feature distributions. Models struggled to establish boundaries. DBSCAN labeled over 30% of samples as noise

Boosting moderately successful (F1 = 0.71), KNN underperformed.

UNSW NB15 showcases behavior patterns across traffic statistics. Techniques like DBSCAN and Boosting yielded the separation of high risk flows. Clustering succeeded; DBSCAN formed 3 high-intensity groups (Silhouette = 0.27) Boosting effective (AUC = 0.86), leveraging dload, sload, rate.

In conclusion; Gradient Boosting and DBSCAN (used in Weeks 9 and 11) provided the most valuable and comprehensible insights. I intend to enhance DBSCAN further and experiment with Isolation Forests for detecting anomalies, and also look into deep clustering methods as my next course of action.

APA CITATION

Analytics Vidhya. (2021, June 22). DBSCAN clustering simplified: A machine learning approach.

Krishna, N. (2021, February 10). Outlier detection with Isolation Forest. Towards Data Science.

Starmer, J. (2017, August). Hierarchical clustering explained. StatQuest with Josh Starmer [YouTube Channel].

Rajendran, P., & Dhanasekaran, R. (2022). Application of DBSCAN for traffic incident detection using video analytics. International Journal of Intelligent Transportation Systems Research, 20, 135–147.

Zhou, J., Pan, J., Wang, W., & Vasilakos, A. V. (2020). Machine learning on big data: Opportunities and challenges. Neurocomputing, 414, 370–381.