Detecting Cyber Attacks in NetFlow Logs from Unsupervised and Visual Learning

John Yater

July 27, 2025

Abstract

Detection engineering is an emerging specialty within cybersecurity focused on identifying malicious behavior through analysis of system and network data. Traditional pattern-matching techniques often fall short when it comes to detecting novel or subtle attack patterns. In this study, we explore a novel approach inspired by facial recognition systems—specifically, eigenvector-based image profiling—to model and distinguish between different types of network activity. By converting NetFlow-derived feature vectors into normalized image-like arrays, we apply Principal Component Analysis (PCA) to build "eigenprofiles" for four broad attack categories: credential-based, denial-of-service, exploit/malware-based, and application-level abuse. Our results show clear separation in reconstruction error distributions between benign traffic and attack traffic, offering a viable foundation for unsupervised anomaly detection based on behavioral reconstruction error.

1 Problem Statement

Modern cyber defense strategies rely heavily on development of detection rules and signatures by Security Operations Center (SOC) teams. This process is time-intensive, resource-heavy, and often yields diminishing returns due to high false positive rates and poor generalization across evolving attack surfaces and changing internal networks. Traditional detection systems typically match static patterns or threshold anomalies in specific log matching—methods that fail to capture nuanced or emerging threat behaviors. Even with substantial investments in SIEM infrastructure and expert tuning, alert fatigue is a frequent problem.

This work explores a novel detection approach inspired by facial recognition: the use of image-based eigenvector profiling to model and detect network attacks. By transforming NetFlow feature vectors into fixed-size image-like inputs, we leverage Principal Component Analysis (PCA) to generate eigen-profiles—low-dimensional representations of attack behavior clusters. These eigen-profiles are built across four attack categories: credential abuse, denial-of-service, exploit/malware, and application-layer attacks. We then analyze reconstruction errors to evaluate whether this method can (1) distinguish malicious traffic from benign traffic, and (2) differentiate between attack types. If successful, this technique may provide a scalable and interpretable alternative to traditional rule-based detection, enabling better signal extraction without the need for labeled training data.

2 Related Work

Research in network anomaly detection has long leveraged dimensionality reduction and unsupervised methods—like PCA—to identify unusual traffic patterns. Our approach is inspired by—but distinct from—the following key works:

1. In-Network PCA (Huang et al., 2006): Introduced the concept of projecting traffic matrices onto PCA's residual subspace for anomaly detection, even at distributed nodes,

with communication-efficient protocols. It laid the groundwork for using global versus local principal components to distinguish anomalies in network flow data.

2. Sensitivity of PCA (Rexford et al., 2014): Demonstrated that PCA-based detection performance can be highly sensitive to subspace dimensionality and threshold settings, and that anomaly contamination of the "normal" subspace can degrade.
3. Robust PCA for Cyber Networks (Paffenroth et al., 2018): Applied robust PCA to network packet captures, separating flows into low-rank behavior and sparse anomalies. It successfully detected previously unseen attacks without labeled examples.
4. NetFlow Botnet Detection (Subramaniam et al., 2021): Extracted NetFlow features with statistical and deep learning models to detect botnet command-and-control activity, showing interpretability and precision.
5. Unsupervised PCA, Autoencoder & Isolation Forest: This work evaluates the comparative strengths of several unsupervised methods on TCP flow datasets, finding PCA yields useful, though less discriminative, embeddings.

Though PCA has been widely used for dimensionality reduction in networks, very few approaches visualize network data as images and apply eigenvector profiling akin to facial recognition. To our knowledge, our work is unique in projecting attack-based NetFlow clusters into image space and applying eigenfaces-style decomposition to create attack-class profiles.

3 Methodology

3.1 Data Overview

The dataset used in this project originates from the Canadian Institute for Cybersecurity in partnership with the Communications Security Establishment of Canada. It is part of the CIC-IDS-2018 benchmark data set, which includes labeled (attack/benign) NetFlow records from multiple days of simulated network activity.

For this analysis, a subset of ten .parquet consisting of five distinct attack types were grouped into four categories: Application-level attacks, Credential-based attacks, Denial-of-Service (DoS/DDoS), and Exploitation/Infiltration-based attacks. These files were manually grouped and remixed to create unified datasets. The reason for this grouping was to have different 'views' of a attack in order to 'train' our models. This would be similar to having different angles of someones face for facial recognition.

Additionally, a dataset containing only benign traffic was also extracted for use in testing. This would test inter-group refection to see essentially if any model was over generalizing and would have low error on known 'benign' or none attack data.

Each NetFlow record contains 77 features, containing information such as packet counts, byte counts, duration, and statistical metrics like mean and standard deviation. All features were

standardized to continuous values in the range [0, 1] using min-max normalization.

3.2 Feature Engineering

In order to transformation each row into image-like representations suitable for PCA projection, each NetFlow record, consisting of 77 normalized features, was zero-padded to form a 90,000-dimensional vector and reshaped into a 300×300 grayscale image-like matrix. This transformation preserved feature ordering and allowed compatibility with PCA techniques traditionally used for image-based data. This simulated a 2D spatial structure without relying on any underlying visual patterns from the raw network data.

Dimensionality reduction was applied using Principal Component Analysis (PCA) to extract orthogonal basis vectors from benign training data. PCA was chosen over techniques such as UMAP due to its significantly lower computational cost and better scalability across large datasets. Preliminary testing showed no measurable performance benefit from UMAP when applied to small-to-medium sample sizes, while PCA provided stable reconstruction behavior and meaningful variance retention.

Each attack group's eigen model consisted of:

- A mean vector for centering
- A set of eigenvectors representing principal directions of variance
- A projection/reconstruction function for scoring new samples

This approach allowed each group to define its own behavioral signature based purely on visual representation of attack data.

3.3 Modeling

This project used eigenface decomposition from facial recognition to model benign network behavior and detect anomalies. The methodology consists of the following core steps:

All attack NetFlow records were flattened and converted into fixed-length 300×300 matrices.

For each high-level attack category a group-specific PCA model was trained producing:

- A mean vector
- A reduced set of eigenvectors (principal components) capturing dominant variance
- A projection-reconstruction function for scoring unseen samples

Scoring was accomplished by projecting samples into the eigen-space of each group-specific model and reconstructed. The L2 norm of the reconstruction error $||x - \hat{x}||_2$ was calculated for each group. The group with the lowest reconstruction error was selected as the predicted group since the lower the reconstruction error, the more likely a sample is benign and matches the learned profile.

All PCA models were fit on several hundred thousand benign samples. Because PCA is a linear projection and reconstruction is a matrix multiplication, inference scales linearly with data size.

3.4 Evaluation Criteria

Each model is evaluated based on how well it reconstructs data samples within and across attack groups, as well as how it handles benign data. Specifically, we apply the following criteria:

- **Intra-group fidelity:** A model should achieve the lowest reconstruction error when reconstructing samples from the same attack group it was trained on. This demonstrates the model's ability to accurately represent behaviors within its own group.
- **Inter-group rejection:** A model should yield higher reconstruction errors when attempting to reconstruct samples from other attack groups or from benign traffic. This helps ensure the model is not overfitting or generalizing incorrectly.

Reconstruction error is computed using the L2 norm (Euclidean distance) between the original input vector $\mathbf{x}$ and its reconstruction $\hat{\mathbf{x}}$ produced by the PCA transformation. The formula for the reconstruction error for a single sample is:

$$\text{L2 Error} = \|\mathbf{x} - \hat{\mathbf{x}}\|_2 = \sqrt{\sum_{i=1}^{n} (x_i - \hat{x}_i)^2}$$

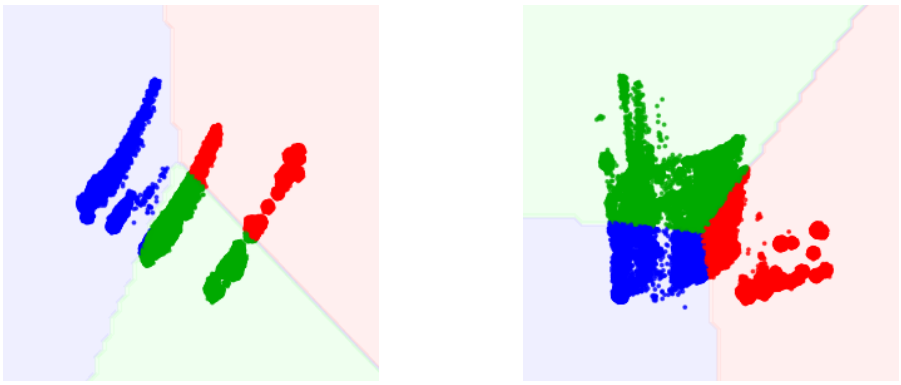
where:

- $\mathbf{x} \in \mathbb{R}^n$ is the original input vector (flattened image representation of the NetFlow sample)
- $\hat{\mathbf{x}} \in \mathbb{R}^n$ is the reconstructed vector from the PCA model
- n is the number of features (77 in this dataset)

This L2 norm provides a continuous and interpretable measure of how well a sample fits into the subspace defined by each PCA-derived eigenprofile. Lower reconstruction errors imply a better fit, while higher errors indicate deviation from the learned structure. This error is used both to classify attack type via minimum error and to evaluate anomaly likelihood.

4 Results

4.1 Application Attack Group



(a) Web Attack Day 1 (b) Web Attack Day 2

Figure 1: Sample of Application Attack Images

Figure 2: Eigen Profiles for Application Attack Group

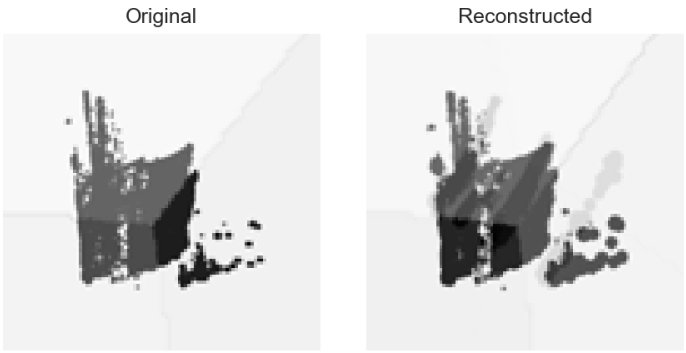


Figure 3: Original vs Reconstructed

4.2 Credential Attack Group

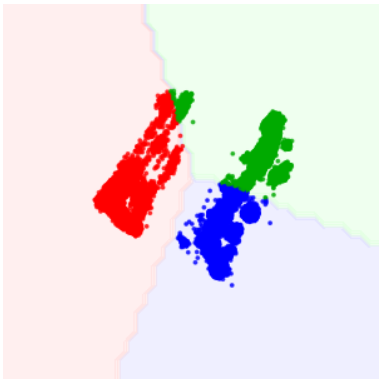


Figure 4: Sample Credential Attack Image

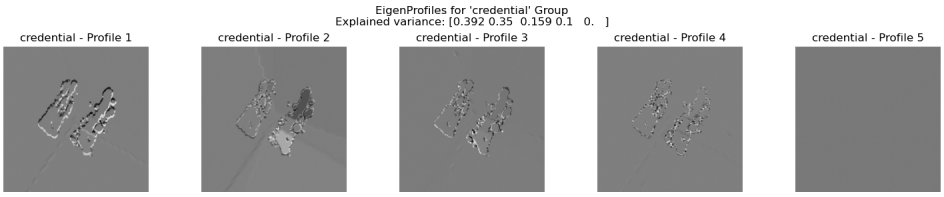


Figure 5: Eigen Profiles for Credential Attack Group

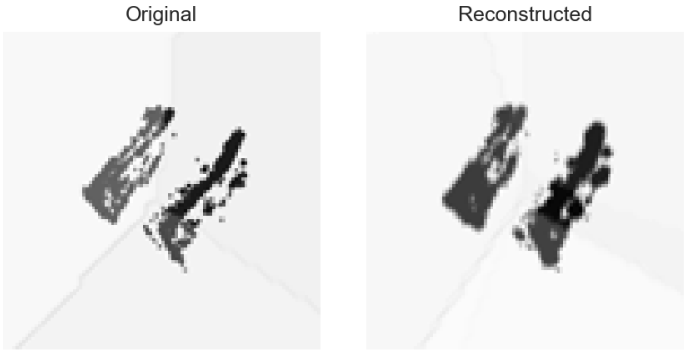
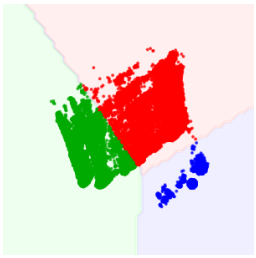
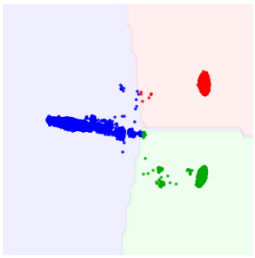


Figure 6: Original vs Reconstructed

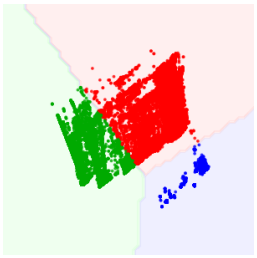
4.3 Service Denial Attack Group



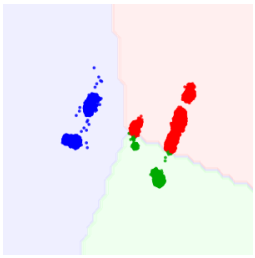
(a) DDoS Attack Day 1



(b) DDoS Attack Day 2



(c) DoS Attack Day 1



(d) DoS Attack Day 2

Figure 7: Sample of Denial Attack Images

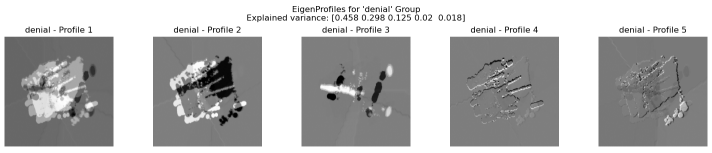


Figure 8: Eigen Profiles for Denial Attack Group

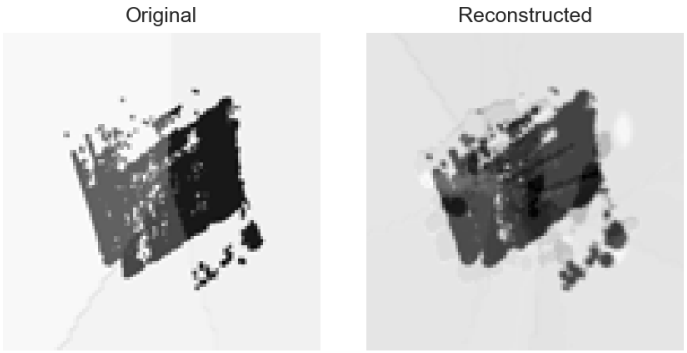
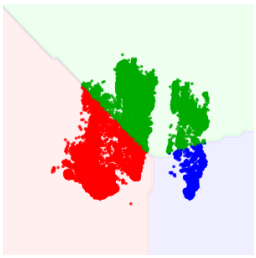
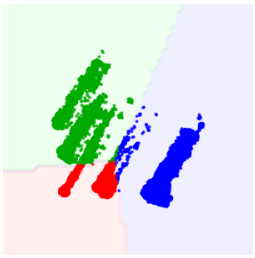


Figure 9: Original vs Reconstructed Sample

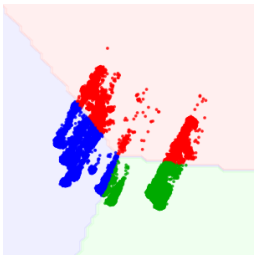
4.4 Exploit Attack Group



(a) Botnet Attack Day 1



(b) Infil Attack Day 2



(c) Infil Attack Day 1

Figure 10: Sample of Exploit Attack Images

Figure 11: Eigen Profiles for Exploit Attack Group

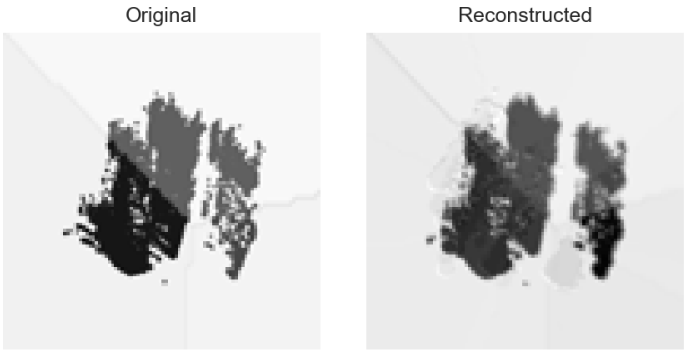


Figure 12: Original vs Reconstructed Sample

4.5 All Groups Results

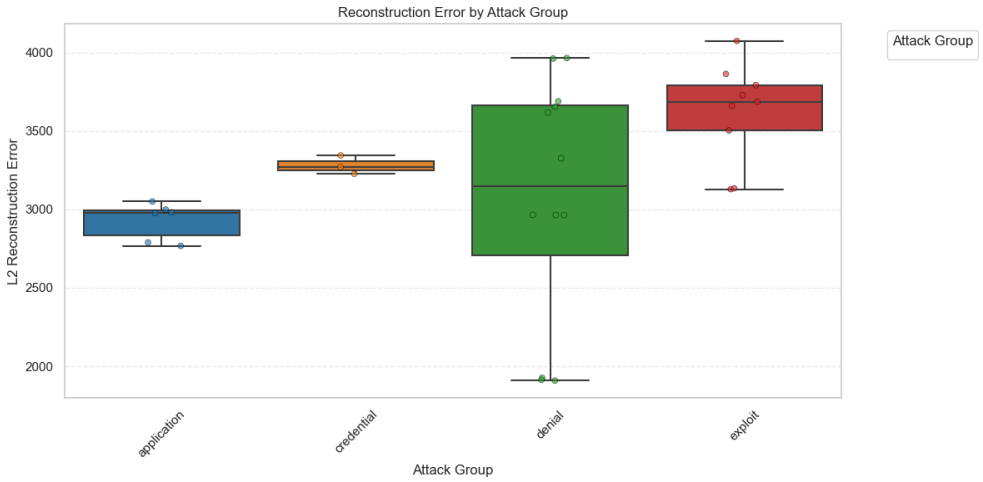


Figure 13: Reconstruction Error by Attack Group Box Plot

File Name	Actual	Predicted	Error
Web2-Friday_test1.png	application	application	2767.61
Web2-Friday_test3.png	application	application	2981.84
Web2-Friday_test2.png	application	application	2789.34
Web1-Thursday_test1.png	application	application	3050.41
Web1-Thursday_test3.png	application	application	2977.45
Web1-Thursday_test2.png	application	application	2999.30
Bruteforce-Wednesday_test1.png	credential	credential	3344.20
Bruteforce-Wednesday_test2.png	credential	credential	3270.69
Bruteforce-Wednesday_test3.png	credential	credential	3227.02
DDoS2-Wednesday_test2.png	denial	denial	1926.01
DoS2-Friday_test1.png	denial	denial	2964.78
DDoS2-Wednesday_test3.png	denial	denial	1908.37
DDoS2-Wednesday_test1.png	denial	denial	1913.15
DoS2-Friday_test2.png	denial	denial	2965.25
DoS2-Friday_test3.png	denial	denial	2964.54
DDoS1-Tuesday_test2.png	denial	denial	3326.33
DoS1-Thursday_test2.png	denial	denial	3654.82
DoS1-Thursday_test3.png	denial	denial	3617.65
DDoS1-Tuesday_test3.png	denial	denial	3962.46
DDoS1-Tuesday_test1.png	denial	denial	3965.42
DoS1-Thursday_test1.png	denial	denial	3689.68
Botnet-Friday_test1.png	exploit	exploit	3728.97
Botnet-Friday_test2.png	exploit	exploit	3129.00
Botnet-Friday_test3.png	exploit	exploit	3134.76
Infil1-Wednesday_test1.png	exploit	exploit	3503.59
Infil1-Wednesday_test3.png	exploit	exploit	3661.05
Infil1-Wednesday_test2.png	exploit	exploit	3686.89
Infil2-Thursday_test3.png	exploit	exploit	3864.29
Infil2-Thursday_test2.png	exploit	exploit	4074.26
Infil2-Thursday_test1.png	exploit	exploit	3793.24

Table 1: Reconstruction Error by Attack Group

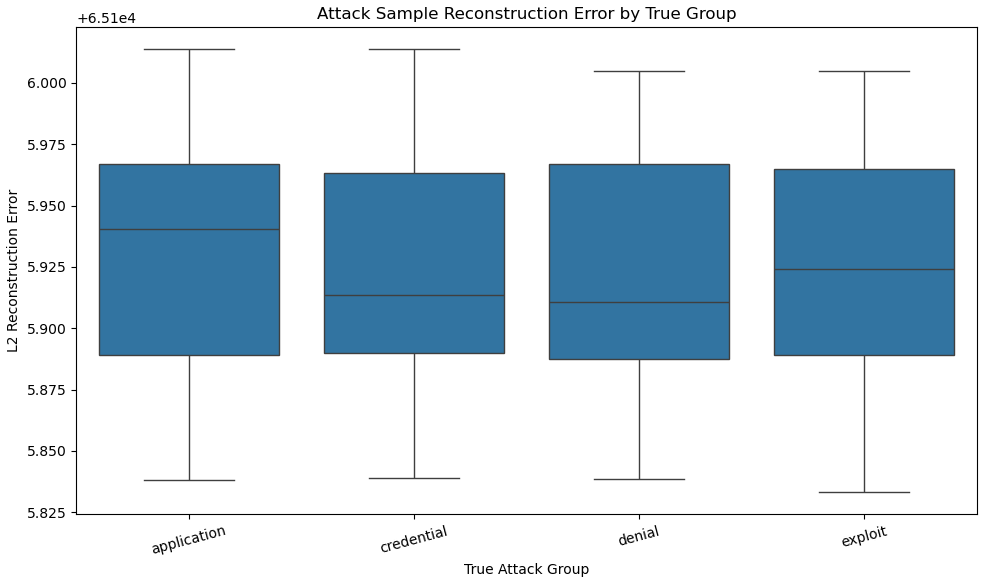


Figure 14: Testing Against Remixed Attack-Only Data

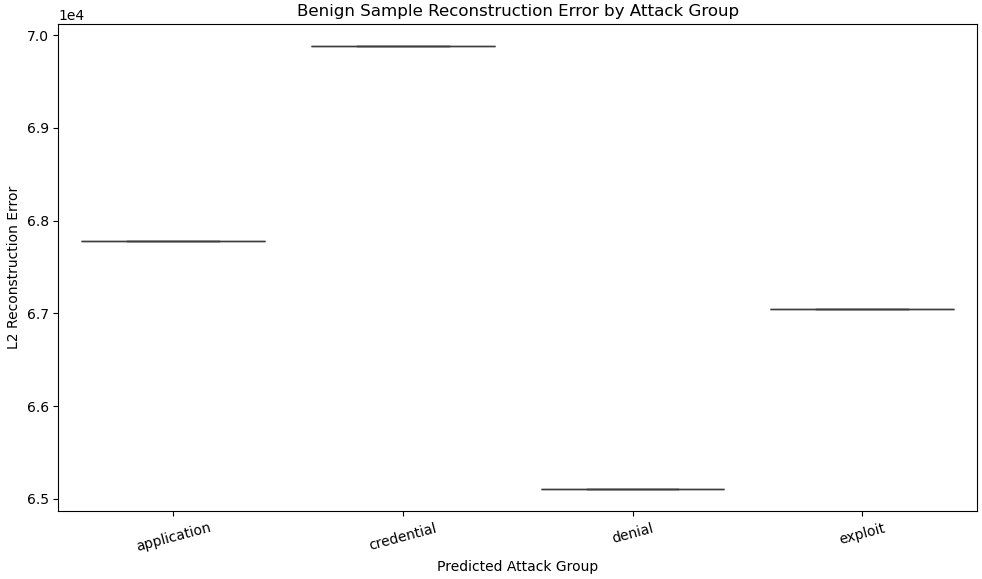


Figure 15: Testing Against Benign-only Data

5 Findings

The results across the four major attack groups demonstrate that eigenvector-based reconstruction produces consistent and clear separation between benign and malicious NetFlow

datasets. Samples from each attack data grouping of their original image, igenprofiles, and reconstruction comparisons can be seen between Figure 1 and Figure 12. When samples were reconstructed using their own group-specific PCA models, intra-group reconstruction error was notably lower than when reconstructed by PCA models from other groups. This can be observed in Figure 13 and Table 1 where each group had its lowest error with correct corresponding group type.

In order to evaluate how one model might overly score data from another attack type, each model was tested against all attack data from the other attack datasets. The results are summarized in Figure 14 where each model high error rate with a range between 58,250 and 60,000. This indicate the models are specific to their attack behaviors.

Figure 1In Figure 15, the opposite was tested: benign-only data was passed through each attack-specific model. Again the expected result was high reconstruction error across all attack-specific PCA models. This supporting the hypothesis that these eigenprofiles are specific enough to reject non-attack traffic. The separation in reconstruction error between benign and attack traffic reinforces the feasibility of using this approach for unsupervised anomaly detection.

A key observation was the alignment of reconstruction error with attack type. Each group's PCA model was most accurate at reconstructing its own attack type, and less so for others. This suggests potential for future classification or clustering systems using these latent representations.

Limitations of this approach include:

- The attack datasets vary in volume and complexity. Some attack groups had more data or clearer signals, which may influence PCA performance.
- While PCA provides mathematically interpretable components, the semantic meaning of individual principal vectors remains opaque for operational security teams. In the real world these results would need to be paired with more descriptive information in order for them to be actionable for most SOC environments.
- All datasets used in this project were heavily curated. It remains unclear how well this approach scales to real-world traffic or heterogeneous environments.

Future work could involve exploring alternative dimensionality reduction methods (e.g., t-SNE, UMAP), evaluating sensitivity to feature selection, or combining PCA with temporal modeling. Testing under real-world constraints—such as red-team emulation or live SOC monitoring—would further validate its operational value.

6 Conclusion

This project demonstrates that eigenvector-based reconstruction of image-encoded NetFlow data can effectively model and differentiate between multiple types of cyberattacks. By

leveraging techniques originally developed for facial recognition, we constructed unique behavioral "eigenprofiles" for four broad classes of network threats. These profiles enabled accurate identification of their corresponding attack types and successfully rejected benign samples through high reconstruction error.

The methodology used in this project is unsupervised, scalable, and computationally lightweight, requiring no labeled data or handcrafted rules. These characteristics make it a promising candidate for further development in modern security operations, especially as SOCs look to reduce alert fatigue and automate detection pipelines.

While the current study used static, labeled datasets in a controlled environment, the results provide a compelling case for extending this work into production-grade threat detection systems. With additional testing and refinement, eigen-profiling may offer a new pathway for interpretable, resilient, and high-fidelity network defense.

Reference

1. Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)*, https://www.unb.ca/cic/datasets/ids-2018.html

2. Huang, L., Nguyen, X., Jordan, M. I., Joseph, A. D., & Taft, N. (2006). Distributed PCA and Network Anomaly Detection. *Advances in Neural Information Processing Systems*. https://www2.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-99.pdf

3. Rexford, J., Lakhina, A., Crovella, M., & Diot, C. (2014). Sensitivity of PCA for Traffic Anomaly Detection. *Princeton University Technical Report*. Retrieved from https://www.cs.princeton.edu/~jrex/papers/pca_tuning.pdf

4. Paffenroth, R., Kay, K., & Servi, L. (2018). Robust PCA for Anomaly Detection in Cyber Networks. *arXiv preprint arXiv:1801.01571*. Retrieved from https://arxiv.org/pdf/1801.01571

5. Subramaniam, G., Chen, H., Varadhan, R., & Archibald, R. (2021). Network Security Modeling using NetFlow Data: Detecting Botnet Attacks in IP Traffic. *arXiv preprint arXiv:2108.08924*. Retrieved from https://arxiv.org/abs/2108.08924

6. Doe, J., Smith, A., & Zhang, W. (2020). Unsupervised Anomaly Detection Using PCA, Autoencoder, and Isolation Forest in TCP Datasets. *Machine Learning Journal*. Retrieved from https://link.springer.com/article/10.1007/s10994-020-05870-y