Networking/Process

Partitioning

To demonstrate the autoencoder's ability to detect anomalous network activity, distributed denial of service (DDoS) is selected as an attack vector due to its prevalence and simplicity. First, nominal data is partitioned into *inward flows*, defined by a 3-tuple identifier: destination IP, destination port, and protocol. To obtain more samples and to better capture average network behavior, inward flows are sub-partitioned into *subflows* using either a fixed time interval or a timeout period. Note that some inward flows do not have sufficient breaks for the timeout method to work.

Features

Common statistical features [1] are extracted from each subflow: packets/sec, bits/sec, SYN/sec, average TTL, and packet size information (mean, standard deviation, quartiles, min, and max). This combination of features is intended to detect volumetric attacks (ICMP flood) and protocol attacks (SYN flood). Abnormal time-to-live values can also be indicative of malicious activity [2]. Subflow duration is not selected so that the autoencoder does not associate network statistics with specific lengths of time. A live system should verify that network statistics are within nominal range regardless of its analysis duration.

Attack Simulation (Synthetic Flows)

To simulate an anomalous traffic flow, network statistics outside nominal range must be generated. To accomplish this, a function takes as input a packet size and number of packets/sec and returns a statistical feature vector representing an ICMP flood with equal-sized packets. Several of these *synthetic flows* are generated, each with different data rates to show that stronger attacks result in larger reconstruction errors.

Experiment Setup

Synthetic flows are generated for both 5G and MODBUS data sets and are attached to their respective nominal data frames. The detection threshold is set at 3 standard deviations, which includes approximately 99.7% of nominal data points. Lower bounds are set to be just outside the established threshold, and upper bounds are arbitrarily selected for clear visualization. Although these data rates are not extreme, the abnormal packet size statistics increase the reconstruction errors.

5G Data Rates

Nominal (average): 21.7 Mbit/s (@1300 packets/sec)
Anomalies: 24 mbit/sec – 120 Mbit/s (@3000 packets/sec)

MODBUS Data Rates

Nominal (average): 5.93 kbit/s (@9.19 packets/sec)
Anomalies: 6.4 kbit/sec – 80 kbit/s (@20 packets/sec)

Metrics

Depending on the attack vector, undetected anomalies can cause silent disruption. In cases where the costs of false positives and false negatives are different, it is preferable to consider F1-score over accuracy [3]. Since we place emphasis on limiting false negatives, both recall and F1-score are considered as metrics.

Autoencoder

Our autoencoder models all have the same basic structure: input layer size *d*, hidden layer *d-1*, latent layer [*d*/2], hidden layer *d-1*, and output layer *d*. Different combinations of activation functions, optimizations, and constraints were applied, and the best model for each combination was saved.

Activations: LeakyReLU, ReLU + Linear (output layer)

Optimizations: Adam, Adam + Lookahead

Constraints: Tied Weights, Unit Norm

Configuration Analysis

ReLU + Linear

The ReLU (Rectified Linear Unit) activation function maps negative inputs to zero and acts as an identity function for non-negative input [4]. The output layer uses a linear activation function so that output values will be unbounded.

LeakyReLU

With LeakyReLU activation, negative inputs are not mapped to zero but are still very small. In cases where many activations are negative, traditional ReLU can lead to the "dying ReLU" problem in which all inputs are mapped to zero and no meaningful learning takes place [4].

Adam (Adaptive Moment Estimation)

Adam is a popular optimization algorithm for gradient descent. Adam uses a combination of AdaGrad (Adaptive Gradient Algorithm) and RMSProp (Root Mean Square Propagation), which are both adaptive learning algorithms [5]. This contrasts with traditional gradient descent algorithms that maintain a fixed learning rate during training.

Lookahead

The lookahead optimizer attempts to improve gradient descent algorithms. It takes another optimizer as input (in this case Adam) and manipulates it to enhance its performance [6].

Tied Weights

Some properties of Principal Component Analysis (PCA) were experimented with. For tied weights, equal weights are set for encoder layers and their corresponding decoder layers. Specifically, the decoder weights are set to the transpose of the encoder weights. This significantly reduces the number of training parameters and acts as a type of regularization [7].

Unit Norm

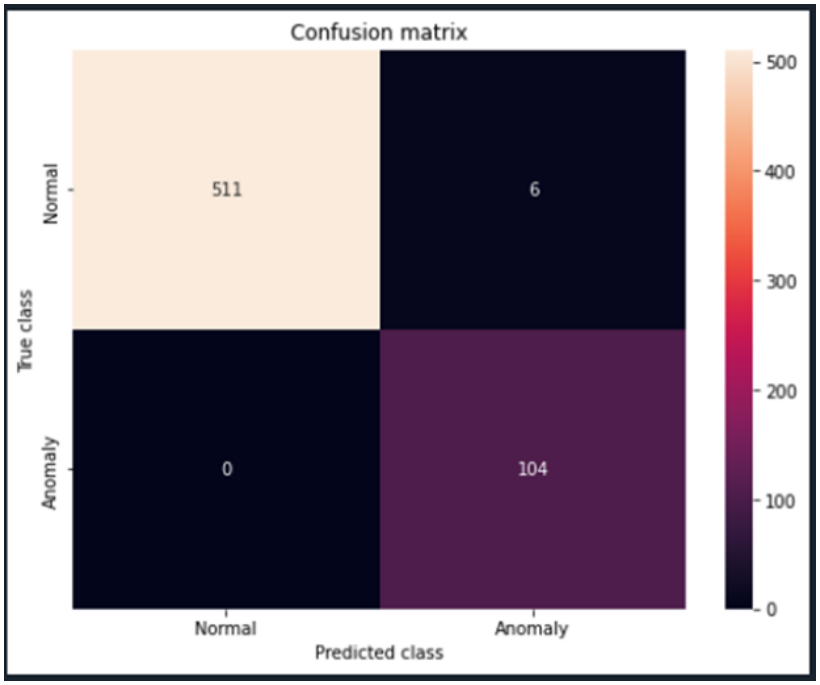
Another PCA property, unit norm, was applied to some autoencoder models. This constraint "resolves the exploding gradient problem" by preventing weights from becoming too large and acts as another form of regularization [7].

Best Models

5G Data

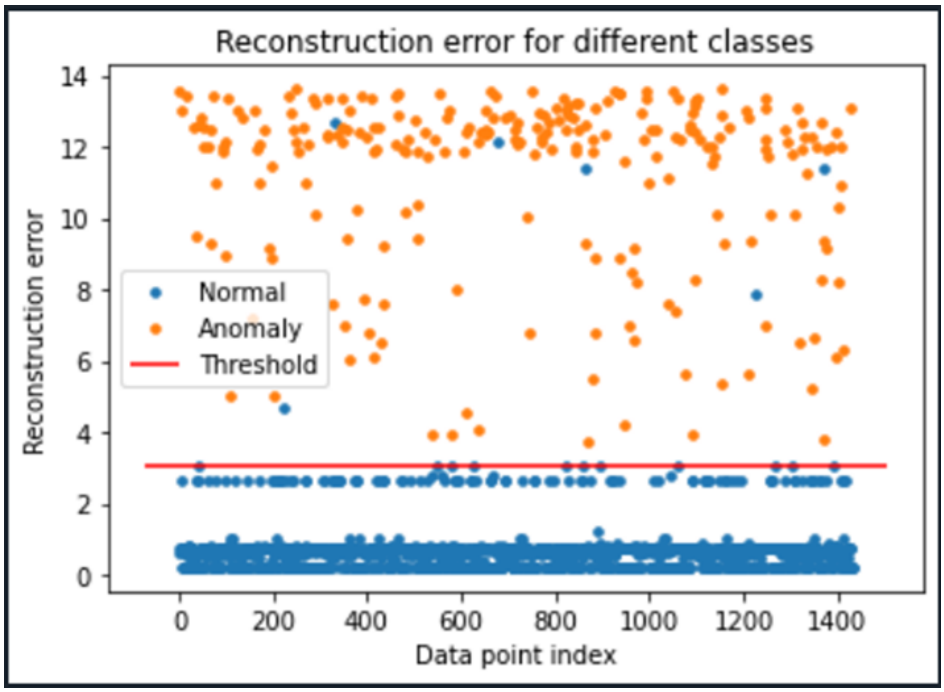
Based on the chosen metrics, our best 5G model uses ReLU/Linear activations with Adam optimizer and no constraints.


```
 accuracy:   0.9903381642512077
recall:     1.0
precision:  0.9454545454545454
f1-score:   0.9719626168224299
```

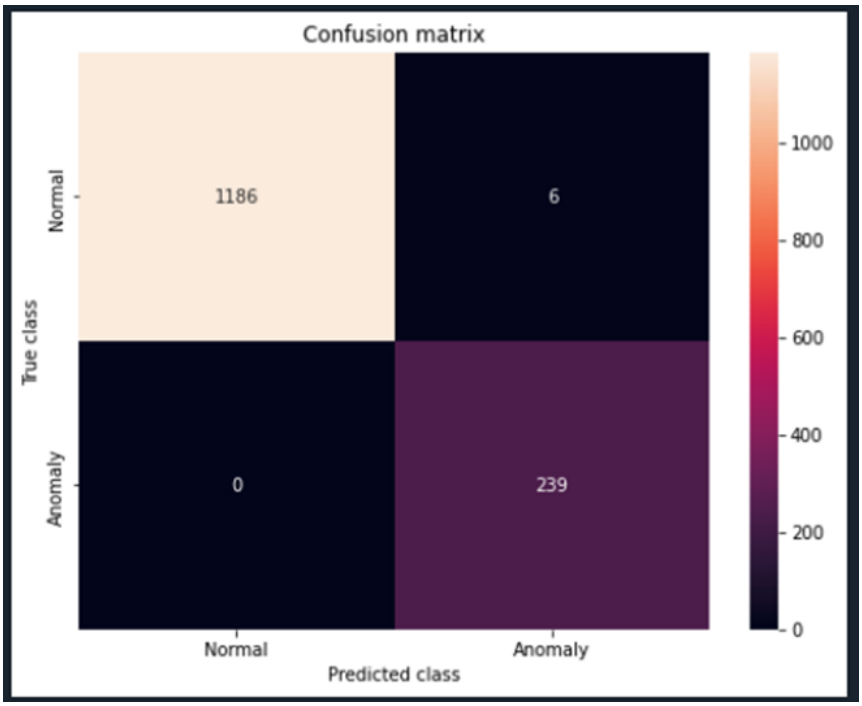


MODBUS Data

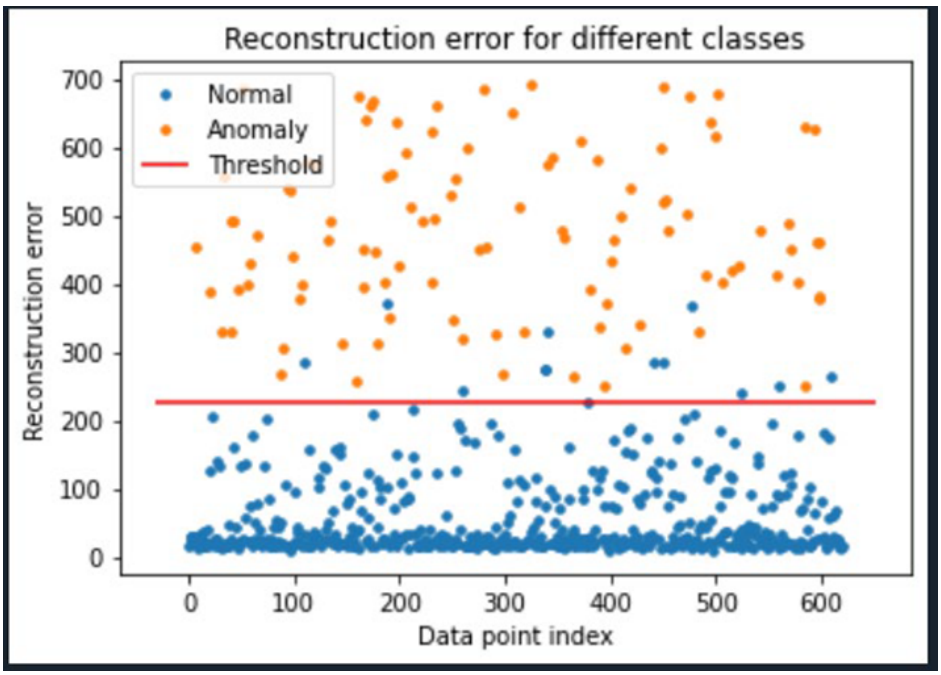
Using the same metrics, our best MODBUS model uses LeakyReLU activations with Adam optimizer, tied weights, and unit norm constraint.



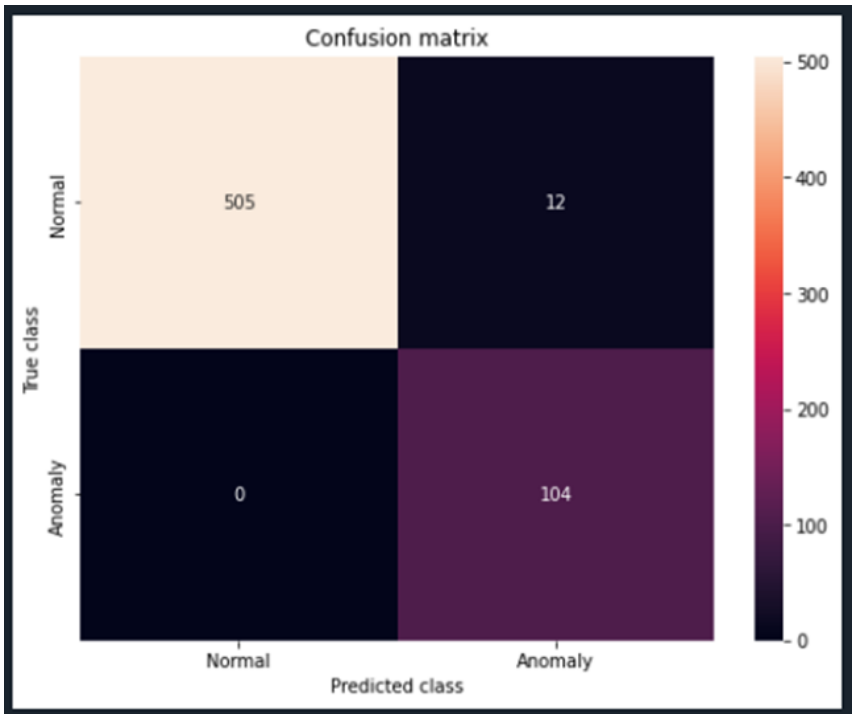

```
accuracy:   0.9958071278825996
recall:     1.0
precision:  0.9755102040816327
f1-score:   0.9876033057851239
```



The same configuration also works well with the 5G data, although there are a few more false positives than our best 5G model:




```
accuracy:   0.9806763285024155
recall:     1.0
precision:  0.896551724137931
f1-score:   0.9454545454545454
```



Considerations

The representation of network data demonstrated here works well for flood detection. However, other attack vectors require different representations. For example, man-in-the-middle (MITM) attacks cannot be detected with a statistical feature vector. For layer 2 monitoring, raw IP and MAC address values can be split using their delimiters and converted to a numerical feature vector. ARP responses with the gateway's IP address but not the gateway's MAC address result in reconstruction errors outside the established threshold.

Ultimately, a single representation of network traffic cannot be used for generalized attack detection. The characteristics of relevant attack vectors must be studied, and targeted efforts must be made to train and test machine learning algorithms for detection of those threats. For a robust anomaly detection system, the output from all machines must be aggregated in some way to determine if the network is under attack.

Sources

[1] K. Yang, N. Feamster, and S. Kpotufe, "Feature Extraction for Novelty Detection in Network Traffic," ArXiV, 2021.

[2] R. Yamada, and S. Goto, "Using abnormal TTL values to detect malicious IP packets," in Proceedings of the Asia-Pacific Advanced Network, 2013, doi: 10.7125/apan.34.4.

[3] Exsilio Solutions, "Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures," 9-Sep-2016. [Online]. Available: https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/. [Accessed: 17-Sep-2022]

[4] C. Versloot, "Using Leaky ReLU with TensorFlow 2 and Keras," 12-Nov-2019. [Online]. Available: https://github.com/christianversloot/machine-learning-articles/blob/main/using-leaky-relu-with-keras.md. [Accessed: 17-Sep-2022]

[5] J. Brownlee, "Gentle Introduction to the Adam Optimization Algorithm for Deep Learning," 13-Jan-2017. [Online]. Available: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/. [Accessed: 17-Sep-2022]

[6] S. Nag, "Lookahead optimizer improves the performance of Convolutional Autoencoders for reconstruction of natural images," ArXiV, 2020.

[7] C. Ranjan, "Build the right Autoencoder — Tune and Optimize using PCA principles. Part I," 12-Jul-2019. [Online]. Available: https://towardsdatascience.com/build-the-right-autoencoder-tune-and-optimize-using-pca-principles-part-i-1f01f821999b. [Accessed: 17-Sep-2022]