Supplementary Information

A BETH Dataset Features

Table S1: Description and type of each feature from the original BETH dataset (Highnam et al., 2021). In our implementation, an anomalous event is determined as by the sus (suspicious) label.

Feature	**Description**
timestamp	Seconds since system boot
processId	Integer label for the process spawning this log
threadId	Integer label for the thread spawning this log
parentProcessId	Parent's integer label for the process spawning this log
userId	Login integer ID of user spawning this log
mountNamespace	Set mounting restrictions this process log works within
processName	String command executed
hostName	Name of host server
eventId	ID for the event generating this log
eventName	Name of the event generating this log
argsNum	Length of arguments
returnValue	Value returned from this event log (usually 0)
stackAddresses	Memory values relevant to the process
args	List of arguments passed to this process
sus	Binary label as a suspicious event (1 = suspicious, 0 = not)
evil	Binary label as a known malicious event (0 = benign, 1 = malicious)

B Classical Autoencoder Architecture

We implemented our CAE using *TensorFlow* and *Keras* in Python. The architecture was determined through extensive experimentation with different hyperparameters.

The encoder compresses the input data from 24 features down to an 8-dimensional latent representation, while the decoder mirrors this structure to reconstruct the original input. The First Dense layer is followed by Batch Normalization and for every Dense layer, a LeakyReLU activation, with a Dropout layer applied after the first activation to reduce overfitting.

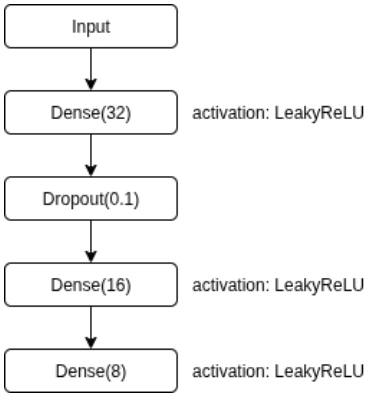

Fig. S1 Encoder Architecture for 24 input feature model

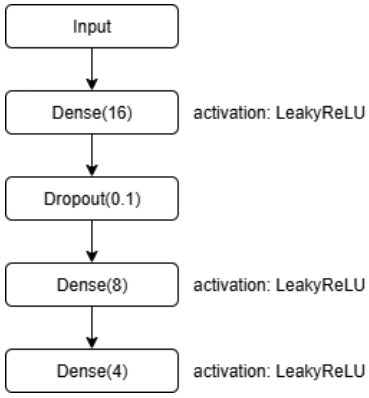

Fig. S2 Encoder Architecture for 8 input feature model

The model was compiled using the *Adam optimizer* and the *Mean Absolute Error (MAE)* as the loss function. To facilitate convergence and improve performance, we employed `EarlyStopping` and `ReduceLROnPlateau` callbacks.

Training was performed for 20 epochs with a batch size of 64, shuffling the data at each epoch. Instead of splitting the training data for validation, we utilized the validation set provided by the BETH dataset.

C Classical Autoencoder Sample Size Effects

In the classical setting, we trained autoencoders on varying dataset sizes and feature inputs. Two configurations were considered: one using the full 24 input features, and another restricted to 8 input features for a fairer comparison with the QAE.

Table S2 summarizes the results for the 24 input feature model across different training set sizes. As shown, the model trained on the full dataset of 761,875 samples achieved the best performance, surpassing the results reported by Highnam et al.. We attribute this improvement largely to the additional feature engineering applied during preprocessing.

Table S2: Performance of the 24-feature CAE across different training set sizes. Reported metrics are F1-score and AUROC.

Features	**Samples**	**F1**	**AUROC**
24	1,000	0.81	0.969
24	5,000	0.78	0.961
24	10,000	0.81	0.892
24	20,000	0.86	0.898
24	761,875	0.87	0.963

D Classical Autoencoder Random Sampling Effects

To ensure comparability with the QAE, we also evaluated models trained on random subsets of 5,000 samples for both 24 and 8 input features. The choice of 5,000 was motivated by our quantum results, where this sample size proved most effective. For robustness, we averaged results over multiple random subsets to account for statistical variation. Interestingly, the model trained on the first 1,000 samples displayed stronger performance than random subsets of equal size, which we attribute to favorable statistical properties of those initial samples.

Table S3: Performance of the 24-feature model trained on 5,000-sample subsets. Results are shown for individual random runs, the first contiguous subset, and the averaged metrics.

Features	**Samples**	**Subset**	**F1**	**AUROC**
24	5,000	Random (1)	0.79	0.872
24	5,000	Random (2)	0.78	0.877
24	5,000	Random (3)	0.72	0.755
24	5,000	Random (4)	0.76	0.846
24	5,000	First	0.78	0.961
24	5,000	Average	0.77	0.862

Table S4: Performance of the 8-feature model trained on 5,000-sample subsets. Results are shown for individual random runs, the first contiguous subset, and the averaged metrics.

Features	**Samples**	**Subset**	**F1**	**AUROC**
8	5,000	Random (1)	0.78	0.914
8	5,000	Random (2)	0.77	0.939
8	5,000	Random (3)	0.76	0.816
8	5,000	Random (4)	0.74	0.925
8	5,000	First	0.79	0.913
8	5,000	Average	0.77	0.901

E Quantum Autoencoder with Feature Reduction using Classical Autoencoder Results

Table S5: Comparison of QAE models across different encoding techniques where the CAE was used to reduce dataset dimensions.

Features	**Encoding**	**Ansatz**	**Repetitions**	**F1**	**AUROC**
8	Amplitude	RA	1	0.76	0.914
8	Amplitude	RA	3	0.58	0.413
8	Amplitude	Pauli	1	0.71	0.859
8	Amplitude	Pauli	3	0.76	0.110
16	Amplitude	RA	1	0.70	0.851
16	Amplitude	RA	3	0.71	0.788
32	Amplitude	RA	1	0.69	0.132
32	Amplitude	RA	3	0.68	0.540
8	Angle	RA	1	0.69	0.537
8	Angle	RA	3	0.69	0.583
8	Angle	Pauli	1	0.53	0.552
8	Angle	Pauli	3	0.71	0.120
8	Dense	RA	1	0.73	0.895
8	Dense	RA	3	0.12	0.086
8	Dense	Pauli	1	0.86	0.918
8	Dense	Pauli	3	0.71	0.131
8	Dense	Pauli	5	0.66	0.893
16	Dense	RA	1	0.31	0.253
16	Dense	RA	3	0.08	0.286
16	Dense	Pauli	1	0.65	0.094
16	Dense	Pauli	3	0.49	0.395
16	EffSU2	RA	1	0.54	0.429
16	EffSU2	RA	3	0.09	0.049
24	EffSU2	RA	1	0.51	0.597
24	EffSU2	RA	3	0.63	0.814
24	EffSU2	Pauli	1	0.66	0.453

F Quantum Autoencoder Sample Size Effects

Table S6: QAE performance across different sample sizes and ansatz repetitions. All experiments used the first 8 features with Dense-Angle Encoding and a Real Amplitudes ansatz.

Samples	**Repetitions**	**F1**	**AUROC**
1,000	1	0.86	0.960
5,000	1	0.74	0.902
5,000	**2**	**0.87**	**0.965**
5,000	3	0.84	0.930
10,000	2	0.84	0.952
10,000	3	0.84	0.912
10,000	4	0.83	0.867
10,000	5	0.86	0.951
20,000	5	0.79	0.949
20,000	6	0.85	0.961
20,000	7	0.84	0.911

References

Kate Highnam, Kai Arulkumaran, Zachary Hanif, and Nicholas R. Jennings. Beth dataset: Real cybersecurity data for unsupervised anomaly detection research. *Proceedings of the Conference on Applied Machine Learning for Information Security (CAMLIS)*, 2021. URL `https://ceur-ws.org/Vol-3095/paper1.pdf`.