Neuro-Evolution for Multivariate Time Series Anomaly Detection

By Aryan Jha

Thesis Committee:
Dr. Travis Desell
Dr. Alexander Ororbia
Dr. Rui Li

Agenda

- Overview
- Motivation
- Related Work
- Methodology
- Results
- Conclusions
- Future Work

Overview

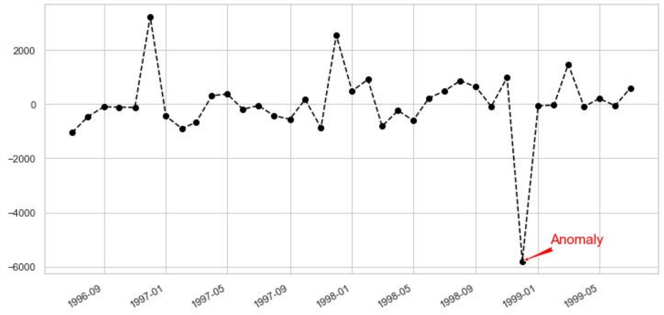


Fig. 1 Anomaly Detection by Aayush Bajaj, neptune.ai, https://neptune.ai/blog/anomaly-detection-in-time-series

What is Anomaly Detection?

- Identifies deviations from normal behavior in time-series data
- Traditionally removes outliers; now focuses on critical issues like fraud or system failures

Why It’s Important:

- Prevents financial losses, data breaches, and operational failures
- Enables proactive responses to anomalies
- Key applications: fraud detection, intrusion detection, system monitoring, process optimization, and predictive maintenance

Overview

Types of Time Series Anomalies

- Abnormal Time Points: Individual points with values significantly different from the rest
- Abnormal Time Intervals: Time spans with consistently unusual patterns
- Abnormal Time Series: Entire series deviating significantly from others

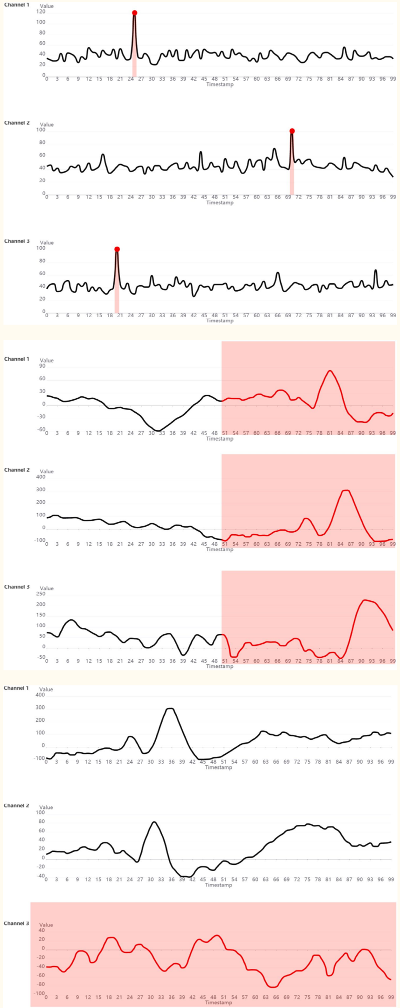


Fig. 2(a) Example of Abnormal Time Point, 2(b) Example of Abnormal Time Interval, and 2(c) Example of Abnormal Time Series by Gen Li and Jason Jung, Information Fusion, https://www.sciencedirect.com/science/article/pii/S1566253522001774

Motivation

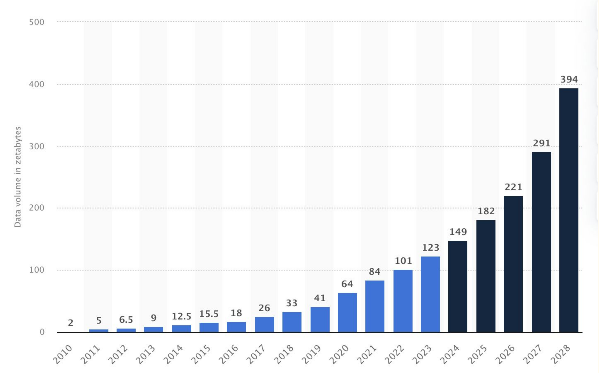


Fig. 2 Volume of Data Created by Petroc Taylor, statistica, https://www.statista.com/statistics/871513/worldwide-data-created/

The Big Data Era

- Global Growth:
  - New data created in 2024 estimated to be 149 zettabytes (149 trillion GBs)[1]
  - Exponential rate of growth
- Impacts critical sectors: IoT, healthcare, finance, navigation, and security

Automation is Essential

- Modern sensors generate vast, continuous data streams
- Manual detection is infeasible and prone to errors, risking severe consequences

[1]: David Reinsel, John Gantz, and John Rydning. Data age 2025: The evolution of data to life-critical. Don't Focus on Big Data, 2, 2017.

Motivation

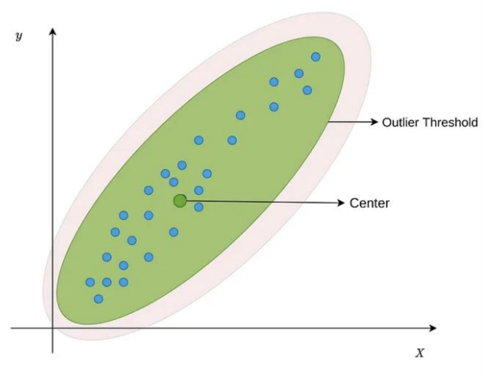
Challenges in Anomaly Detection

- Defining Normal:
  - Strict thresholds are difficult to establish in interdependent multivariate datasets
- Decision Boundaries:
  - Balancing accuracy and false alarms near anomaly boundaries
- Domain-Specific Characteristics:
  - Anomalies vary across domains, limiting adaptability of current methods

Related Work

Distance-Based Methods

- Measure rarity of data points based on distance to neighbors
- k-Nearest Neighbor (kNN) (Ramaswamy et al.)[2]:
    - Ranks and flags anomalies based on their proximity to k closest points



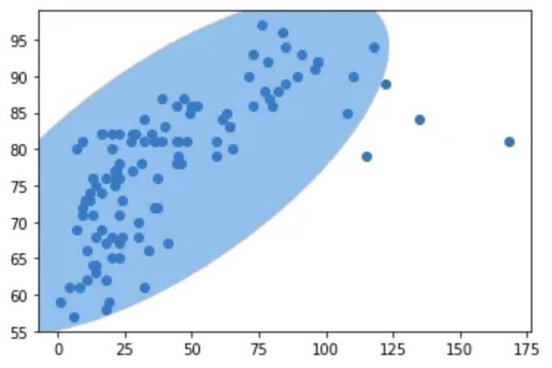


Fig. 3 (a) & (b) Distance-Based Anomaly Detection by Sergen Cansiz, Towards Data Science, https://towardsdatascience.com/multivariate-outlier-detection-in-python-e946cfc843b3

[2]: Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, SIGMOD '00, page 427–438, New York, NY, USA, 2000. Association for Computing Machinery.

Related Work

Clustering-Based Methods

- Group similar data into clusters; detect sparse or distant data points as anomalies
- Multi-Level Conformal Clustering (MLCC) (Nouretdinov et al.)[3]:
  - Applies the Conformal Prediction framework to the unsupervised learning task of clustering
  - Simultaneously clusters and detects anomalies without assumptions about data distribution

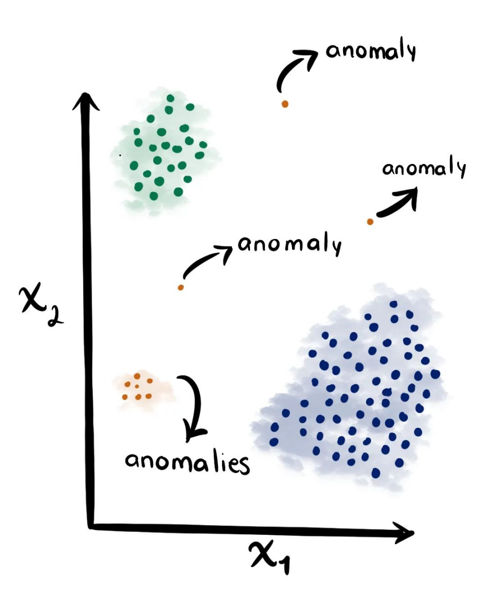


Fig. 4 Two-dimensional Dataset and its Anomalies by Isaac Arroyo, Towards Data Science, https://towardsdatascience.com/unsupervised-anomaly-detection-on-spotify-data-k-means-vs-local-outlier-factor-f96ae783d7a7

[3]: Ilia Nouretdinov, James Gammerman, Matteo Fontana, and Daljit Rehal. Multi-level conformal clustering: A distribution-free technique for clustering and anomaly detection. Neurocomputing, 397:279–291, 2020.

Related Work

Density-Based Methods

- Identify anomalies as deviations from data density peaks
- Gaussian Mixture Model (GMM) (Yang et al.)[4]:
  - Identifies anomalies as outliers in Gaussian clusters
- Kernel Density Estimation (KDE) (Latecki et al.)[5]:
  - Detects outliers via deviations in local density estimates

Fig. 5 Bivariate Kernel Density Estimate by Rob J. Hyndman, otexts, https://otexts.com/weird/05-density.html

[4]: Xingwei Yang, Longin Jan Latecki, and David Pokrajac. Outlier detection with globally optimal exemplar-based gmm. In SIAM International Conference on Data Mining, Conference Proceedings, pages 145–154, 04 2009.
[5]: Longin Jan Latecki, Aleksandar Lazarevic, and Dragoljub Pokrajac. Outlier detection with kernel density functions. In Petra Perner, editor, Machine Learning and Data Mining in Pattern Recognition, pages 61–75, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

Related Work

Reconstruction-Based Methods

- Detect anomalies with high reconstruction error
- Principal Component Analysis (PCA) (Dani et al.)[6]:
  - Highlights anomalies through deviations in reduced-dimensional space
- Matrix Factorization (Liang et al.)[7]:
  - Detects anomalies by approximating data with a low-rank structure
- LSTM Autoencoder: Explored further in slide 13

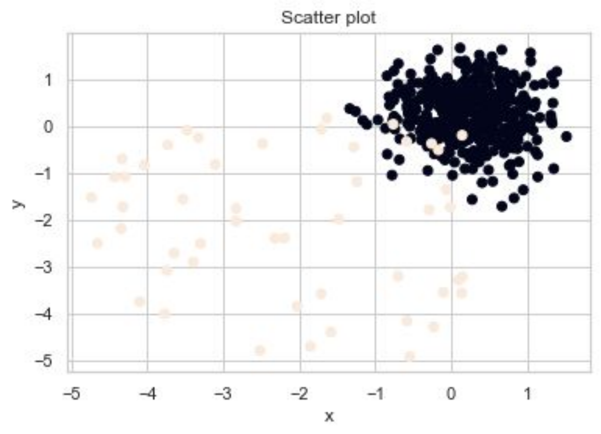


Fig. 6 PCA Reduction by Aayush Bajaj, neptune.ai, https://neptune.ai/blog/anomaly-detection-in-time-series

[6]: Saurav Kumar Dani, Chander Thakur, Naman Nagvanshi, and Gurwinder Singh. Anomaly detection using pca in time series data. In 2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), volume 2, pages 1–6, 2024.
[7]: Liang Xiong, Xi Chen, and Jeff Schneider. Direct robust matrix factorization for anomaly detection. In 2011 IEEE 11th International Conference on Data Mining, pages 844–853, 2011.

Related Work

Ensemble Learning

- Combine multiple models to enhance detection robustness
- Isolation Forest (iForest) (Liu et al.)[8]:
  - Random splits isolate anomalies
- Deep Isolation Forest (Xu et al.)[9]:
  - Extends this approach for complex datasets using neural mappings

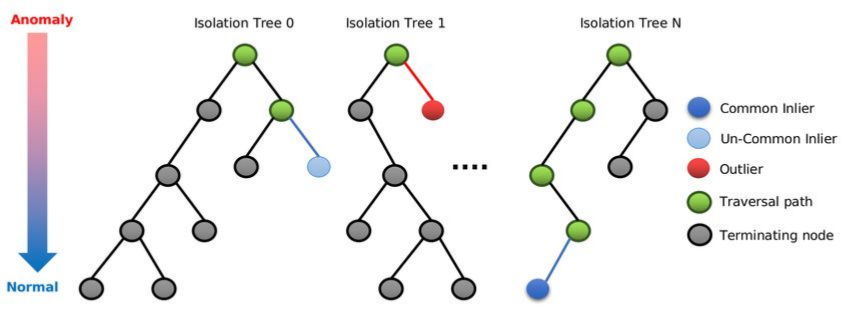


Fig. 7 Isolation Forest for Anomaly Detection by Dairi et al., IEEE Access, https://www.researchgate.net/publication/365288037_Efficient_Driver_Drunk_Detection_by_Sensors_A_Manifold_Learning-Based_Anomaly_Detector

[8]: Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation-based anomaly detection. ACM Trans. Knowl. Discov. Data, 6(1), mar 2012.
[9]: Hongzuo Xu, Guansong Pang, Yijie Wang, and Yongjun Wang. Deep isolation forest for anomaly detection. IEEE Transactions on Knowledge and Data Engineering, 35(12):12591–12604, 2023.

Related Work

Dynamic Graph-Based Approaches

- Detects anomalies in multivariate time series via deviations in correlation patterns
- Deng and Hooi (2021): Combine structure learning and *Graph Neural Networks (GNNs)*[10]
- Zhao et al. (2020): Use *Graph Attention Networks (GAT)* and *Graph Convolutional Networks (GCN)*[11]
- Chen et al. (2022): Utilized a transformer-based architecture[12]

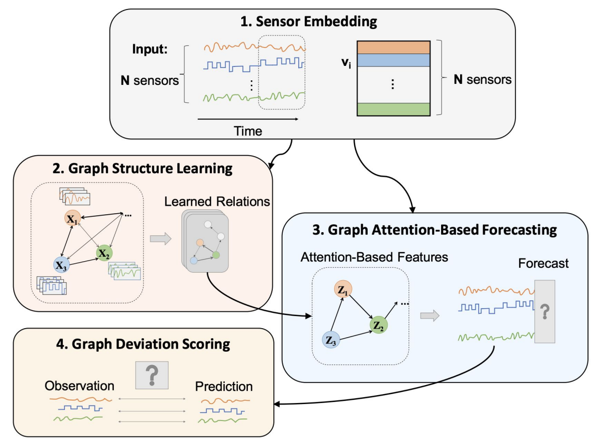


Fig. 8 Framework Overview by Deng and Hooi[12]

[10]: Ailin Deng and Bryan Hooi. Graph neural network-based anomaly detection in multivariate time series. Proceedings of the AAAI Conference on Artificial Intelligence, 35(5):4027–4035, May 2021.
[11]: Hang Zhao, Yujing Wang, Juanyong Duan, Congrui Huang, Defu Cao, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. Multivariate time-series anomaly detection via graph attention network. In 2020 IEEE International Conference on Data Mining (ICDM), pages 841–850, 2020
[12]: Zekai Chen, Dingshuo Chen, Xiao Zhang, Zixuan Yuan, and Xiuzhen Cheng. Learning graph structures with transformer for multivariate time-series anomaly detection in iot. IEEE Internet of Things Journal, 9(12):9179–9189, 2022.

Related Work

LSTM-based Approaches

- Capture long-term dependencies and temporal features for anomaly detection
- Lin et al. (2020): Use a VAE-LSTM hybrid to capture anomalies across multiple time scales[13]
- Niu et al. (2020): Propose an LSTM-based VAE-GAN to model normal distributions for anomaly detection[14]

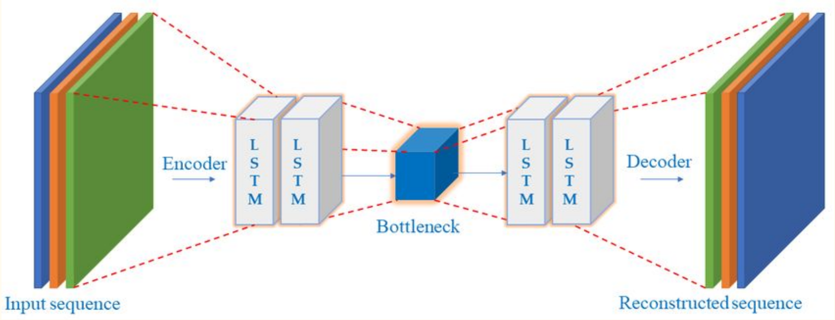


Fig. 9 LSTM-Autoencoder Architecture by Do et al., MDPI, https://www.researchgate.net/publication/367170294_LSTM-Autoencoder_for_Vibration_Anomaly_Detection_in_Vertical_Carousel_Storage_and_Retrieval_System_VCSRS

[13]: Shuyu Lin, Ronald Clark, Robert Birke, Sandro Schönborn, Niki Trigoni, and Stephen Roberts. Anomaly detection for time series using vae-lstm hybrid model. In ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4322–4326, 2020.
[14]: Zijian Niu, Ke Yu, and Xiaofei Wu. Lstm-based vae-gan for time-series anomaly detection. Sensors, 20(13), 2020.

Methodology

Evolutionary eXploration of Augmenting Memory Models (EXAMM)

- EXAMM, developed by Ororbia, El Said, and Desell[15], evolves Recurrent Neural Networks (RNNs) for time series forecasting
- Uses various memory structures like Δ-RNN, GRU, LSTM, MGU, and UGRNN
- Utilizes multiple population islands for improved performance over single steady-state models
- Genetic Operations:
  - Mutation: Alters nodes and edges (e.g., add/remove, split/merge, enable/disable)
  - Crossover: Combines parent RNNs by recombining weights
  - Cloning: Duplicates parent genomes
- RNNs are replaced based on fitness, mimicking natural selection

[15]: Alexander Ororbia, AbdElRahman ElSaid, and Travis Desell. Investigating recurrent neural network memory structures using neuro-evolution. In Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '19, page 446–455, New York, NY, USA, 2019. Association for Computing Machinery.

Methodology

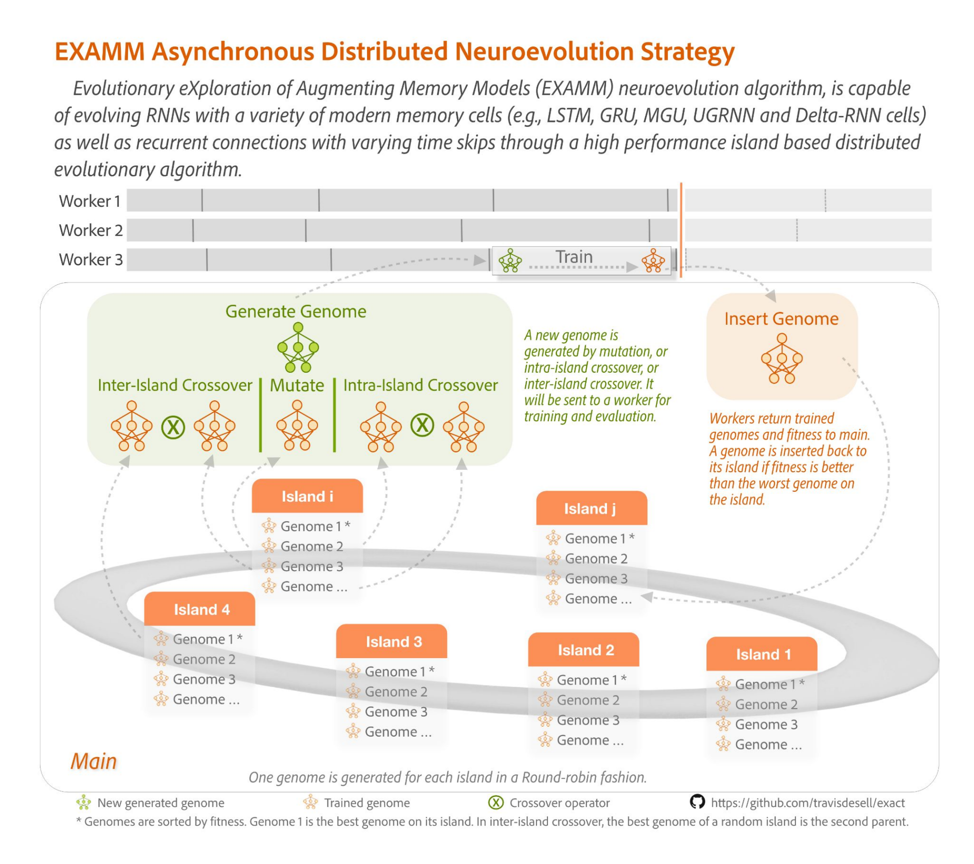


Fig. 10 EXAMM Neuroevolution Strategy by Zimeng Lyu, https://zimenglyu.com/

Methodology

EXA-STAR

- Limited implementation of EXAMM using PyTorch
  - Evolves and trains regular recurrent genomes
  - Utilizes single steady-state models instead of multiple population islands
- Performs the same genetic operations and evolutionary processes of genome replacement based on fitness

Methodology

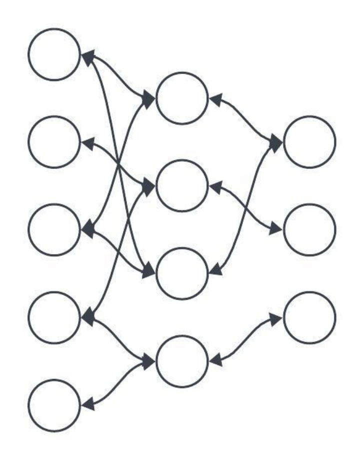
Implementation Details - LSTM

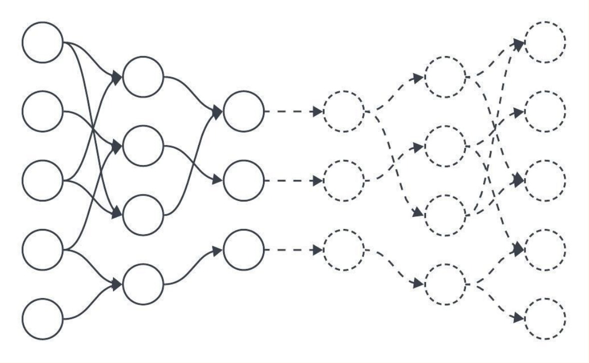
- Implemented an LSTM Node, using an individual LSTM Cell with multiple input and output edges
- The LSTM node has a 50% probability of being added alongside regular nodes
- LSTM nodes enhance the model's capability for capturing long-term dependencies in time series forecasting

Methodology

Implementation Details - Symmetrical Bidirectional Autoencoder

- Each node participates as both an individual encoder and an individual decoder, leveraging bidirectional symmetry
- Maintains symmetry during mutation and crossover operations, ensuring proper reconstruction of inputs via shared representations
- Implemented Bidirectional Node class





Methodology

Implementation Details - Regular Autoencoder

- Architecture
  - The autoencoder strictly separates the encoder and decoder layers
  - Connections across the encoding layer are disallowed
  - The encoding layer's connections are frozen to maintain strict separation
- Mutation Constraints
  - All mutation and genetic operations (add/remove, split/merge) are constrained to operate strictly within either the encoder or decoder
  - Operations are governed by autoencoder architecture rules to ensure separation between the encoder and decoder

Methodology

Implementation Details - Regular Autoencoder

- Add Node: Creates a node at a random depth (0–1); nodes < 0.5 join the encoder, while nodes > 0.5 join the decoder, connecting only within their respective layers
- Add Edge/Recurrent Edge: Operate within a randomly assigned range (encoder: 0–0.5, decoder: 0.5–1), connecting nodes only within the chosen range
- Merge Node: Combines two nodes within the same layer (encoder or decoder)
- Enable/Disable/Split Node/Edge: Randomly picks an existing node/edge to enable/disable/split

Methodology

LSTM Autoencoder for Anomaly Detection

- Nguyen et al.[16] demonstrate an LSTM Autoencoder network-based method combined with a One-Class Support Vector Machine (OCSVM) algorithm for detecting anomalies in sales data
  - Combines LSTM cells in both encoder and decoder networks to handle temporal data and capture long-term dependencies
  - Trained on normal data, it produces higher reconstruction errors for anomalous sequences, enabling their identification
  - OCSVM finds an optimal hyperplane in feature space to separate normal data from outliers
  - OCSVM is used on reconstruction errors to identify anomalies

[16]: H.D. Nguyen, K.P. Tran, S. Thomassey, and M. Hamad. Forecasting and anomaly detection approaches using lstm and lstm autoencoder techniques with the applications in supply chain management. International Journal of Information Management, 57:102282, 2021

Methodology - Implementation Summary

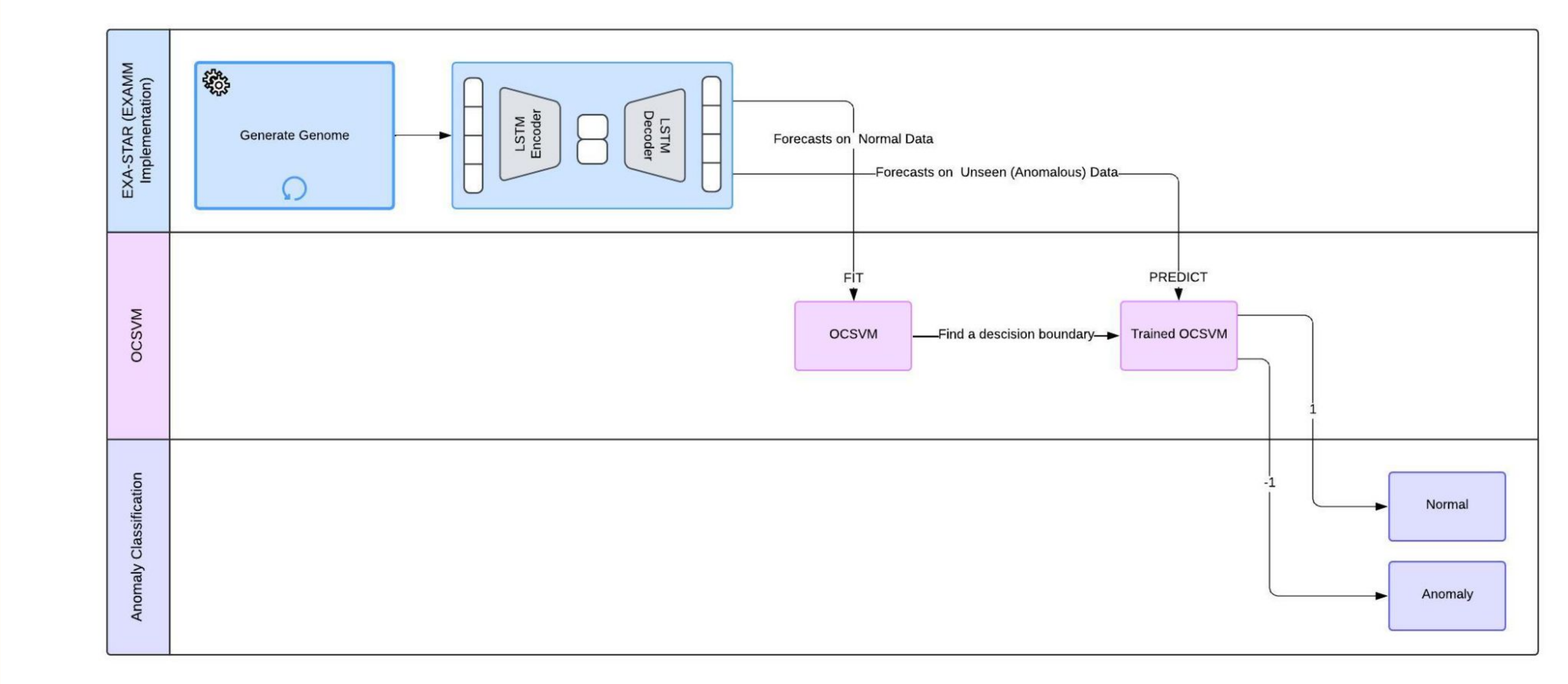
EXASTAR-Evolved LSTM Autoencoder

- Combines autoencoder (for local feature capture) with LSTM (for long-term trend extraction) to analyze multivariate time-series data
- Trained with normalized inputs to minimize mean squared error (MSE) loss
- EXA-STAR fine-tunes network parameters like cell count and dropout

OCSVM for Anomaly Detection

- Uses residual errors from input-reconstructed sequences to train a One-Class SVM (OCSVM) on healthy data
- Detects anomalies by identifying outliers in the residual error space

Methodology



Experimental Design

Datasets Used

- Controlled Anomalies Time Series (CATS) Dataset: Simulated system with 200 labeled anomaly sequences across 5M timestamps. Trained on 30K nominal timestamps, tested on 13K with 50% anomaly rate
- Soil Moisture Active Passive (SMAP) & Mars Surface Lander (MSL) Datasets: NASA telemetry datasets with expert-labeled anomalies, anonymized metadata, and one-hot encoded commands. Trained and tested on the Power telemetry.

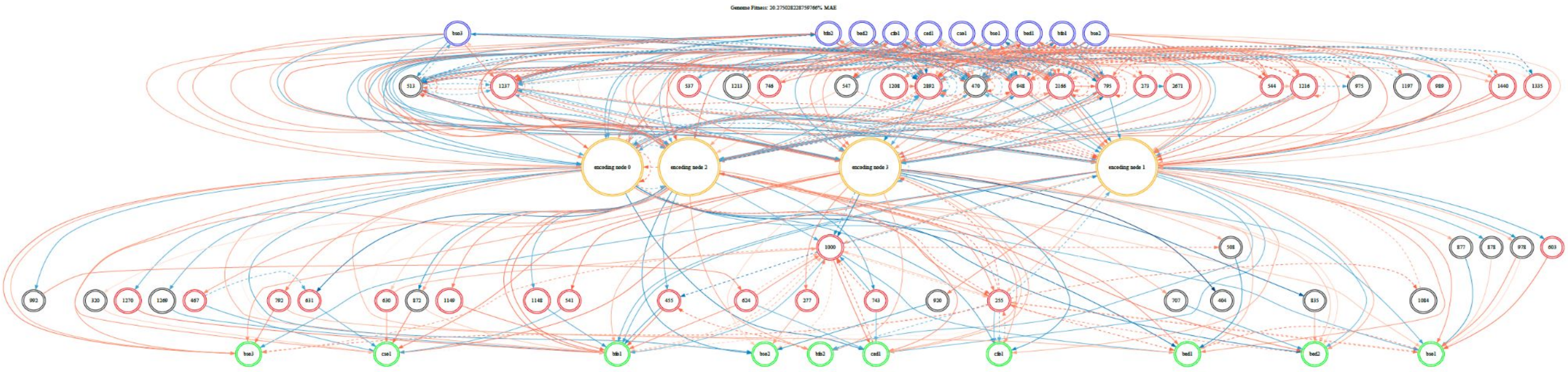
Methodology

Graphviz Implementation

- Layer Structure: Input (Blue), Encoding (Orange), and Output (Green) layers are clearly separated
- Encoder (depth: 0-0.5) and Decoder (depth: 0.5-1.0) divided by the Encoding layer (depth: 0.5)
- Node Representation: LSTM Nodes (Red) highlight memory components in Encoder/Decoder
- Edge Validation: Successfully verified no cross-layer connections between Encoder and Decoder—ensuring robust architecture
- Edge Weights: Color-coded (Blue for positive, Red for negative)
- Time-Skip Edges: Dashed and labeled for clarity

Results

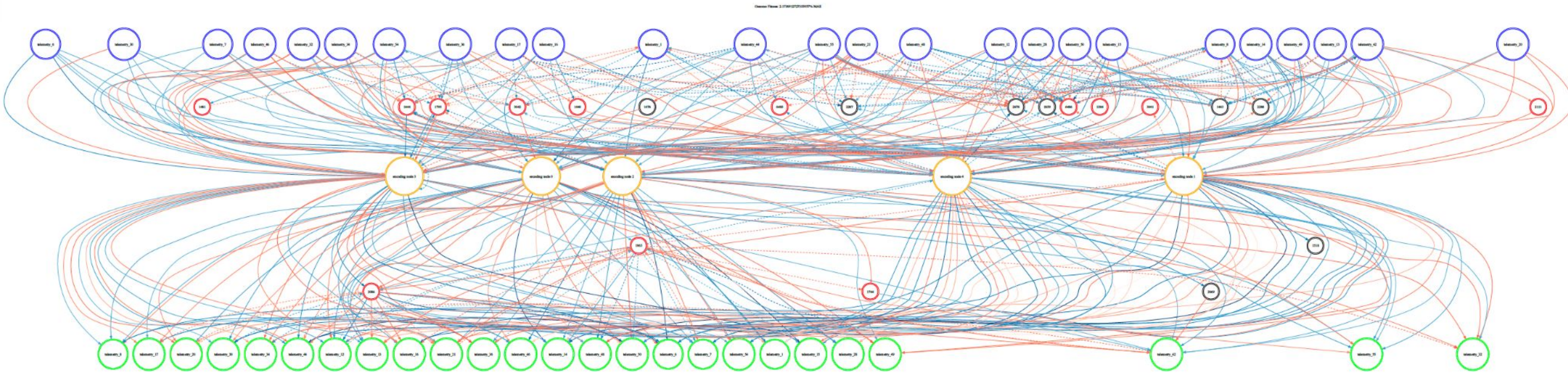
Generated Network - CATS Dataset



- Input Nodes: Blue; Output Nodes: Green; Encoding Layer (Bottleneck Layer): Orange
- LSTM Nodes: Red; Regular Nodes: Black

Results

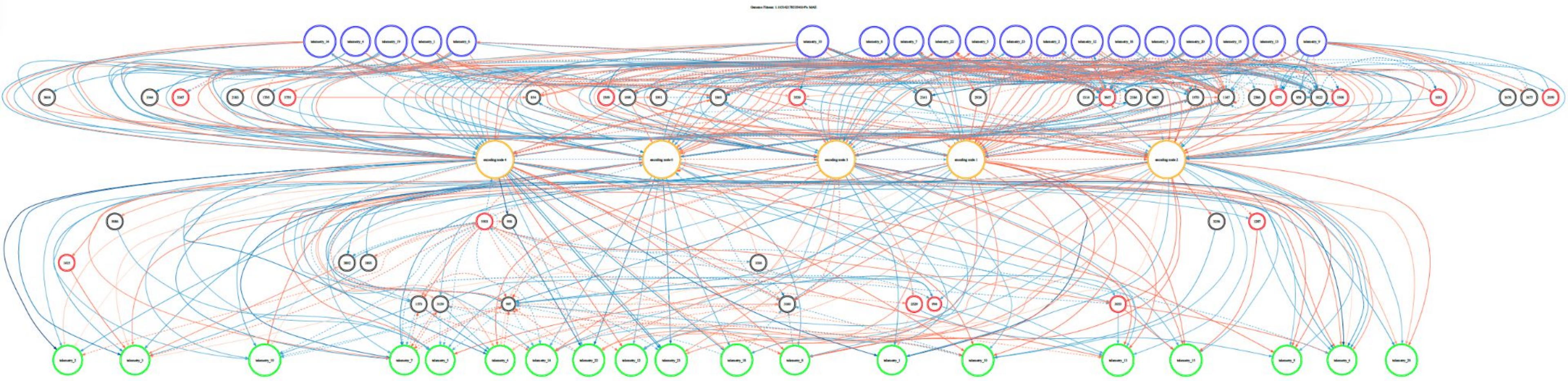
Generated Network - MSL Dataset



- Input Nodes: Blue; Output Nodes: Green; Encoding Layer (Bottleneck Layer): Orange
- LSTM Nodes: Red; Regular Nodes: Black

Results

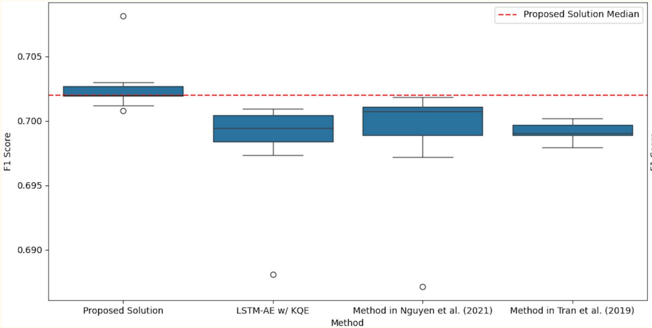
Generated Network - SMAP Dataset



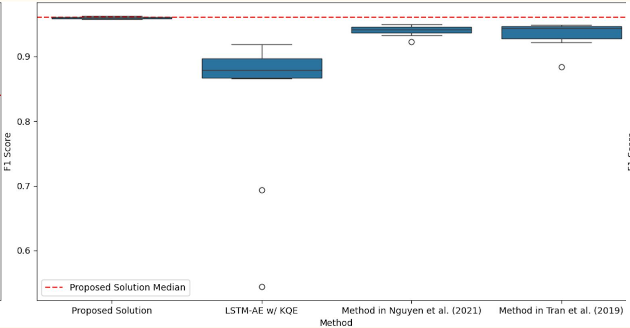
- Input Nodes: Blue; Output Nodes: Green; Encoding Layer (Bottleneck Layer): Orange
- LSTM Nodes: Red; Regular Nodes: Black

Results

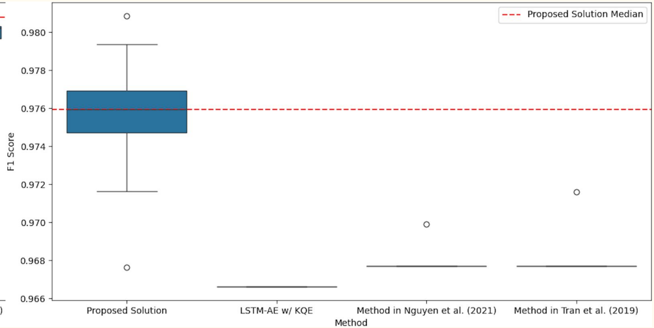
F1 Scores



CATS Dataset



MSL Dataset



SMAP Dataset

H.D. Nguyen, K.P. Tran, S. Thomassey, and M. Hamad. Forecasting and anomaly detection approaches using lstm and lstm autoencoder techniques with the applications in supply chain management. International Journal of Information Management, 57:102282, 2021

Kim Phuc Tran, Huu Du Nguyen, and Sebastien Thomassey. Anomaly detection using long short term memory networks and its applications in supply chain management. IFAC-PapersOnLine, 52(13):2408–2412, 2019. 9th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2019.

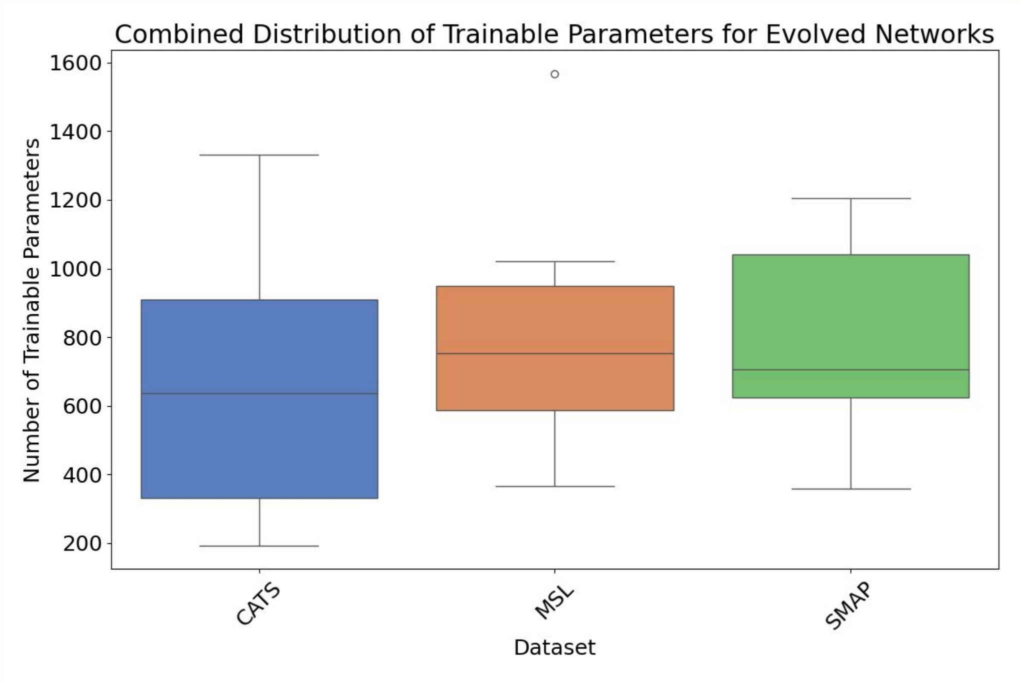
Results

Number of Trainable Parameters

Dataset	Proposed Method (Best Performing Genome)	LSTM-AE w/ KQE	Method in Nguyen et al. (2021)	Method in Tran et al. (2019)
CATS	903	1608202	1608202	491714
MSL	580	739097	739097	497069
SMAP	715	50771	50771	494927

Far fewer number of trainable parameters in the networks generated using the proposed method

Results

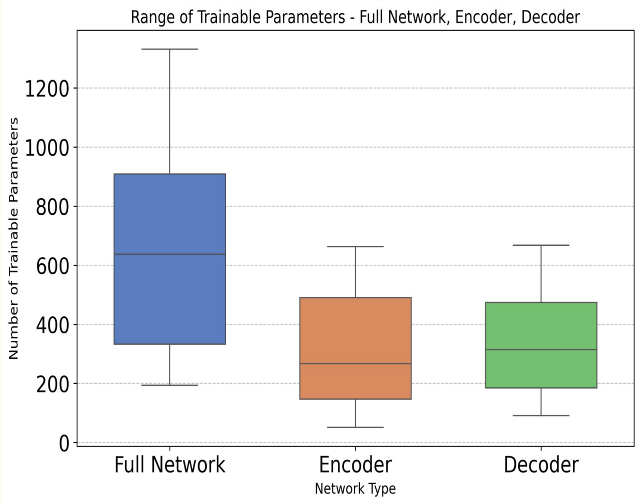


Averages around 700 - significantly lower than benchmark methods

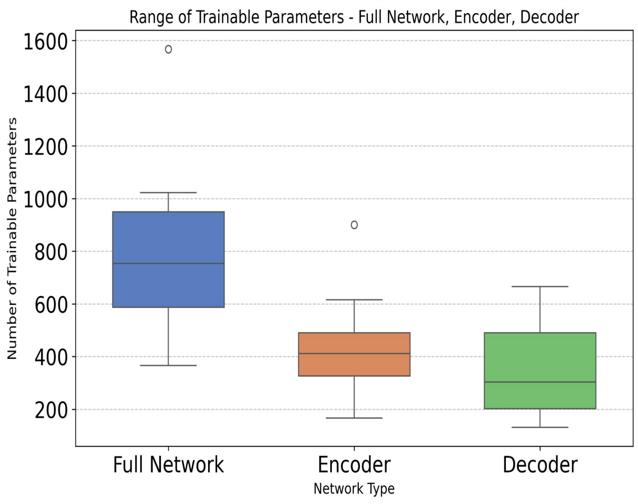
Range of trainable parameters in EXA-STAR networks

Results

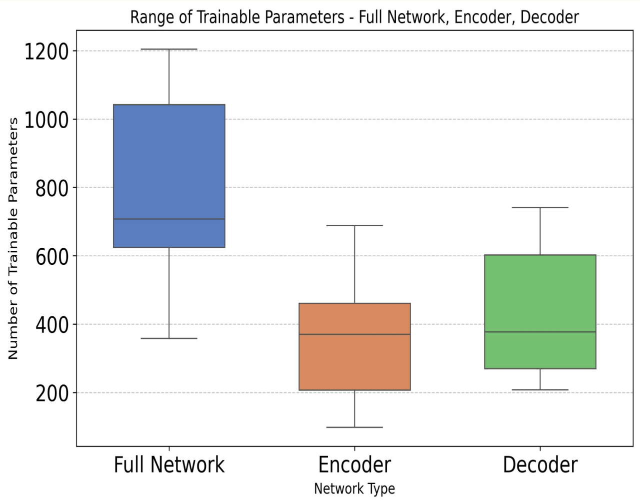
Detailed look at the range of trainable parameters in EXA-STAR networks



CATS Dataset



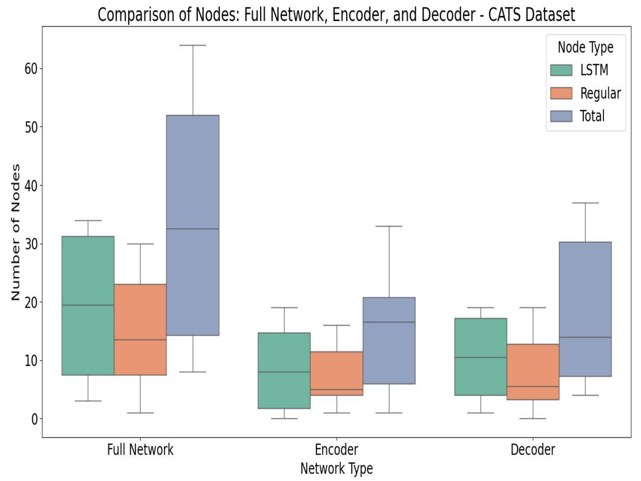
MSL Dataset

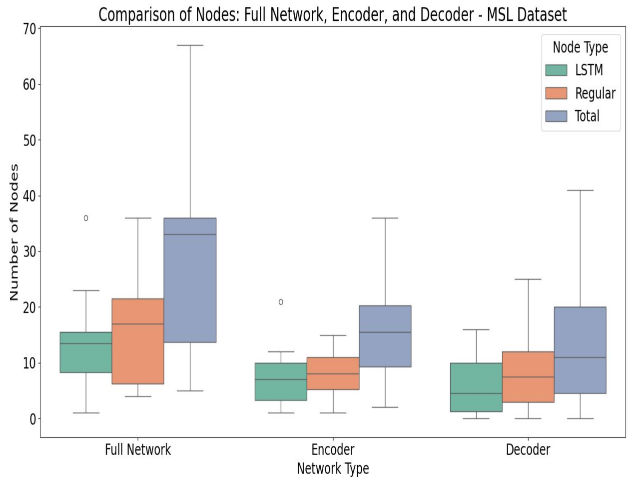


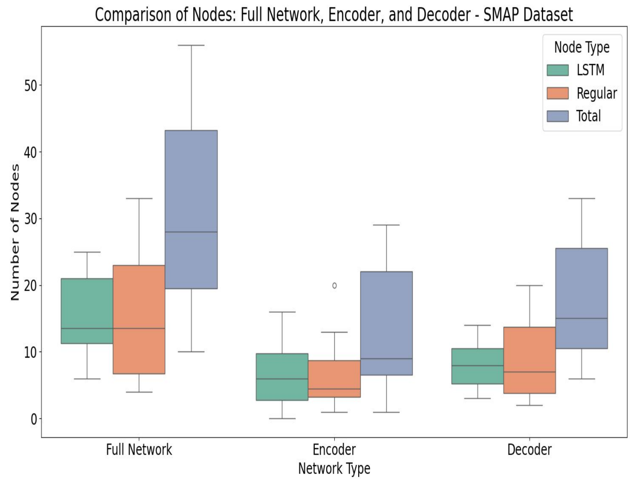
SMAP Dataset

Results

Node Comparison - EXA-STAR Networks

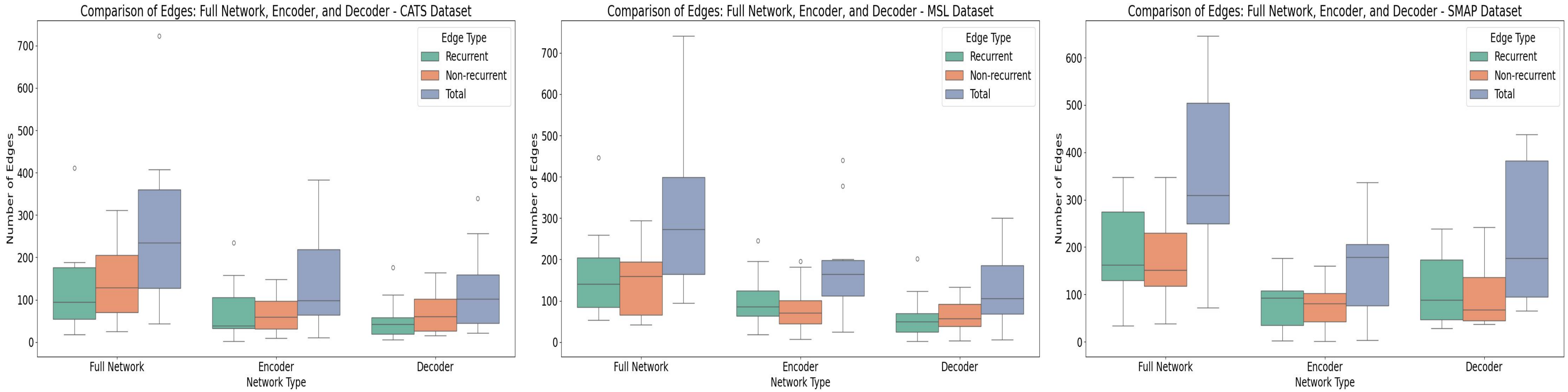






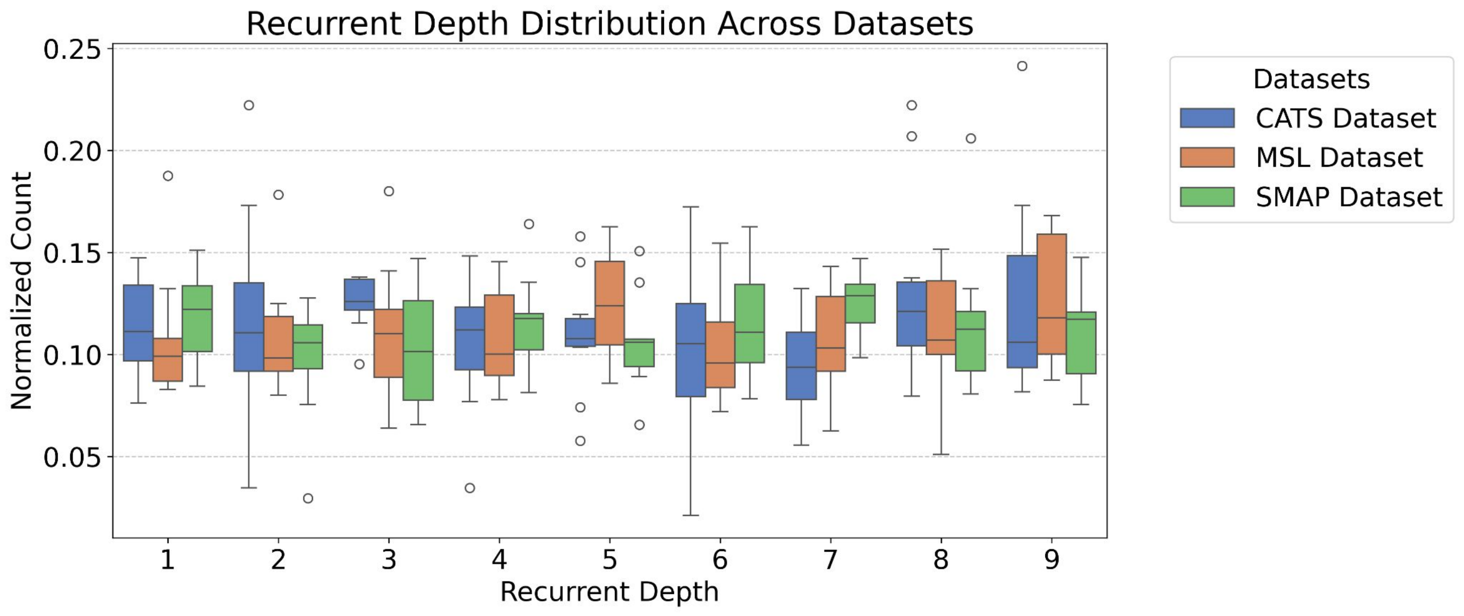
Results

Edge Comparison - EXA-STAR Networks



Results

Ranges of Recurrent Depths on Recurrent Edges - EXA-STAR networks



Results

Node and Edge Statistics (Best Performing Networks):

Dataset	Genome	LSTM Nodes	Regular Nodes	Regular Edges	Recurrent Edges	Encoder Nodes	Decoder Nodes	Encoder Edges	Decoder Edges
CATS	714	29	17	250	189	18	28	291	148
MSL	885	12	7	308	80	15	4	227	161
SMAP	937	14	30	342	149	29	15	309	182

Conclusions

- **EXA-AE effectively identifies anomalies** in diverse and challenging datasets.
- EXA-AE produces networks with **significantly fewer trainable parameters** compared to traditional LSTM-based methods
- EXA-AE results in **more efficient, less computationally expensive networks**.
- Networks generated by EXA-AE **show a preference for recurrent edges**, which help capture long-term dependencies better than non-recurrent edges
- An **even balance between LSTM nodes and simple neurons** was found optimal for capturing complex dependencies while maintaining good generalization
- EXA-AE **tends to favor networks with larger decoders than encoders**, particularly in CATS and SMAP datasets, indicating a larger decoder aids in capturing complex data patterns and improving forecasting performance
- EXA-AE **demonstrates high flexibility** in network architecture, accommodating diverse datasets effectively

Future Work

- **Symmetrical Autoencoders**: Explore the use of bidirectional nodes in autoencoders, as implemented in EXA-STAR, for generating symmetrical architectures
- **Memory Cell Variants**: Extend EXA-STAR to include additional memory cells (e.g., GRU, MGU, UGRNN) and evaluate their impact
- **Population Islands**: Incorporate multi-population islands into EXA-STAR for enhanced performance, aligning with EXAMM's framework
- **Scalability**: Scale testing with larger datasets and improved computational resources to validate robustness and generalizability
- **Alternative Thresholding**: Investigate other methods like Kernel Quantile Estimation (KQE) to refine anomaly detection decision boundaries, beyond OCSVM
- **Attention Networks/Transformer Architectures**: Explore attention mechanisms and transformer-based architectures to better map relationships in time series data. Evaluate their ability to capture global context compared to traditional RNN-based approaches.
- **Optimal Size of Encoding Layer**: Conduct a detailed study on the optimal size of the encoding layer, balancing dimensionality reduction with reconstruction accuracy. This includes assessing the effect of encoding layer size on anomaly detection performance

Questions

Thank you for your participation. Please feel free to ask any questions about the project.