DATASCI 507 Final Project Proposal
Anomaly Detection in Time Series Data using Transformers

Annie Ye yuhanye@umich.edu

1 Overview

1.1 Background & Motivation

Time series data is prevalent across various domains such as finance, healthcare, manufacturing, and network security. Detecting anomalies in such data is crucial for identifying potential issues, preventing failures, and ensuring timely interventions. Traditional anomaly detection methods often struggle with the complexities of temporal dependencies and high-dimensional data. Transformer models, originally developed for natural language processing tasks, have shown promising results in capturing long-range dependencies and learning meaningful representations from sequential data.

This project aligns with my research interests in high-dimensional stochastic processes, time series forecasting, and machine learning. By developing a Transformer-based anomaly detection system, I aim to address the challenges of detecting anomalies in real-time data streams while expanding my knowledge in time series analysis. The project also provides an opportunity to enhance my technical skills in Python and data preprocessing, contributing to the advancement of anomaly detection techniques across multiple domains.

1.2 Data and Model Selection

This project utilizes the Numenta Anomaly Benchmark (NAB) dataset[1], specifically designed for evaluating anomaly detection algorithms. The NAB dataset contains labeled anomalies for each time series, making it suitable for supervised learning approaches. For the model, I will employ Transformer architectures, which have demonstrated remarkable performance in various sequence modeling tasks. Transformers can effectively capture long-range dependencies and learn complex patterns in time series data, making them a promising choice for anomaly detection.

[1] https://github.com/numenta/NAB

1.3 Expected Research Outcomes

Through this project, I expect to gain the following insights:

- Assess the effectiveness of Transformer models in capturing temporal patterns and detecting anomalies in time series data compared to traditional anomaly detection methods and LSTM-based approaches.
- Identify the strengths and limitations of Transformer-based anomaly detection across different domains and data characteristics.
- Evaluate the impact of data preprocessing techniques, model architectures, and hyperparameter settings on the performance of the anomaly detection system.
- Gain insights into the interpretability of Transformer models in the context of anomaly detection, such as understanding which temporal patterns contribute to anomalous behavior.

2 Prior Work

2.1 Literature Review

In recent years, Transformer models have revolutionized various domains, including natural language processing, computer vision, and time series analysis. The seminal work by Vaswani et al. introduced the Transformer architecture, which relies on self-attention mechanisms to capture long-range dependencies in sequential data [1]. This has opened up new possibilities for anomaly detection in time series data.

Several studies have explored the application of Transformers for anomaly detection. Wen et al. proposed a Transformer-based architecture for multivariate time series anomaly detection, demonstrating its effectiveness in capturing complex temporal patterns and identifying anomalies [2]. Li et al. introduced a self-supervised learning approach using Transformers for unsupervised anomaly detection, leveraging the model's ability to learn meaningful representations from unlabeled data [3].

Despite the promising results, there are still challenges and opportunities for further research in Transformer-based anomaly detection. These include adapting Transformer architectures for real-time detection, incorporating domain-specific knowledge, and improving model interpretability. Hundman et al. explored the use of LSTMs and nonparametric dynamic thresholding for detecting spacecraft anomalies, highlighting the importance of considering domain-specific characteristics and the need for interpretable models [4]. Addressing these challenges can lead to more robust and practical anomaly detection systems.

2.2 Proposed Methodologies

To achieve the project goal of developing a Transformer-based anomaly detection system for time series data, I plan to explore the following potential methods:

- Supervised anomaly detection: Utilize the labeled anomaly data in the NAB dataset to train a Transformer model as a binary classifier to distinguish between normal and anomalous instances. This approach leverages the Transformer's ability to learn complex temporal patterns and classify anomalies based on the learned representations.
- Self-supervised learning: Employ self-supervised learning techniques, such as masked language modeling or contrastive learning, to pre-train the Transformer model on unlabeled time series data. This allows the model to learn meaningful representations that can be fine-tuned for anomaly detection tasks, potentially improving performance and generalization.
- Hybrid approaches: Combine supervised and self-supervised learning techniques to leverage both labeled and unlabeled data. For example, pre-training the Transformer model using self-supervised learning and then fine-tuning it with labeled anomaly data can enhance the model's ability to capture complex patterns and detect anomalies effectively.

3 Preliminary Results

3.1 Data Understanding

The Numenta Anomaly Benchmark (NAB) dataset consists of multiple real-world and artificial time series data files designed specifically for evaluating anomaly detection algorithms. Each data stream contains timestamps and scalar values, along with labeled anomalies that have been carefully annotated by domain experts.

3.1.1 Dataset Composition

The NAB corpus contains 58 time series data files across multiple domains:

- realAWSCloudwatch (17 files): CPU and network utilization metrics from various Amazon EC2 instances
- realAdExchange (5 files): Online advertisement clicking rates data
- realKnownCause (7 files): Metrics with known anomaly causes, including machine temperature sensor data and AWS server metrics
- realTraffic (7 files): Highway traffic data from Minneapolis DOT

- realTweets (10 files): Twitter volume data for popular tech companies
- artificialWithAnomaly (6 files): Artificially generated data with known anomaly patterns
- artificialNoAnomaly (6 files): Control group of artificial data without anomalies

3.1.2 Data Characteristics and Preprocessing Strategy

The Numenta Anomaly Benchmark (NAB) dataset provides 58 time series files spanning diverse domains like server metrics, advertisement clicks, and traffic data. These files vary in temporal resolution, ranging from 1-minute to hourly intervals, and include both normalized and raw values. Initial analysis shows no missing data and consistent timestamp formatting, but the dataset presents challenges such as varying anomaly characteristics (spikes, gradual shifts, and seasonal patterns) and differences in sampling rates.

To ensure the dataset is ready for effective anomaly detection, I will use the following preprocessing strategy:

- Resampling: Standardize the temporal resolution to 5-minute intervals for uniformity.
- Normalization: Apply min-max scaling to bring all metrics to a comparable scale.
- Sequence Windowing: Segment the time series into fixed-size windows suitable for LSTM input, with the optimal window size determined experimentally.
- Train-Validation Split: Carefully split the data, accounting for its temporal nature to avoid data leakage.
- Handling Probationary Periods: Properly manage the initial probationary periods outlined in the NAB documentation to maintain evaluation accuracy.

This approach ensures that the dataset's variability is handled effectively, preparing the LSTM model to detect anomalies across a wide range of scenarios and domains.

3.1.3 Insights and Challenges

The NAB dataset reveals diverse anomaly patterns across different domains, presenting both opportunities and challenges for anomaly detection using Transformer models.

The dataset includes a wide range of anomaly types, such as sharp spikes in AWS metrics, gradual degradations in advertisement data, clear daily and weekly cycles in traffic data, and recurring business-hour spikes in AWS workloads. Real-world datasets feature complex anomalies combining multiple changes, while artificial datasets present isolated, well-defined anomalies. Anomaly durations also vary from brief spikes to prolonged degradations.

These insights highlight several challenges that need to be addressed. Varying sampling rates and measurement scales require careful preprocessing to ensure data consistency. Seasonal patterns and long-term trends necessitate the use of Transformer architectures capable of capturing both short-term and long-term dependencies. Real-time detection poses challenges in terms of balancing speed and accuracy, requiring efficient inference strategies.

Furthermore, interpretability is a crucial aspect of anomaly detection systems. Understanding which temporal patterns contribute to anomalous behavior can provide valuable insights for domain experts. Developing methods to interpret Transformer models and visualize their attention mechanisms will be an important focus of this project.

3.2 Basic Model: Transformer

3.2.1 Resource Requirements

- Computing Resources: A GPU-enabled environment is required for efficient training of the Transformer model. We will utilize Google Colab, which provides access to GPU resources, for the development and experimentation phase. For larger-scale experiments and deployment, we may need to consider using cloud-based GPU instances or a dedicated GPU server.
- Libraries and Frameworks: The implementation will rely on popular deep learning libraries such as PyTorch and TensorFlow. Additionally, we will leverage the Hugging Face Transformers library, which provides pre-trained Transformer models and utilities for fine-tuning and adaptation. Other essential libraries include NumPy for numerical computations, Pandas for data manipulation, and Matplotlib for data visualization.

3.2.2 Performance Considerations

- Computational Complexity: Transformer models are known to be computationally expensive, especially when processing long sequences. To address this, we will explore techniques such as sequence chunking, where longer time series are divided into smaller subsequences. This allows for more efficient processing and reduces memory requirements. Additionally, we will investigate the use of efficient Transformer variants, such as the Linformer or the Reformer, which aim to reduce the computational complexity of self-attention mechanisms.
- Scalability: To ensure the scalability of the anomaly detection system, we will design the architecture to handle large-scale time series data. This involves optimizing data loading and preprocessing pipelines, utilizing distributed training techniques, and leveraging batch processing. We will also consider the use of data parallelism and model parallelism to distribute the workload across multiple GPUs if necessary.

3.3 Implementation Strategy

3.3.1 Tools from Class

The implementation of the Transformer-based anomaly detection system will utilize fundamental tools and concepts from the class:

- NumPy & SciPy: Used for numerical computations, data preprocessing, and implementing custom metrics for evaluating model performance.
- Plotly & Pyplot: For visualizing time series data, model predictions, and evaluation metrics, providing clear insights into the dataset and results.
- PyTorch: Serves as the core framework for designing, training, and fine-tuning the Transformer model with GPU acceleration.

3.3.2 Tools to Explore

To push the boundaries of this project, advanced tools and techniques outside the class curriculum will be explored:

- Hugging Face Transformers: This library offers pre-trained Transformer models and tools for fine-tuning. Leveraging these models can reduce training time and improve performance by utilizing knowledge from large-scale pretraining tasks.
- PyTorch Lightning: Provides a structured framework for training deep learning models, simplifying the development process and enabling scalable experimentation with different model architectures.
- TSNE-CUDA: GPU-accelerated t-SNE for visualizing high-dimensional learned representations. This helps uncover patterns in time series data and provides insights into the Transformer model's behavior.
- SHAP (SHapley Additive exPlanations): Explores feature importance and interprets Transformer model decisions, offering transparency in anomaly detection.

4 Goals & Expected Output

4.1 Deliverables and Evaluation Criteria

The project aims to deliver a well-documented Transformer-based anomaly detection system capable of achieving strong performance on the NAB dataset. The key deliverables include:

- A fully implemented Transformer-based anomaly detection pipeline with modular, reusable code written in PyTorch and leveraging the Hugging Face Transformers library.
- Comprehensive experimental results demonstrating the model's performance, including metrics such as precision, recall, F1-score, and AUROC. The target is to achieve an F1-score exceeding 0.85 and an AUROC above 0.90 on the NAB dataset.
- Comparative analysis of the Transformer-based approach against baseline methods and LSTM-based models to assess its advantages and limitations for anomaly detection in time series data.
- Visualization tools for interpreting the Transformer model's predictions, such as attention maps and saliency plots, to provide insights into the temporal patterns contributing to anomalous behavior.
- Detailed documentation outlining the methodology, data preprocessing steps, model architecture, training process, and evaluation results. The documentation will also include instructions for reproducing the experiments and guidelines for adapting the system to new datasets.

4.2 Potential Extensions and Future Work

The Transformer-based anomaly detection system developed in this project can serve as a foundation for further research and extensions. Some potential areas for future work include:

- Real-time anomaly detection: Adapting the Transformer model for real-time streaming data and developing efficient inference strategies to enable timely detection of anomalies in live data streams.
- Domain-specific fine-tuning: Investigating the impact of incorporating domain knowledge into the Transformer model through techniques such as transfer learning and domain-specific pre-training.
- Multivariate anomaly detection: Extending the Transformer-based approach to handle multivariate time series data, where anomalies may involve complex relationships across multiple variables.
- Unsupervised anomaly detection: Exploring the use of self-supervised learning techniques and unsupervised Transformer models for anomaly detection in scenarios where labeled anomaly data is scarce or unavailable.

Addressing these areas of future work can contribute to the development of more robust, versatile, and practically applicable anomaly detection systems based on Transformer models.

5 Timeline

- **Week 1**: Literature Review and Data Exploration
  - Research Transformer-based anomaly detection techniques.
  - Conduct exploratory data analysis on the NAB dataset to understand its characteristics and preprocessing needs.
- **Week 2**: Data Preprocessing and Model Design
  - Implement preprocessing pipelines to standardize and prepare the data.
  - Design the Transformer model, including attention mechanisms, and set up training and evaluation pipelines.
- **Week 3**: Model Implementation and Training
  - Develop the Transformer model using PyTorch and Hugging Face.
  - Train and fine-tune the model on the NAB dataset, optimizing hyperparameters for performance.
- **Week 4**: Evaluation, Interpretation, and Reporting
  - Evaluate the model on the test set, comparing metrics (e.g., F1-score, AUROC) with baseline methods.
  - Create visualizations (e.g., attention maps) for interpreting model predictions.
  - Document the entire workflow and prepare the final report, summarizing findings and insights.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, pp. 5998–6008, 2017.

[2] Q. Wen, L. Sun, F. Yang, *et al.*, "Time series data augmentation for deep learning: A survey," *arXiv preprint arXiv:2002.12478*, 2020.

[3] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng, "Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks," *International Conference on Artificial Neural Networks*, pp. 703–716, 2019.

[4] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 387–395.