Literature Survey

Anomaly Detection using Ethernet Frames

Literature Survey

Team Members

- Prateek P (1MS22CI050)
- Rishi Dave (1MS22CI054)
- Nihar Reddy (1MS22CI043)
- Nandan (1MS23CI405)

Real Time Anomaly Detection based on CNN

- This document discusses an advanced online network traffic anomaly detection system tailored for edge computing environments using Software-Defined Networks (SDN) and Convolutional Neural Networks (CNN). Key points include:

1. **Problem Context**:

- Traditional anomaly detection systems struggle in edge computing due to dynamic network configurations and high computational demands for real-time detection.

- Security threats like denial-of-service (DoS) attacks and other anomalies degrade edge network performance.

2. **Proposed Solution**:

- **SDN Framework**: Utilizes SDN for flexible and zero-configuration network management, enhancing scalability and dynamic adaptation.

- **CNN Integration**: Employs CNN to directly extract original traffic features, improving accuracy and enabling real-time packet detection and mitigation.

- **Mitigation Strategies**: Automatically blocks malicious traffic, discards attack packets, balances loads, and redirects traffic to idle servers.

3. **System Design**:

- Built using Mininet for network emulation, Open vSwitch (OVS) for the data plane, and RYU for the control plane.

- Features centralized traffic monitoring, feature extraction, and intelligent policy generation via SDN's global network view.

4. **Advantages**:

- Real-time anomaly detection with high accuracy.

- Automatic mitigation of threats.

- Enhanced security for edge clusters and reduced network latency through efficient edge computing integration.

5. **Experimental Results**:

- Demonstrated superior detection accuracy compared to traditional methods.

- Timely alerts enable proactive security measures for edge networks.

6. **Challenges Addressed**:

- Limitations of traditional methods in handling high-dimensional, dynamic data streams.

- Poor scalability and lack of intelligence in conventional systems.

7. **Applications**:

- Improved edge network reliability and throughput.

- Mitigation of evolving threats in edge and IoT-based environments.

This solution represents a significant advancement in anomaly detection, combining the strengths of SDN for network flexibility and CNN for intelligent, real-time analysis, thereby addressing modern challenges in edge computing security.

Anomaly Detection Survey

The document provided is an extensive survey on anomaly detection, structured to provide an overview of techniques, challenges, and applications. Here's a summary:

1. **Definition**:

- Anomalies are patterns that deviate from expected behavior. Common applications include fraud detection, intrusion detection, medical diagnostics, and sensor monitoring.

2. **Key Aspects**:

- **Types of Anomalies**: Point anomalies, contextual anomalies, and collective anomalies.

- **Challenges**: Defining normal behavior, evolving datasets, dynamic environments, and handling high-dimensional or noisy data.

3. **Techniques**:

- **Classification-Based**: Using supervised and semi-supervised learning to classify normal vs. anomalous patterns.

- **Clustering-Based**: Grouping data into clusters and detecting outliers.

- **Nearest-Neighbor**: Identifying anomalies based on distances or densities.

- **Statistical Methods**: Leveraging probability distributions to detect rare events.

- **Information-Theoretic and Spectral Approaches**: Capturing patterns using entropy or eigenvalues.

- **Hybrid Approaches**: Combining methods for specific scenarios.

4. **Applications**:

- Fraud detection (credit cards, insurance), network intrusion detection, medical anomaly detection (e.g., ECG data), and sensor networks.

- Specific examples include real-time anomaly detection in edge computing and robust handling of noisy datasets.

5. **Advancements**:

- Incorporation of neural networks, Bayesian models, and support vector machines to enhance detection accuracy.

- Focus on real-time detection and reducing false positives for large-scale, streaming data.

Network Traffic Anomaly Detection using Deep Learning

The document is a comprehensive survey on network traffic anomaly detection using deep learning, exploring methods to enhance network security and performance. Here's a summary:

1. **Overview**:

- Network traffic anomaly detection is vital for identifying deviations that could indicate security threats or operational issues.

- Traditional methods like thresholding and statistical models face challenges in dynamic and complex network environments.

2. **Evolution**:

- Transitioned from manual monitoring to automated thresholding, statistical methods, and later machine learning techniques.

- Machine learning introduced adaptability but required labeled data, leading to the exploration of unsupervised and semi-supervised methods.

3. **Deep Learning Advancements**:

- **Convolutional Neural Networks (CNNs)**: Adapted for analyzing traffic patterns.

- **Recurrent Neural Networks (RNNs) and LSTMs**: Effective for modeling temporal dependencies in traffic.

- **Autoencoders**: Used for reconstructing normal patterns to detect anomalies.

4. **Stages of Anomaly Detection**:

- **Data Collection & Preprocessing**: Gathering data from various sources and preparing it for analysis.

- **Baseline Modeling**: Establishing normal traffic behavior using statistical and machine learning models.

- **Real-Time Monitoring**: Comparing live traffic against the baseline to identify anomalies.

- **Characterization**: Classifying anomalies (e.g., volume, content, behavioral) and identifying root causes.

- **Response**: Automated or manual actions to mitigate threats.

- **Continuous Improvement**: Adapting models to evolving traffic patterns and threats.

5. **Datasets**:

- Includes benchmarks like KDD Cup 99, CICIDS2017, and UNSW-NB15, used for training and evaluating detection systems.

6. **Challenges**:

- Dynamic traffic patterns, high-dimensional data, computational scalability, and minimizing false positives/negatives.

7. **Emerging Trends**:

- Integration of AI and cybersecurity.

- Federated learning for privacy-preserving anomaly detection.

- Explainable AI for transparency in deep learning models.

- Edge computing for real-time detection.

- Enhancing adversarial robustness.

8. **Conclusion**:

- Deep learning has transformed anomaly detection, improving accuracy and adaptability.

- Ongoing advancements aim to address challenges and improve network security.

This survey provides a roadmap for leveraging deep learning to tackle modern cybersecurity challenges effectively.

Summary of "A Study on Network Anomaly Detection Using Fast Persistent Contrastive Divergence"

Authors:

Jaeyeong Jeong, Seongmin Park, Joonhyung Lim, Jiwon Kang, Dongil Shin, Dongkyoo Shin

Published in:

Symmetry 2024, Volume 16, Article 1220

Abstract:

The paper addresses the growing frequency and sophistication of cyberattacks as network technology evolves. It proposes a **Fast Persistent Contrastive Divergence-based Deep Belief Network (FPCD-DBN)** to enhance intrusion detection systems (IDS) by achieving high accuracy with reduced training times. The FPCD-DBN model combines the efficiency of contrastive divergence with the feature extraction capabilities of deep belief networks. The authors evaluated the model using benchmark datasets (NSL-KDD, UNSW-NB15, and CIC-IDS-2017), achieving an accuracy of **89.4%** and an F1 score of **89.7%**.

Introduction:

The introduction outlines the necessity for robust network security systems due to the rise in cyberattacks, including DDoS attacks that constitute a significant portion of internet traffic. It categorizes intrusion detection methods into three types: signature-based detection, anomaly-based detection, and stateful protocol analysis. The paper emphasizes the challenges of modeling normal behavior for anomaly detection and discusses the limitations of traditional methods like Support Vector Machines (SVMs) and Autoencoders (AEs).

Methodology:

The authors propose using **Deep Belief Networks (DBNs)** for their ability to learn complex data representations and extract robust features from incomplete data. The FPCD algorithm

enhances traditional contrastive divergence by improving layer-to-layer information transfer and utilizing fast weights for parameter updates, which speeds up training without sacrificing accuracy.

Related Works:

The paper reviews various studies on deep-learning-based network intrusion detection, highlighting different approaches such as improved conditional variational autoencoders, LSTM networks, and variational autoencoders. It notes that while these methodologies have shown promise, many studies have limitations, such as reliance on a single dataset or insufficient practical application due to the elusive nature of real-world attacks.

Results:

The performance of the proposed FPCD-DBN model was tested against multiple datasets, demonstrating its effectiveness in detecting anomalies with high accuracy and F1 scores. This suggests its potential for enhancing network security against evolving threats.

Conclusion:

The study concludes that the FPCD-DBN model is a viable solution for network anomaly detection, offering a balance between training time and detection performance. The authors advocate for further research to refine these techniques and improve their applicability in real-world scenarios.

Keywords:

Network Intrusion Detection System, Anomaly Detection, Fast Persistent Contrastive Divergence, Deep Belief Network

This summary encapsulates the essence of the research paper while highlighting its contributions to the field of network anomaly detection through innovative methodologies.

Citations:

[1] https://ppl-ai-file-upload.s3.amazonaws.com/web/direct-files/45474818/a8f58492-79c5-467d-a435-2db9c3dfab5c/symmetry-16-01220-2.pdf

Anomaly Detection in Video Sequences

Summary of "Anomaly detection in video sequences: A benchmark and computational model"

Main Points:

1. **Problem Identification:** Existing video anomaly detection databases are limited by scale and lack fine-grained annotations, containing only video-level labels without

precise time durations.

2. **Contributions:**

- A new Large-scale Anomaly Detection (LAD) database is introduced, comprising 2000 video sequences with 14 anomaly categories. It includes both video-level and frame-level annotations.

- A novel multi-task deep neural network is proposed to enhance anomaly detection by utilizing local and global spatiotemporal contextual features.

Key Findings:

- The LAD database is the largest of its kind, offering comprehensive data for training robust anomaly detection models.
- The proposed multi-task neural network outperforms existing state-of-the-art methods by leveraging detailed feature extraction at both local and global levels.

Methodology:

- **Database Construction:** The LAD database was developed with extensive anomaly categories to provide a rich training ground for models.
- **Model Architecture:** The model incorporates an Inflated 3D Convolutional (I3D) network to extract local features and a recurrent convolutional network for global feature extraction. This dual approach allows for simultaneous computation of anomaly type and score.

Implications:

- The introduction of the LAD database addresses critical gaps in current research, providing a more robust framework for anomaly detection.
- The multi-task learning approach enhances the ability to detect a wide variety of anomalies across different contexts, potentially improving applications in security and surveillance.

Conclusion:

The paper presents a significant advancement in anomaly detection through both a comprehensive database and a sophisticated model. This work sets a new benchmark for future research and applications in video sequence anomaly detection.

This summary distills the essence of the research paper, highlighting its contributions, findings, and the potential impact on the field of video anomaly detection.

Datasets

Here are some datasets that you can use for your project on **Anomaly Detection in Ethernet Frames**:

1. **CESNET-TimeSeries24**

This dataset consists of time series data of network entities' behavior, collected from the CESNET3 network over 40 weeks, involving 275,000 active IP addresses. It is designed for forecasting and anomaly detection in network traffic, providing a rich source of variability among network entities. [1]

2. **KDDCUP'99 Dataset**

Widely used for network-based anomaly detection, this dataset contains a variety of simulated intrusions and normal connections, making it suitable for evaluating various anomaly detection algorithms. [5]

3. **CIC-IDS 2017**

This dataset includes benign and malicious traffic data collected over a week, featuring different types of attacks such as DoS and DDoS, which can be useful for training models to detect anomalies in network traffic. [6]

4. **UNSW-NB15**

A dataset that contains a mix of normal and malicious traffic, including various attack types. It provides detailed features for each connection, making it suitable for anomaly detection tasks in network environments.

5. **MAWI Working Group Traffic Archive**

This dataset captures real-world traffic data from the MAWI working group and is useful for studying anomalies in live network traffic.

6. **DARPA Intrusion Detection Evaluation Dataset**

This dataset includes a variety of simulated attacks on a network and is often used to benchmark intrusion detection systems.

7. **CICIDS 2018**

Another dataset from the Canadian Institute for Cybersecurity that contains labeled data for various types of attacks and normal traffic, suitable for training anomaly detection models.

8. **ISOT Dataset**

The Information Security Office at the University of New Brunswick provides this dataset containing normal and malicious traffic data, which can be used to develop and evaluate anomaly detection algorithms.

9. **UNSW-NB15 Dataset**

Contains synthetic network traffic with labeled instances of normal and attack traffic, providing a comprehensive feature set for anomaly detection research.

10. **The UCI Machine Learning Repository: Network Intrusion Detection Data Set**

This repository includes various datasets related to network intrusion detection that can be leveraged for developing anomaly detection techniques.

These datasets offer a range of scenarios and types of network traffic that can help you effectively study and implement anomaly detection methods in Ethernet frames.

Citations:

[1] https://arxiv.org/abs/2409.18874

[2] https://paperswithcode.com/datasets?task=anomaly-detection

[3] https://paperswithcode.com/datasets?task=unsupervised-anomaly-detection

[4] https://github.com/yzhao062/anomaly-detection-resources

[5] https://www.kaggle.com/datasets/anushonkar/network-anamoly-detection

[6] https://www.kaggle.com/code/chethuhn/anomaly-detection-in-network-dataset

[7] https://machinelearningmastery.com/anomaly-detection-techniques-in-large-scale-datasets/

[8] https://www.researchgate.net/figure/Examples-of-frames-with-anomalies-in-different-public-datasets-a-and-b-are-anomalies_fig1_357973884

Other Citations:
[1] https://pmc.ncbi.nlm.nih.gov/articles/PMC11162521/
[2] https://arxiv.org/html/2409.07505v1
[3] https://www.mdpi.com/2079-9292/11/19/3138
[4] http://cucis.ece.northwestern.edu/projects/DMS/publications/AnomalyDetection.pdf
[5] https://www.mdpi.com/2073-8994/16/9/1220
[6] https://www.sciencedirect.com/science/article/pii/S1389128624004493
[7] https://www.researchgate.net/publication/381581975_A_comprehensive_Survey_on_Network_Traffic_Anomaly_Detection_using_Deep_Learning
[8] https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/ipr2.12258

Timeline

Week 1: Setup and Basic Functionality

- Install and configure ns-3 for Ethernet frame simulation.

- Develop modules for frame capture and decoding.
- Extract basic features like packet size and MAC addresses.

Week 2: ML Pipeline Development

- Preprocess captured data for ML analysis (normalize and clean features).
- Implement K-Means Clustering for initial anomaly detection.
- Test and validate clustering on simulated traffic.

Week 3: Autoencoder Integration and Advanced Features

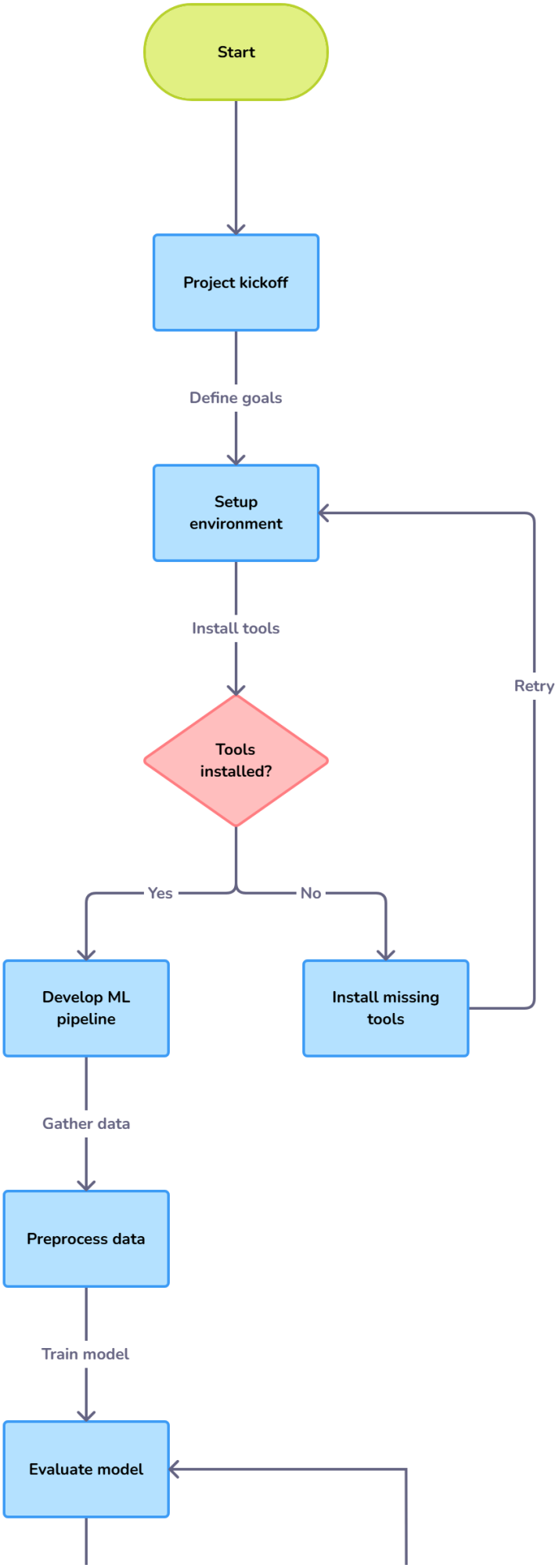
- Train an autoencoder model using normal traffic data.
- Test the autoencoder for detecting anomalous frames.
- Integrate both ML models with the main analyzer.

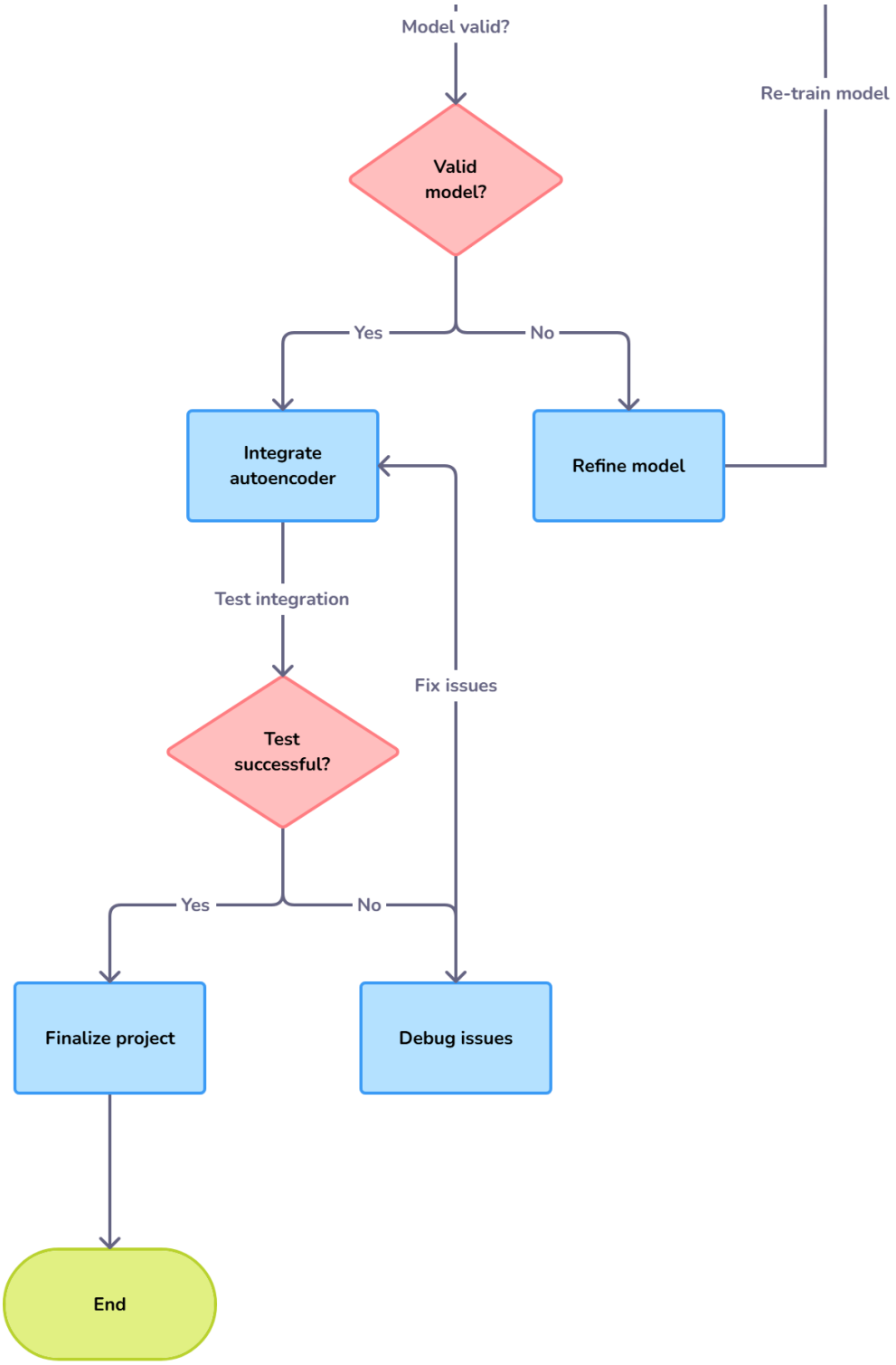
Week 4: UI Design and Final Testing

- Develop a real-time UI for anomaly visualization.
- Simulate attack scenarios (e.g., spoofing, DDoS) in ns-3 and evaluate detection performance.
- Compile a final report and prepare for project submission.

Author [Year]	Address Topic	Methodology	Simulator	Advantage	Disadvantage
Garg et al. [2023]	Ethernet MAC verification	Universal Verification (UVM)	UVM Simulator	Reusable verification components	Requires UVM expertise
IEEE [2020]	Ethernet performance (IPv4/IPv6)	Testing with jumbo frames	Custom setup	Throughput, delay, jitter analysis	IPv4/IPv6-focused, not Ethernet
Singh et al. [2023]	AI-based Ethernet traffic	Deep learning anomaly detection	TensorFlow	Automates anomaly detection	High computational cost
Mininet Research [2022]	SDN optimization for traffic	AI-based adaptive traffic configs	Mininet	Real-time analysis, easy config	Emulation, not simulation
Jaeyong Jeong [2024]	Autoencoder models for anomaly	Stochastic autoencoders	Not specified	Enhanced robustness	Suitable for Ethernet
Boyang Wan [2021]	Video anomaly detection	Pattern recognition	Not specified	Focuses on video data	Inspiration for Ethernet methods

Author [Year]	Address Topic	Methodology	Simulator	Advantage	Disadvantage
Sayantan Roy [2024]	Deep learning for network traffic	Survey of deep learning methods	Not specified	Cutting-edge techniques	Computationally heavy
Science Direct	ML-based anomaly detection	Dataset integrity analysis	Not specified	Focus on data quality	Lacks Ethernet-specific focus
Symmetry [2024]	Autoencoder anomaly detection	Variational Autoencoders	Not specified	Improved detection accuracy	Dataset-dependent
Varun et al. [2009]	Survey of anomaly detection	Classification, clustering	Not specified	Broad methodology coverage	Generic focus


Start
Project kickoff
Define goals
Setup environment
Install tools
Tools installed?
Retry
Yes
No
Develop ML pipeline
Install missing tools
Gather data
Preprocess data
Train model
Evaluate model


Model valid?
Valid model?
Re-train model
Yes
No
Integrate autoencoder
Refine model
Test integration
Fix issues
Test successful?
Yes
No
Finalize project
Debug issues
End