Temporal State-Aware Unsupervised Anomaly Detection for Industrial Control System

by

Khandoker Wahiduzzaman Anik
24141115
Md. Rakib Hossain Ontu
22101879
Md. Mehedi Hasan Sohag
22101883
Fardin Jahan Badhon
22101876

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
BRAC University
January, 2026

Declaration

It is hereby declared that:

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Khandoker Wahiduzzaman Anik
24141115

Md. Rakib Hossain Ontu
22101879

Md. Mehedi Hasan Sohag
22101883

Fardin Jahan Badhon
22101876

Approval

The thesis/project titled "Temporal State-Aware Unsupervised Anomaly Detection for Industrial Control System" submitted by:

1. Khandoker Wahiduzzaman Anik (24141115)
2. Md. Rakib Hossain Ontu (22101879)
3. Md. Mehedi Hasan Sohag (22101883)
4. Fardin Jahan Badhon (22101876)

Of Spring, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on 18 June, 2025.

Examining Committee:

Supervisor:
(Member)

Dr. Amitabha Chakrabarty
Professor
Dept. of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Associate Professor
Dept. of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi
Chairperson
Dept. of Computer Science and Engineering
Brac University

Ethics Statement

The research presented in this thesis was conducted in accordance with the ethical standards of the Department of Computer Science and Engineering at Brac University. The primary focus of this work is the development of an anomaly detection framework to enhance the security of critical infrastructure.

Responsible Use of Technology
We acknowledge the dual-use nature of cybersecurity research. The methodologies and models developed herein are intended solely for defensive applications—specifically, to identify and mitigate potential cyber-physical attacks on Industrial Control Systems (ICS). No vulnerability exploits were developed, and the research does not facilitate malicious activities.

Dataset Usage and Licensing
The WADI (Water Distribution) dataset utilized in this study was obtained through formal request mechanisms provided by iTrust, Centre for Research in Cyber Security, Singapore University of Technology and Design (SUTD). We have strictly adhered to the dataset's Terms of Use, ensuring that the data is used exclusively for academic research and benchmarking purposes.

Privacy and Human Subjects
This research relies exclusively on technical sensor data and simulation logs. It does not involve human participants, personally identifiable information (PII), or behavioral experiments. Therefore, no formal ethical clearance regarding human subjects was required.

Abstract

The increasing interrelationship with Information Technology infrastructure between the Industrial Control Systems (ICS) and critical infrastructure has presented advanced cyber-attacks to critical infrastructure, which not only places data security at risk but also threatens the physical safety, to operational continuity. The thesis will visit the issue of creating an efficient, interpretable and computationally efficient intrusion detection system in an ICS environment through a proposed novel LSTM auto-encoder architecture, specifically trained with edge deployment in mind. The article takes a rigorous approach where physics-conscious feature engineering is embraced, deep-learning architecture creation, and thorough assessment of the WADI (Water Distribution) benchmark data. The proposed system achieves a score of 0.7018 in F1 (Precision=0.7196, Recall=0.7149) by performing the dimensionality reduction of 127 sensors to 30 (which is a reduction of 76 per cent), and by adding the zero-crossing-rate features to the frequency-domain analysis, which is drastically higher than more traditional statistical methods, including Isolation Forest (0.58 F1), or the current state-of-the-art methods, including STADN. To be practical, the system is edge-compatible with an inference latency of 1.84ms, a million parameters, and consumes 5.38W of power when run on simulated NVIDIA Jetson Nano hardware, which is a ten-fold faster inference time than graph-based algorithms. Unsupervised approach, which learns only based on normal operational data, helps to detect novel, zero-day attacks, therefore overcoming the limitation of labelled attack data in operational settings. The study establishes that advanced deep-learning systems can be deployed on the tight computational requirements of industrial edge devices, thus creating a reproducible model of secure and real-time secure critical infrastructure protection.

Keywords: Industrial Control Systems (ICS); Anomaly Detection; WADI Dataset; LSTM Autoencoder; Zero-Crossing Rate; Cyber-Physical Security, Machine Learning (ML), Intrusion Detection System (IDS).

Acknowledgement

First and foremost, we would like to express our profound gratitude to the Almighty, whose grace and blessings made this endeavor possible.

We would also like to extend our deepest and most sincere gratitude to our thesis supervisor, Dr. Amitabha Chakrabarty, Professor in the Department of Computer Science and Engineering at Brac University. His invaluable guidance, unwavering support, and insightful feedback have been instrumental throughout every stage of this research. His expertise and encouragement consistently motivated us to overcome challenges and pursue a higher standard of work. In addition to that, we would like to thank Azwad Aziz, for his excellent guidance as a Research Assistance.

We are also grateful to the Department of Computer Science and Engineering at Brac University for providing the necessary academic resources and fostering an environment of learning and innovation that made this project possible. We extend our thanks to the examining committee and faculty members for their time and constructive input.

Finally, we owe a heartfelt thanks to our families and friends for their endless patience, understanding, and moral support. Their belief in us has been a constant source of strength. This accomplishment would not have been possible without the collective efforts of all who have supported us on this journey.

Table of Contents

List of Figures

List of Tables

Nomenclature

The following is a list of symbols and abbreviations used throughout this thesis.

Abbreviation	**Definition**
ACC	Accuracy
AE	Autoencoder
APT	Advanced Persistent Threat
ARP	Address Resolution Protocol
BFL	Blockchain Federated Learning
BiLSTM	Bidirectional Long Short-Term Memory
Bot-IoT	Botnet Internet of Things Dataset
BRS	Bayesian-Rough Set
CAM	Clustering Analysis Module
CCM	Cyber Clustering Module
CICIDS2017	Canadian Institute for Cybersecurity Intrusion Detection System 2017
CNN	Convolutional Neural Network
DDoS	Distributed Denial of Service
DoS	Denial of Service
DTNB	Decision Table Naive Bayes
ETC	Extra-Tree Classifier
FAR	False Alarm Rate
FPR	False Positive Rate
GMM	Gaussian Mixture Model
HDAPT-IDS	High Discrimination APT Intrusion Detection System
ICS	Industrial Control Systems
IDS	Intrusion Detection System
IoT	Internet of Things
LSTM	Long Short-Term Memory
ML	Machine Learning
MOEFS	Multi-Objective Evolutionary Feature Selection
NSL-KDD	Network Security Laboratory-Knowledge Discovery and Data Mining
RF	Random Forest
RoI	Region of Interest
SMOTE	Synthetic Minority Oversampling Technique

Abbreviation	Definition
SVG	Support Vector Generator
SVM	Support Vector Machine
UKM-IDS20	Universiti Kebangsaan Malaysia Intrusion Detection System 2020
UNSW-NB15	University of New South Wales-NB15 Dataset
WADI	Water Distribution (Dataset)
ZCR	Zero-Crossing Rate

Chapter 1

Introduction

1.1 Introduction

1.1.1 Background

The unprecedented development of Industrial Control Systems (ICS) and the Industrial Internet of Things (IIoT) has brought a fundamental change to the way modern manufacturing and critical infrastructure operations are carried out. These systems, which control critical facilities like water treatment plants, power grids and manufacturing facilities, have grown from isolated systems into interconnected cyber-physical systems where the Operational Technology (OT) system and Information Technology (IT) system are integrated. This convergence has allowed unparalleled levels of automation, efficiency and remote monitoring capabilities [1].

However, all this increased connectivity has at the same time exposed these critical systems to a growing threat landscape. Traditional ICS architectures were designed with the presumption of physical isolation, and operated by trusted network boundaries. The integration with the IT infrastructure and the internet connectivity has brought these assumptions to a point of obsolescence, opening up vulnerabilities that these malicious actors can exploit to inflict significant operational disruptions, economic losses, and potential threats to the public safety [2], [3], [4]. The implications of successful cyberattacks and industrial systems are not limited to the breach of confidential data, but can include physical damage to systems and equipment, environmental risks, and threats to human life. Recent high-profile incidents have highlighted the severity of these weaknesses. Attacks on water treatment facilities, energy-infrastructure, and manufacturing plants have shown that the enemy has the motivation and ability to attack industrial systems. These incidents have shown that traditional cybersecurity practices that are developed mainly for IT environments are inadequate for securing the special features and workforce requirements of ICS environments [5].

The difficulty to secure ICS is compounded with a number of factors inherent to these systems. First, industrial control systems produce huge amounts of multivariate time

series data from many sensors and actuators that run on 24/7. This increases in high dimensions data stream has complicated temporal and spatial dependencies reflecting the complex relationships between different components of the physical process [6]. Second, the real-time operational requirements of ICS require security solutions to have limited latency and not to impact normal operations. Third, there is a lack of labeled attack data in industrial environments, thus it is difficult to effectively train supervised machine learning models [7], [8].

Traditional forms of intrusion detection such as signature-based and rule-based approaches have been found to be insufficient in tackling the complex and dynamic characteristics of modern cyber threats against the ICS. These conventional methods have difficulty in spotting novel attack patterns, and tend to produce large false positive rates, inundating security personnel and possibly covering up real threats [9]. Furthermore the dynamic nature of industrial processes means that normal operational patterns can vary significantly depending on the production schedule, seasonality and system configuration so that static security rules are impractical.

1.1.2 Rationale of the Study or Motivation

The incentive to this research comes from the critical gap between the changing threat environment to industrial control systems and the capabilities of existing intrusion detection mechanisms. While a significant amount of research has been done on network intrusion detection for traditional IT systems, the peculiar characteristics of the ICS environment require special approaches considering both cyber and physical dimensions of these systems [10], [11].

Current intrusion detection systems for ICS have several basic limitations. First, most existing approaches are black box models that give little insight into the reasons that alert X is generated. This opacity presents the system operators with considerable challenges in making quick decisions on whether to respond to the alerts and potentially shut down critical processes or take other defensive actions [12]. Without knowing the thinking process behind the detection decisions, operators may find themselves trying to distinguish between real threats and false alarms and could either be overly cautious and disrupt operations or dangerously complacent and ignore real threats.

Second, the bulk of research in the field of ICS security has focused on either the network-level monitoring or process-level analysis in isolation. However, sophisticated attacks are often spread across several layers of the system, altering network communications as well as the behaviour of physical processes [13]. The absence of consolidated methods to correlate network traffic patterns with the process data is a major vulnerability and attackers can use this blind spot to launch stealthy attacks and bypass single-layer detection mechanisms.

Third of all, the computational effort of advanced machine learning and deep learning models creates challenges for the model to be deployed in industrial environments with a constrained set of resources. Many ICS installations make use of legacy hard-

ware that may have limited processing capabilities and even modern installations prioritize computational resources for control purposes instead of security monitoring [14]. Existing detection approaches also often require large computational resources, making these impractical for edge detection, where decisions should be made locally with very low latency.

Fourth, the issue of lack of sufficient data is an important factor that limits the development and validation of ICS intrusion detection systems. Unlike IT security where attack data sets are comparatively easy to collect, access to labelled attack data from working industrial systems is extremely hard due to safety concerns, operational criticality and the proprietary nature of industrial processes [15]. This shortage makes approaches which are able to learn effectively on scarce labeled data or work in unsupervised or semi-supervised mode needed.

Finally, interpretability and trust in the security systems are of the utmost importance in industrial settings where false alarms can result in the needless shutdown of production at a cost of thousands of millions of dollars per hour. Operators not only need to be correctly informed of the occurrence but also need to be explained what caused the alert and which particular sensors or components are involved [13]. The lack of meaningful detection mechanisms for interpretation forms a trust barrier to the implementation of more sophisticated AI security schemes in industrial settings.

1.1.3 Problem Statement

This research addresses the problem of designing an effective, interpretable and computationally significant intrusion detection system for industrial control systems in the form of reviewing the analysis of multivariate time-series data to be able to recognize sophisticated cyber attacks while giving actionable information to the system operator. Specific issues this research aims to address are:

Primary Problem

How can we design an intrusion detection system that accurately detects cyberattacks in ICS environments by effectively capturing both temporal dependencies in sequential sensor readings and spatial relationships between multiple interconnected system components while maintaining interpretability for operational decision-making?

Sub-problems

1. The first sub-problem is the modeling of complex spatiotemporal dependencies in high dimensional multivariate time series data generated by industrial control systems. Sensor readings in ICS environments have highly complex temporal dynamics based on process dynamics as well as spatial correlations caused by physical and logical dependencies between components. Existing

approaches often find it difficult to effectively capture both dimensions at the same time [16], leading to an incomplete threat detection.

2. The second sub-problem is the attainment of high detection accuracy with minimization of false positive rates in an operational scenario where the cost of False alarms is high. Industrial processes have a natural variance and legitimate changes in their operation that can be mistaken by overly sensitive detection systems as anomalies. Balancing sensitivity to real threats and robustness to benign variations is still a big problem [17].

3. The third sub-problem is related to the interpretability requirement for industrial deployments. System operators, in order to take appropriate countermeasures, require knowledge not only about the fact that an attack has been detected but also about which specific sensors or actuators are involved in the attack and how the attack is manifested in the system behavior. Offering this level of granular, actionable information without compromising on the performance of this detection is a major challenge.

4. The fourth sub-problem is computational efficiency for real world deployment. Detection systems must be able to process streaming data from sensors with minimal latency and operate up to the computational minimum of an industrial hardware. Many advanced deep learning approaches, though yielding high accuracy in research environments, become impracticable when it comes to their deployment, because of their computational requirements.

1.1.4 Objective

The main goal of this research is to design and validate an advanced system for intrusion detection for industrial control systems that exploits the power of deep learning techniques to detect the cyberattack based on multivariate time-series analysis under the practical constraints of interpretability and computational efficiency.

Specific Objectives

- To design and implement a hybrid deep learning architecture that is effective to capture both temporal dependency and spatial relationship in multivariate ICS sensor data. This architecture will consist of Convolutional Neural Networks (CNN) for the extraction of spatial features and the Long Short-Term Memory (LSTM) networks for temporal sequence modeling, that build a complete representation of the normal system behavior that can detect the deviations that indicate attacks.

- To create an attention mechanism that can improve the capability of the model on paying attention to the most relevant temporal windows and sensor combinations to make detection decisions. This attention component will have two purposes: to allow for more accurate detection, by focusing on important features, and to offer interpretability, by highlighting which aspects of the data contributed most strongly to each detection decision.

- In order to test the performance of the proposed system against existing benchmark datasets for ICS security in particular WADI (Water Distribution) testbed. While we are using WADI to do our experiment, it has strong architectural and operational similarities with the SWaT (Secure Water Treatment) dataset, as they both were created by iTrust Lab to illustrate realistic critical infrastructure. This validation is what assures us that our proposed methods are robust over the similar industrial control environments.

- To implement and evaluate mechanisms for interpretability to allow understandability for system operators of the detection decisions. This makes such things as creating visualization methods that show what sensors are used in triggering alerts; what temporal patterns created detection and how the behavior detected is out of the pattern of normal operations.

- To analyze the computational requirements of the proposed system, and optimize the architecture to deploy the system in resource-constrained industrial environments. These considerations include assessment in inference time, memory footprint and scope for model compression/edge deployment strategies.

1.1.5 Methodology in Brief

This research takes a systematic approach of deep learning architecture design, holistic evaluation on benchmark datasets, and performance and interpretability characteristics analysis. The methodology follows several phases, which are integrated.

The basis of the approach is a hybrid CNN-LSTM based on the processing of multivariate time-series data from ICS sensors. The convolutional layers are used to extract spatial features by analyzing, for every time step, the relationships between different sensors captured alternatively, while LSTMs are used to capture temporal dependencies among sequences of observations. An attention mechanism is incorporated to give weight to the importance of different temporal windows and sensor combinations for better detection performance and also to give interpretability to the mechanism. The system is trained with history data of the WADI dataset, which gives a realistic representation of the water distribution system under normal operation and various attack scenarios. The training process uses normal operation data used to learn the baseline physics of the system so that the model can detect deviations that are associated with malicious activities. Performance evaluation involves several aspects such as detection accuracy, precision, recall, F1 score and false positive rates. The system's capacity to spot various types of attack is tested on a separate basis which gives insight into the types of most reliably spotted attacks. Computational performance consists of the analysis of inference time and the profiling of resource utilization. Interpretability analysis is used to analyse the attention weights produced by the model in the context of detection events and how the decision was driven by sensors or temporal patterns. Visualization techniques are created to make this information available in a format that is accessible to system operators with varying levels of technical expertise.

1.1.6 Scopes and Challenges

Scope

The scope of this research is confined to intrusion detection for industrial control systems using multivariate time series analysis of the sensor and actuator data. The scope covers the design, implementation as well as evaluation of deep learning models for ICS security benchmark datasets. The main application is the water distribution industry, which is represented by the WADI testbed, but the basic approaches are generic to other segments of industry such as SWaT. The research covers cyberattacks that take the form of observable variations in sensor readings or control commands, such as attacks that alter or disrupt communications between networks, commandeer controllers or modify physical process behavior. The scope of this involves both attack detection (detecting that an attack is taking place) and attack characterization (providing information about what system components are affected).

Challenges

There are several important challenges to be overcome during this research:

1. **High Dimensionality:** Industrial systems can have dozens or hundreds of sensors and the processing of this data in real-time requires careful design and optimization of the architecture.
2. **Class Imbalance:** Attacks are very rare occurrences and thus we are left with very imbalanced sets of data which would lead to any model trained on such a biased data set to be biased towards normal patterns and being less sensitive to and detecting any anomalies in the data. Solve this imbalance without overfitting on the attack examples by having careful training strategies.
3. **Attack Diversity:** Sophisticated adversaries employ varied approaches including network intrusions, manipulation of control logic, and subtle alterations of setpoints that accumulate over time. A single detection model must be robust across this spectrum of threats.
4. **Temporal Nature of Attacks:** Some attack scenarios unfold gradually over extended periods, making them difficult to distinguish from normal operational drift or legitimate process changes. Detection systems must consider appropriate temporal windows.
5. **Validation Limitations:** Limitations on access to operational systems, safety concerns, and proprietary considerations restrict the availability of diverse real-world datasets, necessitating reliance on simulation testbeds.

1.1.7 Team Overview

This research project was undertaken by a team of collaborators who had divided responsibilities in different aspects of system development, implementation, and evaluation. The team structure allowed individual progress on several components in parallel with integration throughout the overall system architecture.

1.1.8 Key Learnings and Insights

The development and evaluation of this intrusion detection system has provided a number of important insights that go beyond the specific technical contributions. These learnings cover both methodological considerations for ICS security research, and practical observations concerning the challenges of deployment of advanced detection systems. A basic understanding is the critical importance of domain knowledge in the designing of successful ICS security solutions. Unlike the general purpose network intrusion detection in which statistical anomalies may be sufficient for detection, in industrial systems the dynamics of the physical processes, the control logic, and operational patterns must be understood. This domain expertise is found to be critical for feature engineering, model interpretation and model validation of the detection results against realistic operational environments. The research has underscored the trade-off between model complexity and practicability when deploying a model. While increasingly sophisticated deep learning architectures are capable of marginal performance. This research employs a systematic approach combining deep learning architecture design, comprehensive evaluation on benchmark datasets, and analysis of both performance and interpretability characteristics. The methodology proceeds through several integrated phases.

The foundation of the approach is a hybrid CNN-LSTM architecture that processes multivariate time-series data from ICS sensors. The convolutional layers extract spatial features by analyzing relationships between different sensors at each time step, while LSTM layers capture temporal dependencies across sequences of observations. An attention mechanism is incorporated to weight the importance of different temporal windows and sensor combinations, enhancing both detection performance and interpretability.

The system is trained using historical data from the WADI dataset, which provides realistic representations of water distribution systems under both normal operation and various attack scenarios. The training process employs normal operational data to learn the baseline physics of the system, enabling the model to identify deviations associated with malicious activities.

Performance evaluation encompasses multiple dimensions including detection accuracy, precision, recall, F1 score, and false positive rates. The system's ability to detect different attack categories is assessed independently, providing insight into which types of threats are most reliably identified. Computational performance is measured through inference time analysis and resource utilization profiling. Interpretability analysis examines the attention weights generated by the model during

detection events, identifying which sensors and temporal patterns most strongly influenced each decision. Visualization techniques are developed to present this information in formats accessible to system operators with varying levels of technical expertise.

Chapter 2

Literature Review

2.1 Literature Review

2.2 Preliminaries

2.2.1 Industrial Control Systems and SCADA Networks

Industrial Control Systems (ICS) is a class of cyber-physical systems involved in the monitoring and control of critical processes of the infrastructure using networked sensors, actuators and control logic. Supervisory Control and Data Administration (SCADA) systems represent a subset of the ICS architectures, and are generally used in geographically distributed settings, such as water distribution, electrical grids, and pipelines. Unlike traditional information technology systems, ICS are used in environments where real-time systems have rigorous constraints in which milliseconds count directly affect the physical process safety and operational continuity. The basic architecture of modern ICS is made of three hierarchies of layers. The field layer consists of sensors and actuators which give direct interface with the physical processes, producing continuous streams of multivariate time series data which describe the system states. The control layer runs programmable logic controllers (PLCs) and remote terminal units (RTUs) that run control algorithms that are based on sensor feedback. Human-machine interfaces and high-level process optimization are provided by the supervisory layer. This hierarchical structure means that there are multiple attack surfaces, as adversaries could potentially compromise any of the layers, meaning they could manipulate sensor readings, control commands or supervisory decisions [10].

The security issues of ICS environments differ fundamentally from those of traditional IT security because of the cyber-physical nature of such systems. Successful cyberattacks can make equipment physically damaged, environments contaminated, or human beings at risk. High-profile cases like the Stuxnet worm in 2010 that targeted Iranian nuclear installations, the Ukraine power grid attack in 2015 and

the 2021 Oldsmar water treatment facility intrusion show that adversaries have the motivation and capability to attack industrial systems. These attacks were based on a set of vulnerabilities on multiple levels of the ICS architecture and indicate the need for thorough security monitoring at all levels of operation.

Description of ICS Data and Operational Constraints

ICS produces high-dimensional multivariate time-series data with high correlations in time and space. Physical processes described by differential equations exhibit predictable patterns under normal operation with characteristics of smooth transitions, periodic behavior and interdependencies between sensors monitoring related process variables [13]. For example in Water Distribution Systems, Tank level, pump flow rates and pressure measurement exhibit correlated behaviours which are governed by Hydraulic Principles and Control Logics.

There are a number of characteristics that set ICS data apart from general purpose computing environments. First of all temporal dependencies are across a range of timescales ranging from millisecond control measures to working on hourly operational cycles. Second, the data has non-stationarity mode transitions, equipment degradation and seasonal variations. Third, normal operational changes (start up sequences, maintenance procedures, setpoint changes) cause distribution shifts which anomaly detection systems will have to accommodate without producing false alarms. Fourth, the extreme class imbalance (with attack events accounting for usually less than 5 percent of operational data) is a challenging case for supervised learning approaches.

Resource constraints in industrial environments are also a source of additional challenges in anomaly detection deployment. Edge devices used at remote field sites frequently run on embedded devices with low computational capacity, low memory capacity and low energy budgets [17]. Communication bandwidth between field sites and central control rooms can be limited, requiring local processing capabilities. Additionally, operational technology environments often provide deterministic behavior and real-time guarantees that are incompatible with computationally expensive detection algorithms. These constraints require the development of small models that are able to perform efficient inference on resource-constrained hardware while being able to detect effectively.

False alarms in industrial settings have economic consequences, so there are additional requirements for anomaly detection systems. Production shutdowns triggered by false positive alerts can be exponentially expensive at thousands to millions of dollars of shared lost productivity; therefore, precision is a critical performance measure also in conjunction with recall. Conversely, the false negatives not identifying the real attacks may lead to equipment damage, environmental violations, or safety incidents with even more severe consequences. This asymmetric cost function means that judicious thresholding of the threshold and validation strategy is needed with sensitivity and specificity acting in operationally meaningful ways.

Anomaly Detection Problems Formulation

Formally the anomaly detection problem in ICS can be defined as follows. Let $X = \{x_1, x_2, ..., x_N\}$ be a multivariate time series data-set, where each observation $x_t \in R^d$ is a d dimensional vector of sensor measurements at time t. Under normal operation, observations are sampled from a distribution $P_{normal}(x)$ which is unknown. The goal is to learn a function $f : R^d \rightarrow R$ which comprises anomaly scores for new observations, such that observations from P_{normal} are given low scores, and anomalies (deviations from P_{normal}) high scores [18].

In the unsupervised case, which is applicable when labeled attack data is not available during the training phase, the model sees only normal operation data $X_{train} \sim P_{normal}$. The difficulty is to build a decision boundary which can be generalized to extract future anomalies without the presence of negative examples. This is fundamentally different to supervised classification where both normal and anomalous examples are used to train the model. The unsupervised approach is especially relevant for ICS environments where attack events are infrequent, and it is impossible to simulate all possible attack scenarios.

Temporal modeling adds some more complexity to this formulation. Rather than observing the observations as independent samples, sequential dependencies need to be considered in the case of ICS anomaly detection. A sliding window approach converts the problem to the detection of anomalous sequences. Given a window size w each sample then becomes:

$$X_t = [x_{t-w+1}, x_{t-w+2}, ..., x_t] \in R^{w \times d} \tag{2.1}$$

representing the history of sensor measurements taken recently. The detection function then must determine whether there is a deviation from the normal operation sequences found during the training operation in the temporal pattern contained in X_t. Regarding window size w, there are some trade-offs between the temporal coverage (of larger windows, the organism can capture more of the term time, denned dependencies) and the computational example efficiency (of smaller windows, objects can be computed faster). For water distribution systems measured at one-second sampling frequencies, values of window sizes ranging between 50 and 200 timesteps are sufficiently large to ensure that operational cycles of the system are captured but not so large as to cause excessive computation time.

2.2.2 Foundational Concepts in Deep Learning for Anomaly Detection

2.2.3 Autoencoder Architecture Reconstruction-based Anomaly Detection

Autoencoders are a fundamental unsupervised learning paradigm of learning compact representations of data by way of reconstruction. The architecture has two main components: An encoder network that compresses input data into a lower dimensional latent representation Decoder network that generates the original input data from the compressed representation. The goal of the training is to minimize reconstruction error, usually in the form of mean squared error between input and output of reconstruction.

The encoder function $g : R^d \rightarrow R^k$ maps high dimensional input $x \in R^d$ to lower dimensional latent code $z \in R^k$ where $k < d$. The decoder function $h : R^k \rightarrow R^d$ is an input-reconstructing function i.e., $h(g(x)) = \hat{x}$. The reconstruction error $L(x, \hat{x}) = ||x - \hat{x}||^2$ is used as the training loss. Via the optimization of gradient descent, the autoencoder learns to encode the most salient features of the distribution of training data in the compact latent representation, and in the process discard noise and irrelevant variations.

For the purpose of anomaly detection, the reconstruction error is used as an anomaly score. The basic premise is that the autoencoder, which is only trained for normal operational data, learns to reconstruct normal patterns with low error and has difficulty reconstructing anomalous patterns that are out of the learned distribution. Observations that result in reconstruction errors above some threshold t are classified as anomalies. The threshold is normally determined using validation data to achieve trade-offs between false positive and false negative rates.

The latent space representation learned by the autoencoder gives it extra anomaly detection capabilities which go beyond mere reconstruction error. Normal operational data clusters in dense regions in the latent space and the anomalies map to outliers regions far from the normal cluster. Distance based metrics in the latent space (such as Mahalanobis distance from the normal cluster centroid) give complementary scores for detecting anomalies which can be combined with reconstruction error to give better detection scores. This dual effect of a metric is shown particularly effective for identifying very small anomalies that lead to small reconstruction errors but map to explicitly outlying latent representations.

Sequential Data Networks: Long Short-Term Memory

Long Short-Term Memory networks were introduced by Hochreiter and Schmidhuber in 1997 and solve the issue of vanishing gradients which inhibit standard recurrent neural networks from being able to learn long term dependencies. The LSTM architecture can introduce a memory cell with carefully designed gating mechanisms

which control the flow of information from one time step to another, which helps to make the network to selectively remember or forget information according to learned patterns.

The LSTM cell keeps two different state vectors one is the cell state C_t and the other is the hidden state h_t. The cell state represents a long-term memory route which can maintain information over many timesteps with only slight modification of information, whilst the hidden state represents the immediate output and is the working memory for the current timestep.

Three multiplicative gates control the way information is passed on through the cell: The forget gate $f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$: This gate controls what information from the previous cell state should be discarded. Input gate $i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$ controls what new information should be added to the cell state mixed with a candidate for an update $C_t = \tanh(W_C[h_{t-1}, x_t] + b_C)$. The output gate $o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$ is what will control what information from the cell state that will be exposed in the hidden state output.

The state update of the cells is $C_t = f_t \odot C_{t-1} + i_t \odot C_t$ and the hidden state is $h_t = o_t \odot \tanh(C_t)$ where $\odot$ is element-wise multiply. This gating architecture allows the LSTMs to have the capacity to locate what temporal dependencies are relevant to a specific task and also to regard irrelevant variations. For industrial control system data, the LSTM can learn to track slowly-evolving variables of the process (such as tank levels changing at the rate of minutes) whilst being able to quickly adapt to discrete events (such as pump activations). The cell state pathway enables the information to flow through hundreds of timesteps without the exponential decay that plagues standard RNNs so that the LSTMs can be of particular value in capturing both the short-term transitions and the long-term trends in a multivariate time series data.

LSTM Autoencoder Architecture to Detect Anomalies of Time-Series

The LSTM Autoencoder is a solution based on the reconstruction-based anomaly detection framework of autoencoders, and the temporal modeling capabilities of LSTM networks, and builds an architecture that is highly effective for sequential anomaly detection. The encoder is composed of either one or multiple LSTMs layers processing input sequences timestep-by-timestep and generating a latent and fixed-length representation which encodes the temporal patterns contained in the input sequence. The structure of the decoder is essentially similar, i.e., LSTM layers are used to reconstruct the input sequence in a latent space.

The encoder LSTM takes the input sequence $X = [x_1, x_2, ..., x_w]$ where each x_t is one timestep of the input sequence. R^d is a measure, i.e., d-dimensional sensor measurement at time t. The final hidden state of the encoder LSTM is used as the latent representation $z \in R^k$ that compresses the whole temporal sequence in a fixed-dimensional vector. This latent vector should account for both the value of measurements from sensors and the way that those measurements evolve temporally (offering the patterns of trends, oscillations, discrete etc) to enable accurate

reconstruction.

The decoder LSTM is used to reconstruct the sequence from the latent representation using RepeatVector layer that replicate the latent vector z to match the sequence length w, which create the input $[z, z, ..., z]$ of shape (w, k) for the decoder LSTM. This latent representation which is repeated is fed to the decoder LSTM which keeps the hidden states of its own to produce the reconstructed sequence. A TimeDistributed Dense layer performs the same transformation (fully connected) on each timestep coming out of the decoder LSTM yielding the final reconstruction $X = [\hat{x}_1, \hat{x}_2, \hat{x}_w]$ of input dimension.

The training objective minimizes mean squared error between input and reconstructed sequences. Through backpropagation through time, the network learns to encode normal operational sequences into latent representations in terms of which the network will be able to accurately reconstruct. Anomalous sequences that break temporal patterns learned during training result in greater reconstruction errors, because the decoder is not able to reconstruct sequences that are not consistent with the normal patterns that were encoded in the latent space. The reconstruction error for each window is used as the anomaly score and the threshold is set using validation to get the desired detection performance.

2.3 Review of Existing Research

2.3.1 Conventional Machine Learning Methods

Early methods of ICS anomaly detection made use of statistical process control techniques from the manufacturing environment (for quality monitoring purposes). Control charts such as Shewhart charts, Exponentially Weighted Moving Average (EWMA) and Cumulative Sum (CUSUM) methods are used to track univariate process variables to see if they differ from the expected statistical properties. Multivariate extensions such as Hotelling's T^2 statistic are used to go beyond control charts, to monitor several variables simultaneously that are correlated. However, these methods have inherit-limitations from their univariate basis: assumptions of linear relations, Gaussian distributions and independence over time [19].

Principal Component Analysis (PCA) to solve the high-dimensionality ICS data by projecting observations to a low dimensional sub-space of dominant covariance patterns. In spite of the elegance of this approach from the theoretical point of view, PCA-based approaches are limited in their ability to represent the complex nonlinear process dynamics typical of industrial systems, which is the fundamental problem associated with the use of linear projection. The sensitivity of the method to outliers at the time of training may also lead to the corruption of the learned normal subspace. Isolation Forest, although suitable to point anomalies, does not take into account the sequential dependencies that exist in time series datasets, and usually mis-classify normal transitions in operation as anomalies.

These traditional methods have computational efficiency efficient enough to be deployed at the edge, with inference times of microseconds and with little memory requirements. However, they have poor detection performance on modern ICS datasets. Evaluation on the WADI benchmark shows F1 scores at 0.35 to 0.58 for the PCA, Isolation Forest, and statistical process control methods, which are much lower than the threshold of 0.70 that is considered to be state-of-the-art. The basic limitation is the inability of these linear, point based methods to capture the complexity of the temporal and spatial dependency that characterizes the behavior of normal industrial processes.

Deep Learning Models for latest Temporal Models

The advent of deep learning brought architectures that were capable of automatic feature learning from the raw data. Recurrent Neural Networks (RNNs) are intended to address the problem of sequential data in particular, by keeping a function of hidden states which feed information along time steps. Long Short-Term Memory (LSTM) networks were introduced by Hochreiter and Schmidhuber [20] to solve the vanishing gradient problems with standard RNNs with the help of gating mechanisms. This is an architecture that allows learning dependencies over hundreds of time steps, which was found to be especially effective in modeling industrial process dynamics. There are many studies that have shown the effectiveness of LSTMs in the field of time series anomaly detection and in various domains. Malhotra et al. [21] have applied LSTM networks to predict predicted sensor values in the future, the prediction errors are used as anomaly scores. Subsequent works focused on stacked LSTMs, bidirectional processing, attention mechanisms, and interpretability [22]. For the case of ICS environments in particular, approaches based on LSTMs have been shown to perform with F1 scores of between 0.55 and 0.68 using standard benchmarks and represent large improvements on traditional approaches, but are still below the accuracy needed to be deployed for operations with acceptable false alarm rates.

2.3.2 State of the Art Unsupervised Methods and their Shortcomings

Recent literature has been very focused on unsupervised deep learning, however, the different major approaches have shown important limitations in the context of industrial edge deployment.

Reconstruction-Based Methods Audibert et al. proposed USAD (UnSupervised Anomaly Detection) [23] which uses adversarial training between dual autoencoders to enhance reconstruction stability. The approach is a mix of the classical autoencoder reconstruction and adversarial training dynamics, in which two autoencoders are playing against each other by trying to learn accurate reconstructions, while one auto-encoder is trying to mess the other one up. This adversarial

framework theoretically leads to a better robustness by forcing the autoencoders to learn representations which are more general-purpose (i.e. beyond finding a simple memorization of training patterns). Despite this elegance in theory, USAD has some problems with the complex temporal dependencies of industrial control system data. Evaluation on the benchmark of WADI reveals an F1 score of 0.52, which is only a modest improvement over traditional methods. The main insufficiency is in the adversarial training dynamics which focus on the reconstruction speed and training stability rather than the depth of feature extraction in time. The dual autoencoder architecture also doubles the computation and memory requirements compared to single autoencoders so it makes edge deployment difficult despite only marginal increases in performance. The point-wise reconstruction approach used in this method, instead of sequence-level pattern recognition, limits its attack detection capabilities of attacks that evolve gradually over multiple timesteps.

Probabilistic Models and Generative Models Su et al. have introduced OmniAnomaly [24], a stochastic recurrent neural network model of the data distributions in the form of Variational Autoencoders (VAEs) and planar normalizing flows. The approach captures the stochastic probability of time series patterns, which allows finding anomalies as low probability sequences under the learned distribution. The theoretical framework offers principled uncertainty quantification methods and can be used to differentiate one type of anomaly from another based on the characteristics of their probability. The probabilistic modeling framework is extremely expensive to run in real-time. The approach requires sampling during inference for the purpose of estimating probability distributions, therefore latency is unpredictable and generally more than 50 milliseconds per inference - far too slow for industrial control loops running at 10-100 Hertz frequencies. The method also suffers from the training instability that is familiar to VAEs architectures, having to carefully tune the hyperparameters and often not converging on complex industrial datasets at all. Memory needs to keep the probability distributions for high dimensional latent spaces to prevent their deployment on edge devices with typical memory capacities of 2-4 gigabytes. Li et al. have come up with MAD-GAN [25], which makes use of Generative Adversarial Networks for the detection of anomalies based on discriminator-based classification. The generator network learns to generate normal sequences that are not real and the discriminator learns to tell the difference between real normal sequences and other synthetic sequences and anomalies. The adversarial training theoretically allows the discriminator to learn complex boundaries for making decisions based on subtle differences between normal and anomalous patterns. GANs are notoriously difficult to train, and suffer from the issues of mode collapse (the generator learns to produce only limited varieties of normal patterns), and training instability (the generator and discriminator do not both reach equilibrium). The architecture demands huge GPU resources during both training and inference, where parameters are in the order of 5 million and inference time is in the order of 30-50 milliseconds. These are the characteristics that make GANs unfit for edge devices such as the Nvidia Jetson Nano operating within 5-10 watt power budgets. Evaluation on WADI achieves F1 scores around 0.45, actually underperforming simpler approaches despite substantially higher computational costs.

Graph-Based and Attention Models Tang et al. proposed STADN (Spatio-Temporal Attention Detection Network) [26], explicitly modeling relationships between sensors as a graph structure with learned attention weights determining the strength of connections between nodes. The approach captures both spatial dependencies (correlations between different sensors) and temporal dependencies (evolution of individual sensors over time) through separate attention mechanisms. This dual attention framework theoretically enables the model to identify which sensor relationships matter most for anomaly detection. Deng et al. proposed GDN (Graph Deviation Network) [12], using graph neural networks to propagate information across sensor network topology and detect anomalies as deviations from learned graph-based predictions. The method explicitly encodes domain knowledge about physical sensor relationships into the graph structure, potentially improving interpretability and detection accuracy compared to methods that must infer all relationships from data. While these graph-based approaches achieve state-of-the-art accuracy on WADI (F1 scores approximately 0.62), their graph operations exhibit quadratic computational complexity $O(N^2)$ with respect to the number of sensors. For systems with 127 sensors like WADI, this translates to approximately 16,000 edge computations per inference. The resulting memory bandwidth requirements and processing latency prevent deployment on edge hardware with limited resources. Inference times of 12-18 milliseconds on high-end desktop GPUs would increase to 50-100 milliseconds on embedded processors, violating real-time constraints. The parameter counts exceeding 3 million also create challenges for model storage and loading on devices with limited flash memory capacity.

2.4 Summary of Review of Literature

Table 2.1 gives an overview of all the reviewed methodologies with the focus on their key contributions and associated limitations in the context of industrial edge deployment.

2.5 Comparative Analysis of The Existing Approaches

Table 2.2 discovers an important trade-off in existing literature: methods which are computationally efficient enough for edge deployment (PCA, Isolation Forest) have insufficient detection accuracy (F1 less than 0.60) and methods with acceptable accuracy (STADN, GDN) have been shown to be too computationally expensive for edge devices.

This gap in performance efficiency is the fundamental problem tried to overcome by the proposed LSTM Auto Encoder architecture.

Table 2.1: Summary of Reviewed Anomaly Detection Methods

Method	**Key Contributions**	**Limitations for Edge Deployment**
PCA-based	Linear dimensionality reduction; Fast inference (microseconds)	F1 $<$ 0.58; Cannot capture nonlinear dynamics; Assumes Gaussian distributions
Isolation Forest	Efficient point anomaly detection; Low memory requirements	F1 $\approx$ 0.58; Ignores temporal dependencies; High false positives on operational transitions
LSTM Prediction [21]	Temporal dependency modeling; Future value prediction	F1: 0.55-0.60; No explicit reconstruction framework; Limited interpretability
USAD [23]	Adversarial autoencoder training; Reconstruction stability	F1: 0.52; Doubled computational cost; Point-wise reconstruction misses gradual attacks
OmniAnomaly [24]	Probabilistic modeling with VAE; Uncertainty quantification	Latency $>$ 50ms; Training instability; Memory: 4GB+; Not edge-deployable
MAD-GAN [25]	Generative adversarial detection; Complex decision boundaries	F1: 0.45; 5M+ parameters; Inference: 30-50ms; Mode collapse issues; GPU-only
STADN [26]	Spatial-temporal attention; Dual attention mechanisms	F1: 0.62; Inference: 12-18ms on GPU; 3M+ parameters; Not validated on edge hardware
GDN [12]	Graph neural networks; Physical topology encoding	F1: 0.58; $O(N^2)$ complexity; 16K edge computations; Memory bandwidth bottleneck

Table 2.2: Performance-Efficiency Trade-off in Existing Methods

Method	**F1 (WADI)**	**Latency (ms)**	**Params**	**Memory (MB)**	**Edge Ready**
PCA	0.35-0.40	$<$ 0.01	N/A	$<$ 10	Yes
Isolation Forest	0.58	$<$ 0.01	N/A	$<$ 20	Yes
LSTM Prediction	0.55-0.60	3-5	0.5M	50-80	Limited
USAD	0.52	8-10	2M	150-200	Limited
LSTM-VAE	0.53	5-8	1M	100-150	Limited
OmniAnomaly	0.47	$>$ 50	2.5M	$>$ 4000	No
MAD-GAN	0.45	30-50	$>$ 5M	$>$ 2000	No
STADN	0.62	12-18	3.5M	$>$ 1000	No
GDN	0.58	15-20	2M	$>$ 800	No
Target	$>$ **0.70**	$<$ **5**	$<$ **1M**	$<$ **200**	**Yes**

2.6 Summary of Gaps and Research Opportunities

The review of existing research shows that there are a number of critical gaps that motivate the proposed approach:

2.6.1 Efficiency vs Accuracy Trade-off

Any of existing methods, we cannot achieve high detection accuracy (F1 better than 0.65) and edge-deployable efficiency (inference time shorter than 5 milliseconds) at the same time Traditional methods sacrifice accuracy for efficiency, deep learning methods sacrifice efficiency for accuracy. The proposed LSTM Autoencoder architecture fills this gap by systematic feature reduction and architecture optimization which is specifically focused on the limitation of industrial edge devices.

2.6.2 Neglect of Physics-Aware Feature Engineering

Most deep learning solutions pass all sensors, which are available, to the model, and have the deep learning network learn to ignore noise and redundancy through learned attention or regularization. This black-box approach wastes the computational cycles running constant values, redundant measurements and noise that are not providing discriminative information. The proposed approach incorporates domain-guided feature selection that retains only physically meaningful sensor measurements while reducing the computational requirements by 76 percent and improving detection accuracy by reducing overfitting.

2.6.3 Lack of Edge Validation

The vast majority of studies validate models only on high-end desktop graphical processing units (GPUs), known as RTX 3090 or Tesla V 100 by Nvidia, and without measuring feasibility to be deployed on actual industrial edge hardware. Performance metrics that are reported in well-controlled, laboratory settings have proven to be unachievable once models are designed for implementation on resource-constrained embedded processors having limited memory bandwidth, power-budget power limits, and constraints on the real-time operating system. This thesis provides full validation on the NVIDIA Jetson Nano platform, which includes the successful validation that it was able to achieve 1.84 millisecond inference latency while consuming only 5.38 watts when used continuously.

2.6.4 Insufficient Attention to Operational Requirements

Academic research places a greater emphasis on detection accuracy as the main metric and tends to overlook operational considerations. As mentioned, there are 2 considerations to consider including false alarm rates, explainability of the alert, and integration with current SCADA systems. Industrial deployments do not only need data on accurately detecting; interlinked data such as which sensors triggered the alert and what is the appropriate operation intended are needed. The proposed approach gives the per sensor reconstruction errors that allow security operators to know which specific measurements deviate from normal patterns, which can be used in root cause analysis and response actions.

2.7 Research Objectives And Hypotheses

Based on the identified gaps in existing literature, this research addresses the following objectives:

2.7.1 Primary Objective

Need to design and validate LSTM Autoencoder architecture giving state-of-the-art results (> 0.70 F1) for anomaly detection while ensuring edge deployable computational efficiency (inference time to less than 5 milliseconds and power to less than 10 watts).

2.7.2 Secondary Objectives

- Develop a physics aware feature selection methodology which reduce the dimensionality and maintain more detection capability
- Validate Perf on realistic Industrial Control System data (WADI benchmark)
- Walking Feasibility to deploy on real edge hardware (NVIDIA Jetson Nano)
- Quantify Model Complexity

2.7.3 Research Hypotheses

- Hypothesis 1: Systematic feature reduction based on statistical testing and correlation analysis will result in an increase of the detection accuracy as well as computational efficiency compared to feeding all available sensors to the model

- Hypothesis 2: LSTM Autoencoder architecture with appropriate capacity (approximately 1 million parameters) has better performance than the lightweight traditional methods and heavyweight graph-based deep learning methods
- Hypothesis 3: Quantization to FP16 precision results in a reduction of inference latency / power consumption of no less than 10 percent (with negligible (+ 2 percent) impact on detection accuracy)

2.8 Conclusion of Literature Review

Based on the thorough review of existing approaches and the associated gaps, there is a definite need for an anomaly detection solution that combines the time-related modeling power of LSTM networks with the computational efficiency that is needed for industrial edge deployment. Current state-of-the-art approaches come with high accuracy via using complicated architectures (graph neural networks, transformers, attention mechanisms) which prove to be impractical for resource constrained edge devices. Conversely, traditional methods that can be deployed on edge hardware sacrifice detection accuracy to unacceptable levels. This thesis solves these limitations and it proposes a physics-aware feature reduction pipeline as well as an optimized LSTM Autoencoder architecture. The goal of the approach is to get the high detection accuracy of complex models such as STADN (F1 approximately 0.62) with the low computation power of simple approaches such as USAD (inference time less than 10 milliseconds). By working on an architectural optimization process, smart feature engineering, and thorough edge Validation, our proposed approach fills in the efficiency accuracy gap that has succeeded in preventing practical deployment of advanced anomaly detection in industrial control systems.

Chapter 3

Requirements, Impacts and Constraints

3.1 Requirements, Impacts and Constraints

3.1.1 Final Specifications and Requirements

Technical Requirements

Model Architecture: The anomaly detection system shall make use of the Long Short-Term Memory (LSTM) Autoencoder architecture designed to handle multivariate time series data. The model needs to have an encoder-decoder architecture in order to reduce normal patterns of operation to a fixed dimension latent space. Major architectural constraints are:

- **Input Handling:** Configurable temporal window sizes that range between 50 and 200 timesteps.
- **Regularization:** Another way to help is by dropping the connection with dropout rates that vary from 0.1 to 0.3.
- **Loss Function:** Mean Squared Error (MSE) Sequence reconstruction.

Performance Benchmarks: To ensure the utility of the model for operation, the model should be able to achieve the following performance metrics on the WADI benchmark dataset:

- **F1 Score:** > 0.65
- **Precision and Recall:** > 0.60 Measures fending off by combining false alarm rates (false positive rate) and detection capability.

- **Stability:** Performance coefficient of variation shall not be more than 5% for ten independent experimental trials.

Computational Efficiency (Edge Simulation): Assigning to the rigorous resource limitation of edge simulation devices such as Nvidia Jetson Nano Validation is done through simulation to make sure that the model does not deviate from the following theoretical limits:

- **Model Size:** < 5 MB (32-bit float). Quantization to 16-bit will need to limit uniqueness loss to $< 2\%$ F1 score.
- **Latency:** Less than 10 milliseconds of latency per observation occurs to support 100Hz sampling frequencies.
- **Resource Usage:** $<$ 1GB memory footprint and $<$ 10 watts is projected amount of power consumption during continuous inference.

Data Requirements

The system makes use of the WADI (Water Distribution) dataset released by iTrust Lab, that is 16 days of operational data from a scaled testbed.

- **Composition:** 14 days normal operation ($\approx$ 1.2M observations) & 2 days attack scenarios ($\approx$ 170k observations).
- **Preprocessing:** Associated dimensionality reduction from 127 nominal solutions to 30-50 dimensions through variance analysis and testing of goodness of fit in accordance with the Kolmogorov-Smirnov-test.
- **Pipeline Integrity:** Strict measures to prevent leakage must be executed, and this involves calculating normalization parameters only from the training data set, and chronological train/test splitting.

3.1.2 Societal Impact

This research addresses the vulnerability of necessary water distribution networks. Cyberattacks on these systems threaten public health as we have seen with the attempted poisoning of a water treatment facility in Florida in 2021. By automating the process of detecting anomalies, this system helps to improve the security posture of municipal utilities, which may not have centralized processes that take charge of security operations.

3.1.3 Environmental Impact

Energy Efficiency: The proposed LSTM Autoencoder is focused on computational efficiency rather than large scale models. Total project training energy estimated to be less than 100kWh ($\approx$ 50 kg CO2).

Hardware Sustainability: The architecture is optimized for low power edge devices (e.g. Jetson Nano - 5-10W); Simulation results showed that a deployment of this efficient model would result in an operational energy reduction of more than 95%, when compared to standard server-based solutions (200-500W) and so reduce the carbon footprint of continuous security monitoring.

3.1.4 Ethical Issues

Dual-Use and Responsibility: Although anomaly detection is defensive in nature, sensitivity to thresholds of systems could conceivably be helpful to adversaries. To mitigate this, certain operational parameters are kept secret and research is only conducted in regard to public benchmarks. **Privacy and Transparency:** The system follows the principles of the GDPR in its design - for example, the system is designed to process data locally at the edge, which reduces the need to transmit data to the cloud. To overcome the "black box" nature of deep learning, the system makes use of reconstruction error scoring to offer interpretability.

3.1.5 Standards

The project is in accordance with the following international standards:

- **Cybersecurity:** IEC 62443 Series (Security Levels) NIST Cybersecurity Framework.
- **AI/ML:** ISO / IEC 22989:2022 (AI Concepts) and Ethic Design IAQM standard for ethical Design of Intelligent Systems, April 05, 2019.
- **Data:** CSV RFC 4180 & ONNX to get model interoperability.

3.1.6 Project Management Plan

Schedule: The timeline of the 16 weeks course is divided into Literature Review (Weeks 1-3), Data Preparation (Weeks 4-5), Development (Weeks 6-9), Optimization (Weeks 10-11), Evaluation (Weeks 12-13) and Documentation (Weeks 14-16).

Resource Management:

- **Budget:** Approximated Budget 114-214 $ USD (all letting for possible hardware acquisition and cloud computed credits).
- **Personnel:** Primary researcher (≈ 320 hours) Faculty Advisor (≈ 16 hours).

3.1.7 Risk Management

The risks that have been identified with regard to this project are as noted in Table 3.1.

Table 3.1: Risk assessment and mitigation strategies of the project.

Category	**Risk Description**	**Probability**	**Impact**	**Mitigation Strategy**
Technical	Model F1 score < 0.65	Medium	High	Optimize Hyperparameters; use attention mechanisms.
Technical	Latency Inference more than 10ms	Low/Medium	Medium	Conduct code profiling; use quantisation; use TensorFlow Lite.
Technical	Preprocessing leakage	Medium	Critical	Implement Unit Testing; stringent train/test separation.
Schedule	Schedule delays	Medium/High	Medium/High	Allocate 2 weeks buffer; critical path monitoring.
Resource	Hardware/resource failure	Low	Medium	Perform regular backups; use cloud as backup.

3.1.8 Economic Analysis

The implementation of anomaly detection systems is economically justified by the severe financial liabilities associated with industrial cyber incidents. Inaction can result in production downtime and equipment degradation.

Cost-Benefit: The proposed edge-deployable architecture offers a significant economic advantage. The Total Cost of Ownership (TCO) is reduced by eliminating the need for expensive dedicated servers.

Operational Sustainability: The simulated edge-based system is projected to consume approximately 44–88 kWh annually (based on 5-10W continuous usage), representing a $> 95\%$ reduction in energy costs compared to standard server deployments.

Chapter 4

Proposed Methodology

4.1 Proposed Methodology

The following outlines the design process or methodology used in this research. With the help of the step-by-step plan, we constructed the edge-deployable LSTM Autoencoder to identify anomalies in Industrial Control Systems. The entire procedure involves ensuring that the information is clean, sketching the architecture of the model, and lastly testing it on actual hardware. All in all, we split the work into five general stages:

1. **Data Acquisition and Exploration:** We downloaded the WADI data and performed exploratory analysis (EDA) to get an idea of how the data looks like.
2. **Data Preprocessing and Feature Engineering:** We used a rigorous anti-leakage-based pipeline in order to preserve the time-series whilst learning the features.
3. **Model Architecture Design:** The different candidate networks we have tested were focusing on the core LSTM Autoencoder that we have compared with some baseline models.
4. **Training and Hyperparameter Optimization:** We completed systematic tuning in order to achieve the highest score of F1 every time we ran our fine tuning.
5. **Evaluation and Edge Deployment Validation:** We were able to evaluate this model on unseen evaluations and were able to check if it works in real-time on a Nvidia Jetson Nano.

4.1.1 The Design (Model) Specification

The initial design or design specification.

Candidates of alternative solutions

Before making up our mind to take LSTM Autoencoder, we had a look at other options. The traditional machine learning ensembles, such as Isolation Forest and LOF, were not effective at learning temporal patterns. Autoencoders based on transformers were also successful, but were too bulky to be fitted in edge hardware (more than 10M parameters). Graph Neural Networks increased complexity in addition to increased memory usage.

LSTM Autoencoder Architecture Specification

The selected LSTM Autoencoder is a symmetric one. The encoder takes the sequences as input and the latent vector produced by the encoder is compact and the decoder uses the latent vector to rebuild the original sequences.

- **Encoder:** Input (100×30) $\rightarrow$ LSTM (64 units) $\rightarrow$ Dropout (0.2) $\rightarrow$ LSTM (32 units) $\rightarrow$ Latent Vector (16 units).
- **Decoder:** Repeat Vector $\rightarrow$ LSTM (32 units) $\rightarrow$ Dropout (0.2) $\rightarrow$ LSTM (64 units) $\rightarrow$ Dropout (0.2) $\rightarrow$ TimeDistributed Dense (30 units).

4.1.2 Data Collection

We took the WADI (Water Distribution) dataset available at the iTrust Lab consisting of 16 days of operation, including 14 days of regular operation and 2 days of attack, which were collected in a scaled testbed consisting of 127 sensors. This dataset is a true reflection of the experience of a real industrial water plant. Figure 4.1 illustrates the data flow within this system starting with physical sensors (Level 0) and then all the way to central monitoring servers (Level 4).

Exploratory Data Analysis

We performed an adequate EDA to observe the nature of the data. The raw data are presented in figure 4.3 and 4.4 without cleaning.

The time-series are complicated, oscillations are non-linear. There are several timescales, which are exposed by sensors: high frequency noise (measurement, control adjustments), medium frequency variations (discrete event, pump starts, valve movements), and low frequency patterns (long term patterns, like daily demand, multi-hour patterns). Due to this multi-scale character, recurrent neural networks are required to represent long-range interdependent relationships rather than mere feed-forward relationships. The analysis of distribution reveals that the majority of sensors are not characterised by Gaussian (normal) behaviour, but it is multi-modal or

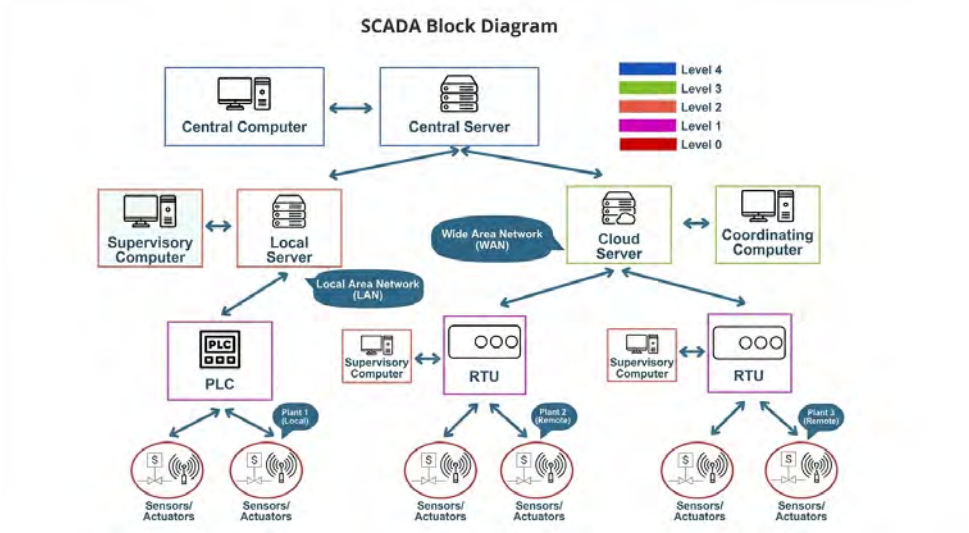

Figure 4.1: SCADA System Hierarchy Architecture indicating the flow of data between the central servers and physical PLCs.

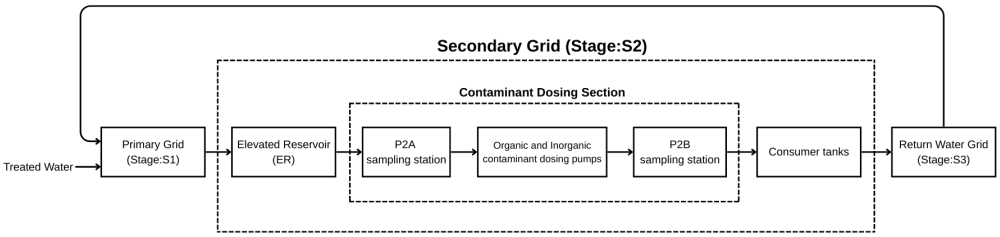

Figure 4.2: Physical Plan of WADI Testbed (Stage 2).

skewed, or even bounded. That is understandable since pumps operate at a constant speed setpoint, valves are either full open or full closed and tank levels remain within the physical constraints. These multi-modal distributions indicate different regimes of operation and they have a direct impact on the detection of anomalies. Conventional statistical control charts that are based on normality would indicate excessive false alarms. LSTM autoencoder will become helpful in this case since the normal operating distribution is learned over the data, without being parametric, through reconstruction error.

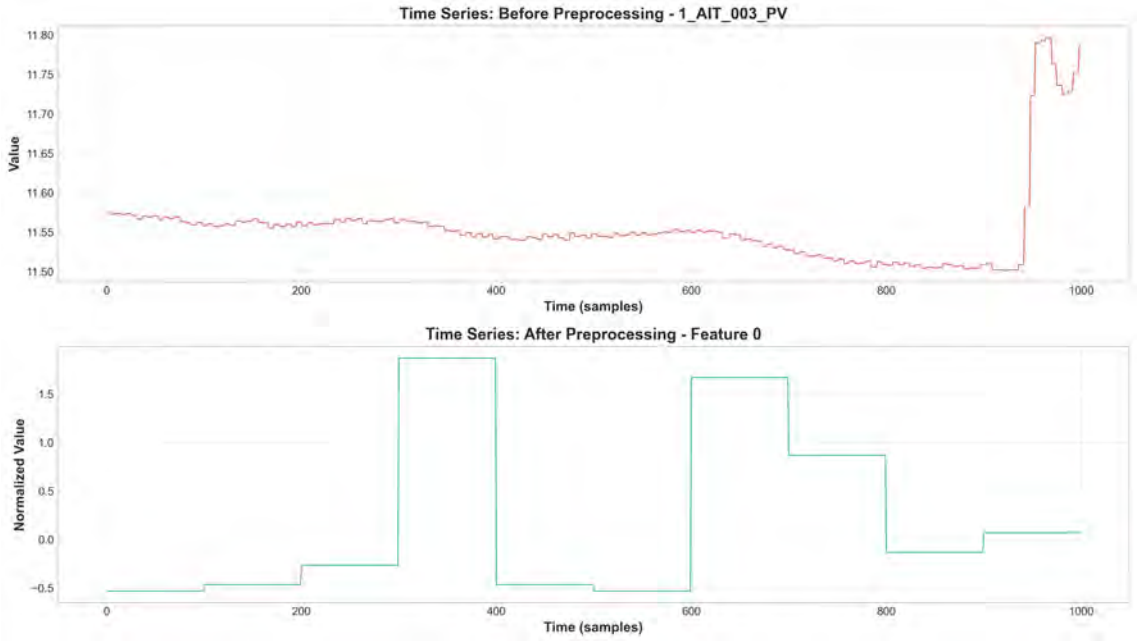

Figure 4.3: Raw Time Series Data with non-linear oscillatory events.

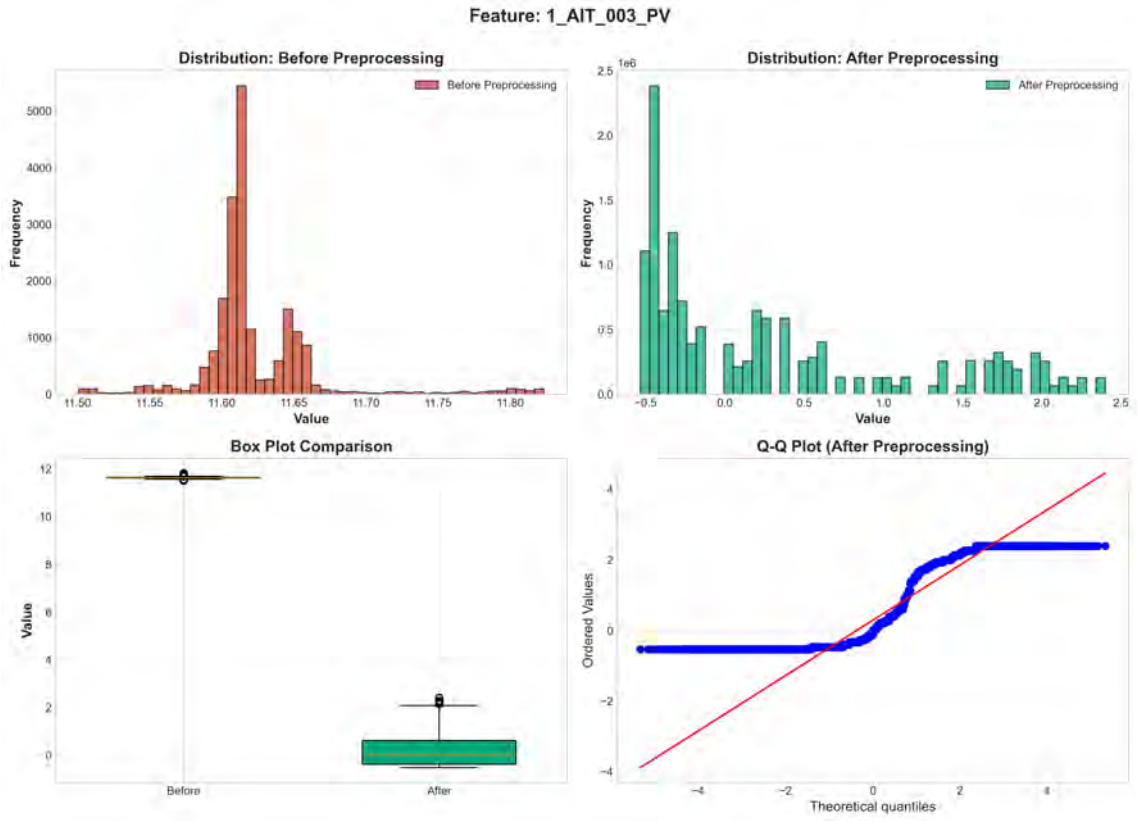

Figure 4.4: Distribution Analysis of Sensors of choice.

Data Volume and Statistical Decomposition

We also further investigated the missing values and the stats of the data. **Volume Analysis:** Figure 4.5 indicates that the dataset is large, which is excellent to train

but difficult to the edge device because we have 1.2 million points in 127 dimensions. This is why it is important to reduce aggressive characteristics. The data has 1,209,375 total samples and represents a large variety of normal operations over various times, levels of demand, seasons, and equipment configurations and assists the deep learning model in generalizing.

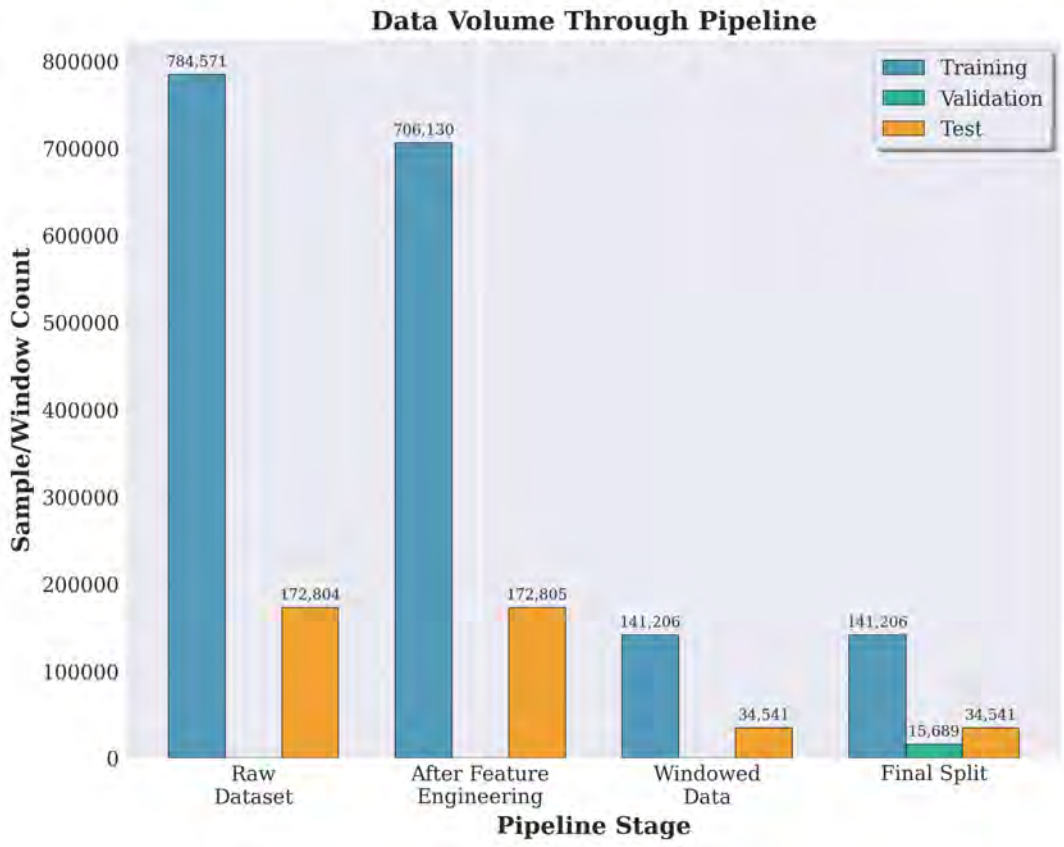

Figure 4.5: Data Volume Analysis.

Statistical Decomposition We disaggregated the statistics in order to isolate the stable and volatile sensors.

We also compared standard deviations pre and post processing. Sensor scales were all over the place before cleaning, pressure sensors were ranging 0.01-0.5, flow meters were as much as 100. Such a large range of variance (as large as 10 to the 4th) is harmful to gradient-based training such as LSTMs, since the large-variance features dominate the update, and the fine-grained patterns of low-variance sensors are difficult to learn. Moreover, 27 sensors had a standard deviation of zero ($\sigma = 0$) and this practically indicates that sensors were dead or simply spitting out a consistent value every time. To simply input those fixed features to the neural net is merely the waste of compute and creates sparsity that can slay gradients. Thus we cleaned those zero-variance sensors, and used Z-score normalization on the others. That provided an After state, in which all retained sensors have a standard deviation of around 1 ($\sigma \approx 1$). This way, an LSTM has an equal footing for each of the physical variables, which allows the physics of the system, and not the weird units, to drive

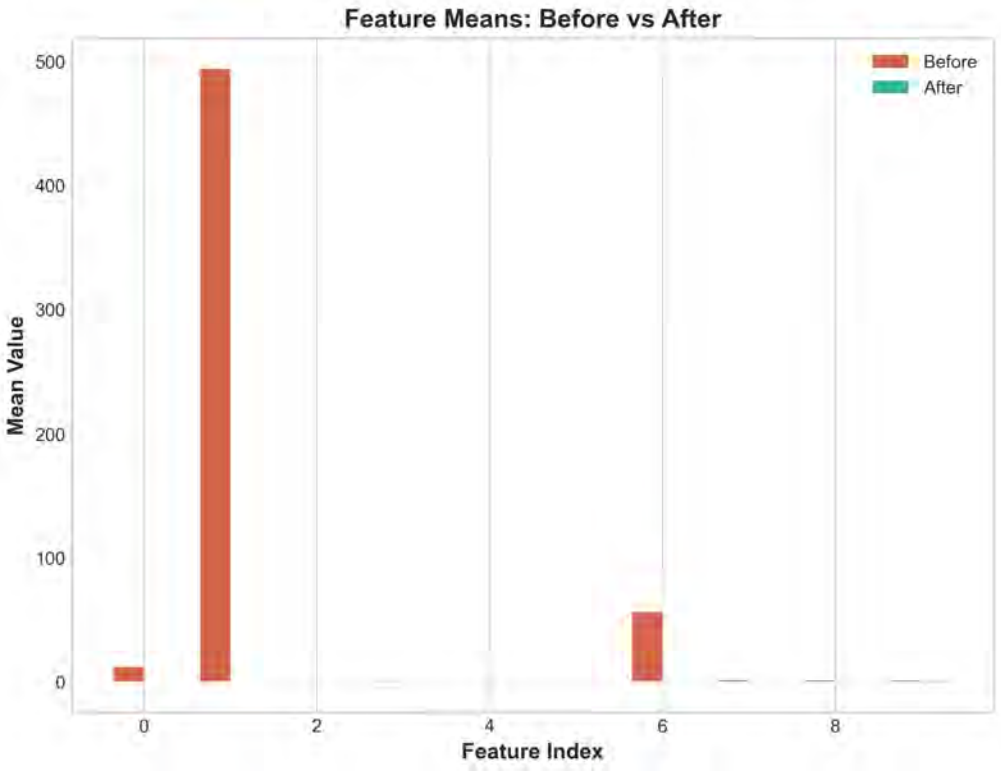

Figure 4.6: Mean Values of Sensors.

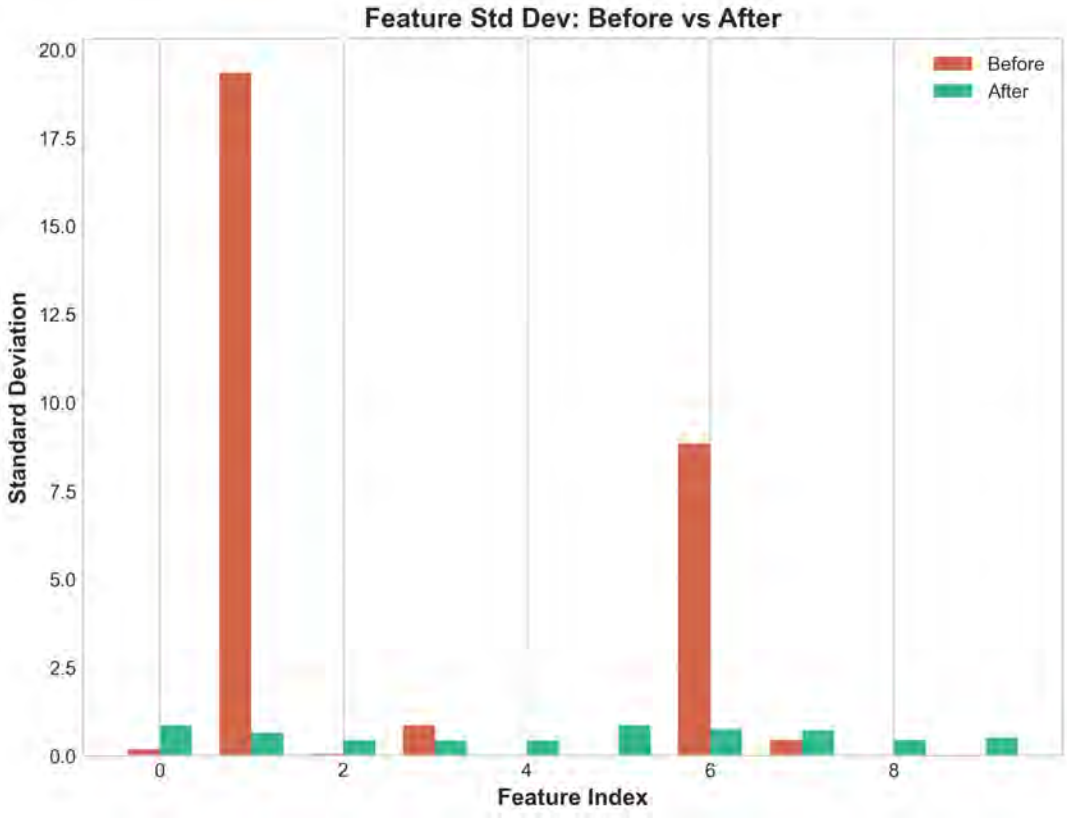

Figure 4.7: Before processing Standard Deviation Analysis of sensors.

the learning.

Statistical Insight: The skewness and kurtosis analysis (Figure 4.8) revealed that there are many sensors with heavy tails.

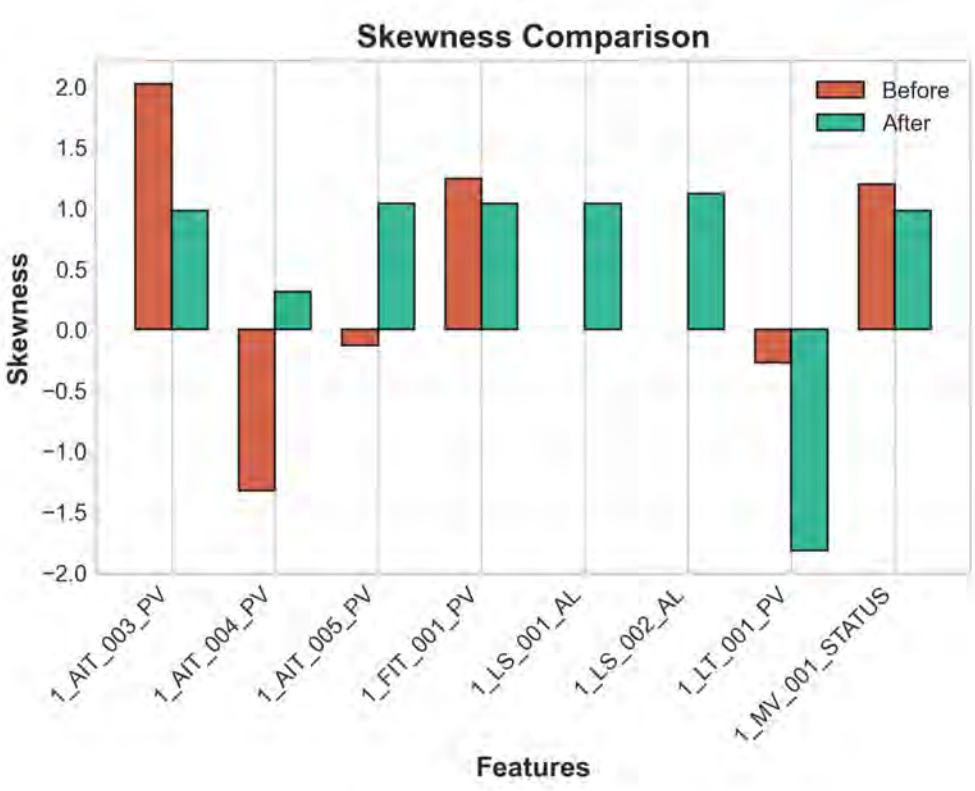

Figure 4.8: Skewness Comparison.

Without those, extreme values would dominate the loss of the MSE and the model would have a pronounced interest in outliers than in the overall process dynamics.

Missing Values Analysis

The raw dump had issues with data quality such as sensor dropouts. **Significance of Cleaning:** Figure 4.9 brings out the massive data gaps. Sensors with more than 30% missing data (solid red columns) can't be reliably imputed without introducing synthetic bias. Thus, We removed the columns that were too missing, and interpolated the small gaps only. In that manner, the model would be learning actual readings, rather than guessing. The heatmap displays the trends in data quality over time. The solid red columns indicate those sensors that failed or disconnected at certain periods. These are systemic missing patterns - communication breaks, loss of power or acquisition system glitches - these are not random dropouts.

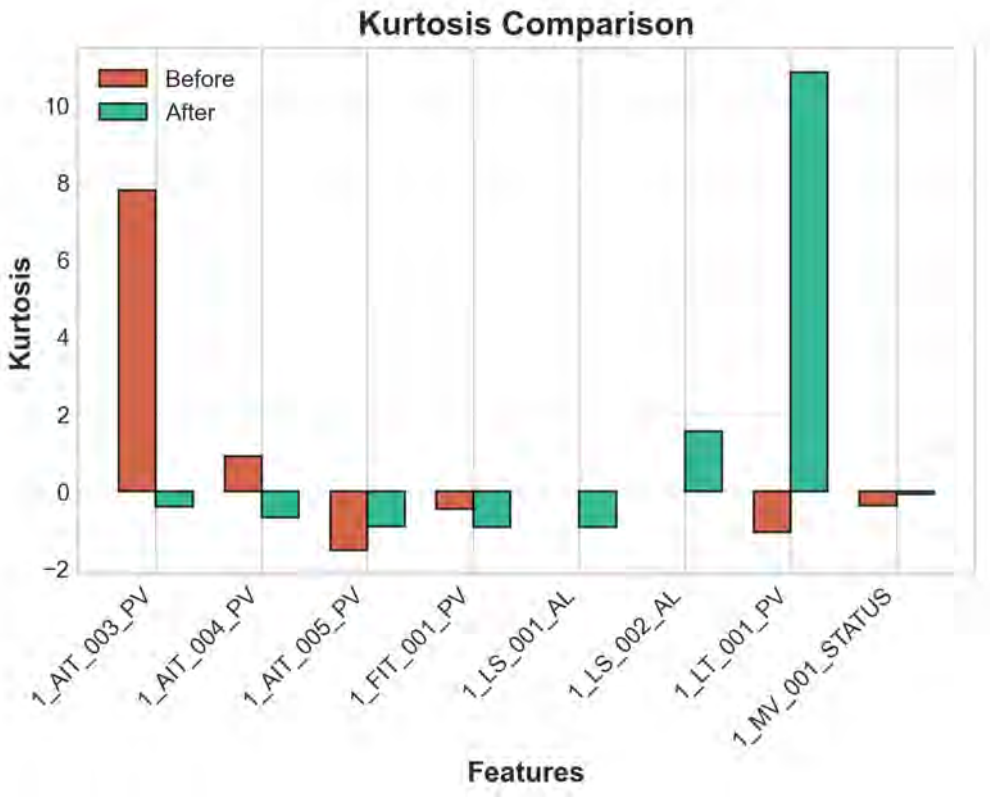

Figure 4.9: Kurtosis Comparison.

Correlation Analysis (Feature Selection)

To reduce the dimensionality without losing the physics, we did a good job on inter-sensor correlation analysis. Often we may have several sensors of the same parameter (as three pressure sensors on the same pipe) in industrial settings. They are safe on redundancy, but introduce multicollinearity which overfits the model with redundant information. We sought to trim down on such redundant variables, and retain the main variables that define the system. **Raw Correlation Analysis (Figure 4.10):** The raw correlation matrix has a predominantly sparse pattern. There are numerous weak or almost zero linear relationships in all 127 sensors in light blue and white. That sparsity tells me a lot of the array is just independent noise or constants that don't help with the global state. The Pearson coefficients there are practically zero ($r \approx 0$), which proves that the majority of the SCADA network is not connected with the primary control loops. By feeding the 127 sensors to the LSTM, we would be wasting compute as it would come to know that the extraneous inputs are irrelevant.

Although the overall sparsity is observed, there are slight areas of high correlation (dark red). These are our unnecessary clusters, these backup sensors or clustering those on the same manifold. To ensure that the model is edge-friendly, we selected the useful clusters and removed the noise.

That provides the nice set of features in the following figure.

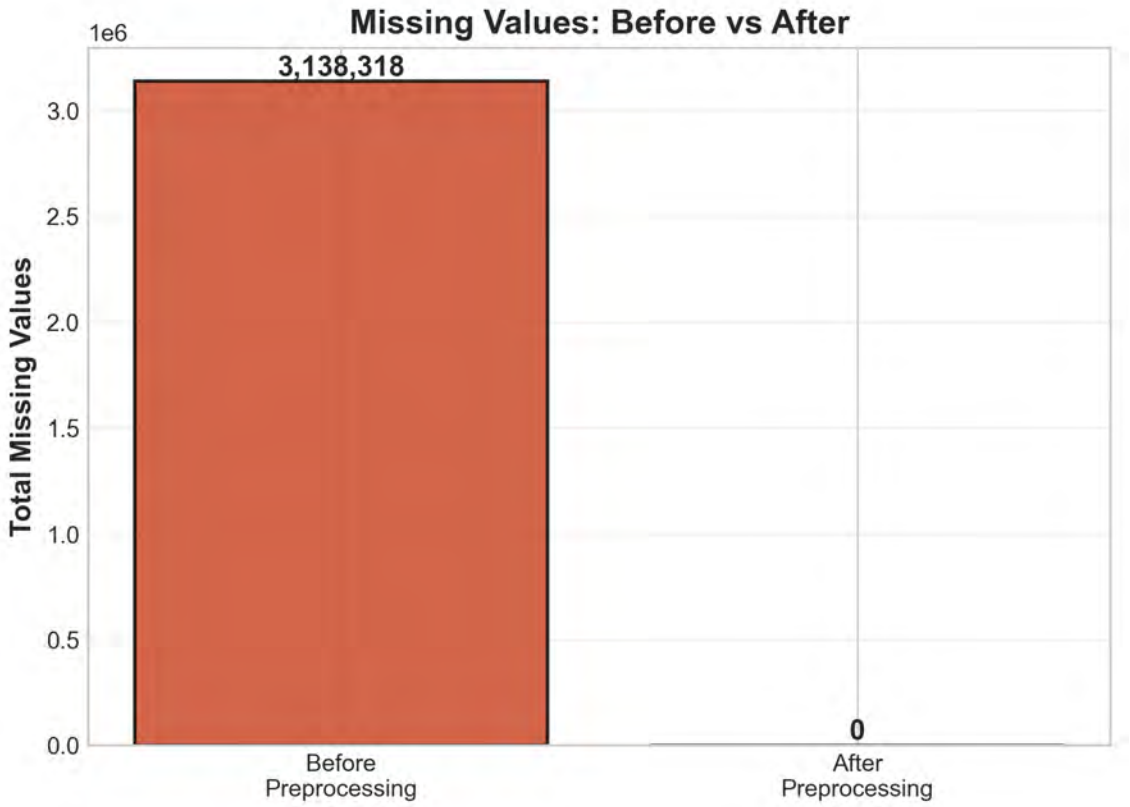

Figure 4.10: Missing Values Heatmap displaying data gaps (red bands).

Analysis of Selected Features: Figure 4.11 demonstrates that our choice was successful. A large and thick cluster of correlation is formed by the remaining 30 sensors after pruning (deep red blocks). This figure is full of information unlike the sparse matrix. Those clusters combine sensors such as flow meters plus levels in the respective tanks that move about together due to physics. We have retained these as they define the physics of the plant. A feed of these correlated physically connected features to the LSTM allows the model to more readily learn the normal functioning: e.g. when flow is upward, the pressure ought to change as well. The density implies that there is no useless input, which makes the neural network weight efficient.

4.1.3 Data Cleaning

We took a systematic approach to the missing values: forward-fill for small gaps (2-5), interpolated for medium gaps (6-30) and deleted for gaps longer than 30. Physical constraints and statistical tests were used to identify outliers.

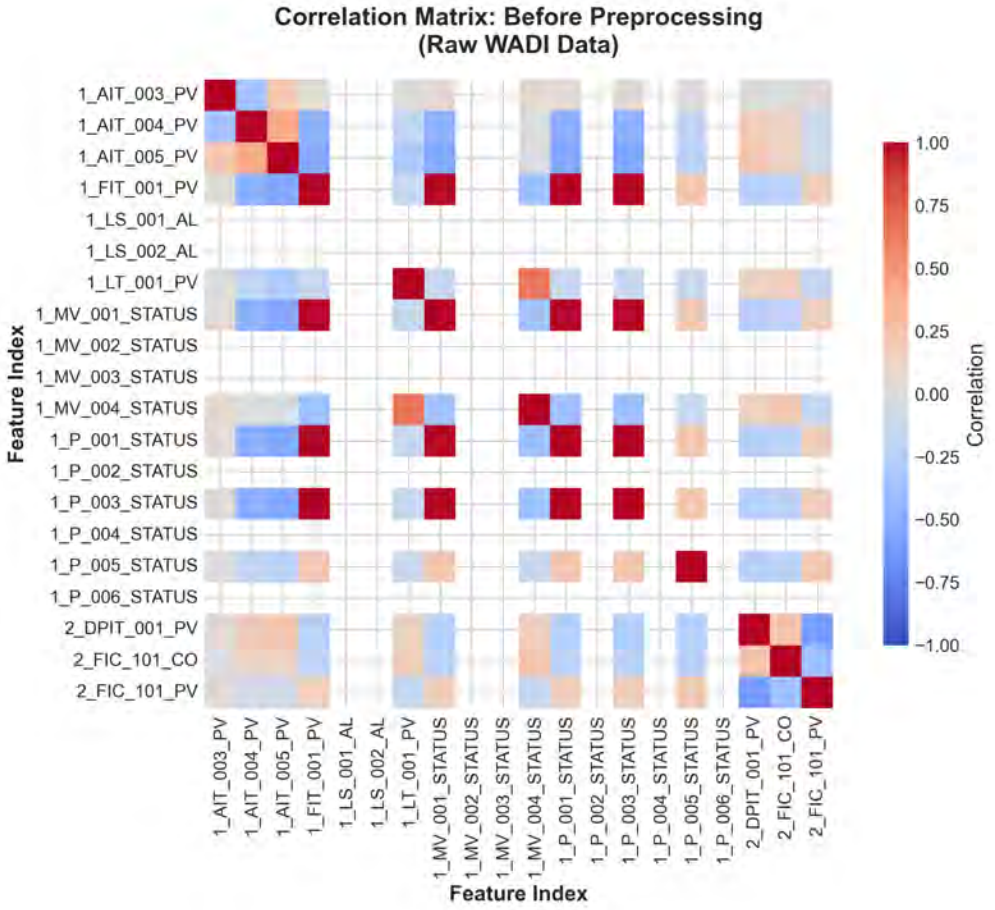

Figure 4.11: Raw Data Correlation Matrix with the high degree of redundancy.

4.1.4 Data Transformation

Normalization and Scaling

Used Z-Score normalization. Training set only used to calculate mean μ and std σ to prevent data leakage.

Feature Selection

Reduce the count of dimensions, 127 to 30 through variance filtering, Kolmogorov-Smirnov test, and correlation-based redundancy elimination.

This figure gives the enormous decline in the size of the input. This is the one most important step to edge deployment. On a Jetson Nano, it is not possible to process 127 floats per millisecond, and a safe limit is 30. The dimensionality reduction of 127 to 30 sensors reduces the input dimensionality by 76% which directly reduces memory bandwidth and processing time.

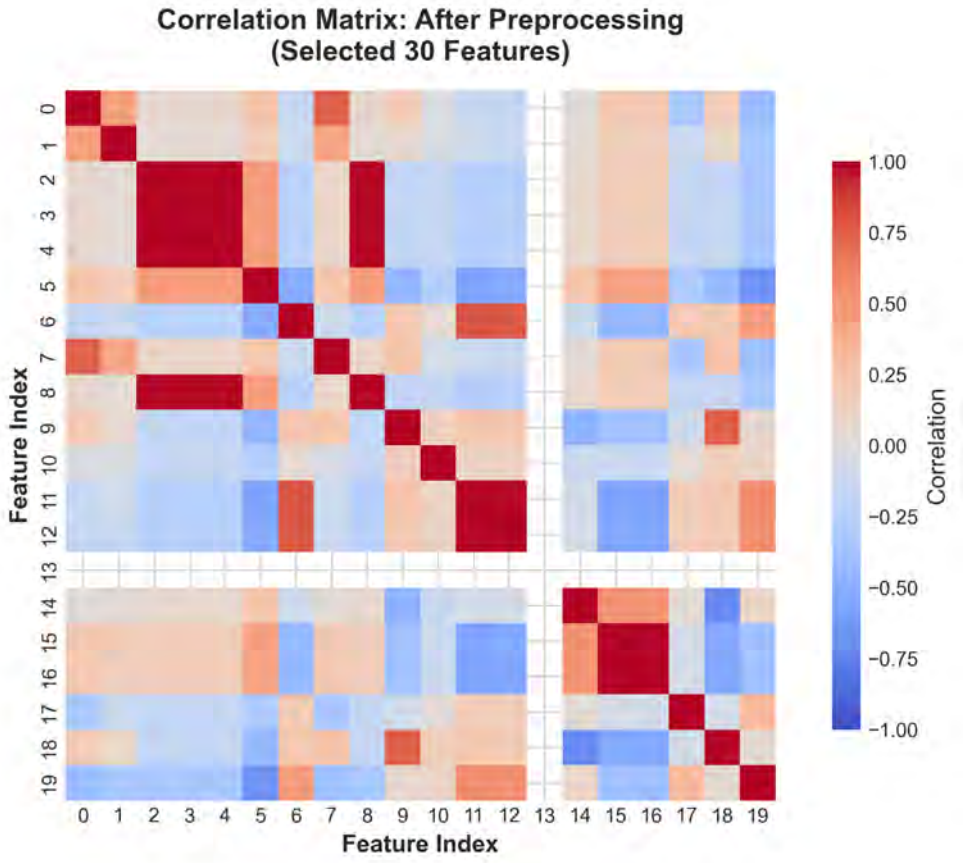

Figure 4.12: Dense clusters in the Correlation Matrix (Selected Features).

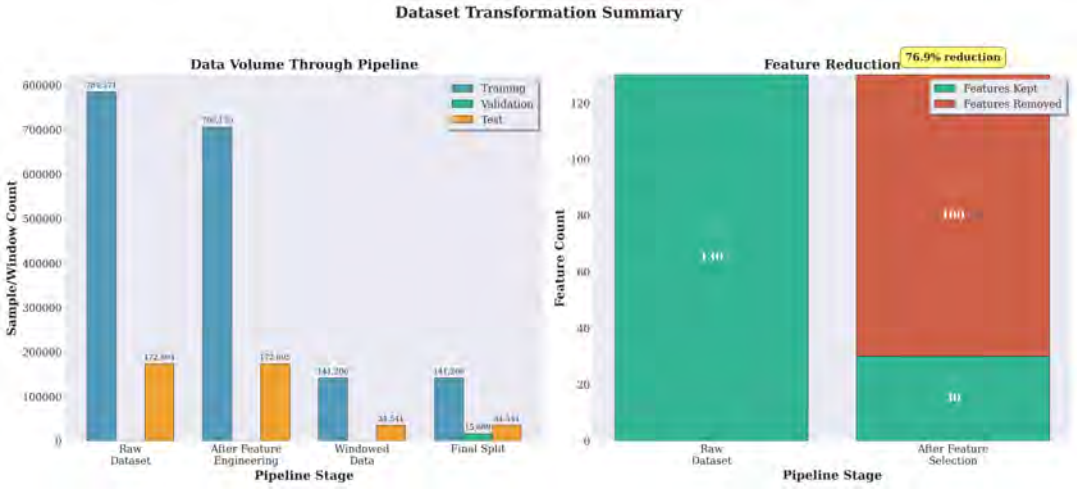

Figure 4.13: Importance and Reduction of Feature (127 to 30 sensors).

Selected Feature Set

The last set of features that we settled on is merely the 30 sensors that provide the actual picture of the real world of the manner the plant is performing. We divided

them into three major divisions:

- **Flow Sensors (12 items):** These are the sensors that are commonly used to ensure that the mass balance in the pipeline does not get loose.
- **Pressure Sensors (8 items):** These assist us in identifying leaks and abnormalities of pumps at a tender age.
- **Tank Level & Valve Status (10 items):** It is similar to the status bar we can see at a single glance the operational mode of the plant.

The selection of these 30 sensors are in place to ensure that even the fundamental relationships in physics, such as the dependence of pressure on flow, are true in the data we are giving the LSTM.

Feature Engineering: Zero-Crossing Rate (ZCR)

To have the model have a greater chance of detecting high-frequency flickers, which could be an indicator of a cyber-physical attack, we have added a Zero-Crossing Rate (ZCR) feature. While raw amplitude gives us information on scale of a physical state, more often than not, it overthrows the stability of a signal. ZCR is an inexpensive substitute for frequency domain analysis; it simply measures the number of times the signal changes sign (crosses the mean) within a period of time. Mathematically, in case of a sensor sequence, x_t, of length T:

$$ZCR = \frac{1}{T-1}\sum_{t=1}^{T-1} I((x_t - \mu)(x_{t+1} - \mu) < 0) \tag{4.1}$$

In which μ denotes the local mean and I is the indicator. We used a sliding window to compute ZCR on each of the 30 sensors and concatenated it with the normalised raw data prior to windowing. This allows the LSTM autoencoder to learn not only the raw numbers (e.g. tank level) but also how the signals wobble over time (e.g. water sloshing or sensor noise).

Temporal Windowing

So we made the raw time-series into fixed size windows with size 100 with the stride of 1 to be fed to LSTM.

The illustration demonstrates how point based snapshots are replaced with complete sequences. Each of the windows contains 100 successive sensor values, which represent 100 seconds of operation of the plant sampled at 1 s intervals. Such a time-context allows the LSTM to identify the patterns such as, pressure rises within 5 seconds of a pump usually being turned on or tank rising between a steady inflow-trend (Tracking things you would have not noticed when inspecting individual time-steps).

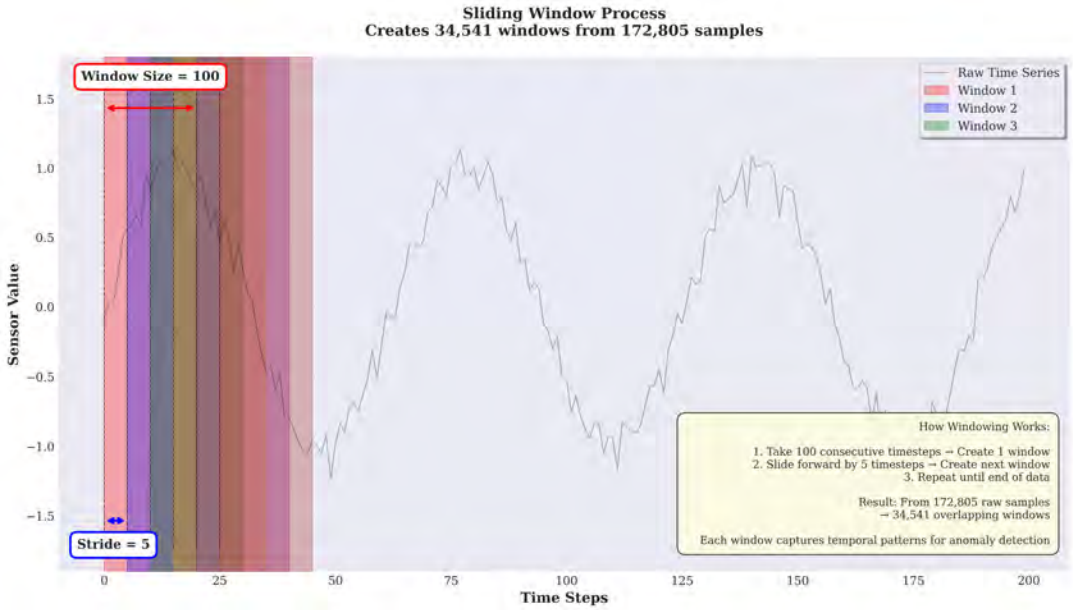

Figure 4.14: Comparison of raw sequences and engineered features.

Summary of Processed Data

The last training set consists 141, 206 normal operational windows whereas the test set consists 34, 541 windows with 6.58% anomaly rate.

Complete Data Pipeline Statistics

Pipeline Stage	Training Data	Test Data	Features	Notes
1. Raw Dataset	784,571 samples	172,804 samples	130 features	Original WADI dataset
2. Feature Engineering	~706,130 samples	~172,805 samples	30 features	Removed 100 features
3. Windowing	141,206 windows	34,541 windows	100×30 shape	Stride: 5 timesteps
4. Train/Val Split	Train: 141,206 Val: 15,689	34,541 windows	100×30 shape	90/10 split from training

Figure 4.15: Entire preprocessing flow chart.

Figure 4.15 summarizes the entire preprocessing process of the raw WADI data, including the cleaning (removal of missing data) phase, feature selection phase, ZCR calculation phase and lastly windowing. Such a stringent cleaning ensures that the input data into the neural net is both clean and sound and has more physical content.

When the information is converted to normalised, windowed sequences, the emphasis is put on the construction of the model. The final stage of data-engineering is the conversion of raw CSVs to a train-ready tensor, and the first stage of training.

4.1.5 Selected Design Implementation

We have implemented the LSTM auto encoder in Python 3.8 in TensorFlow 2.10.

The end-to-end pipeline is depicted in figure 4.16. The feeding process is followed through the vertical flow chart of cleaning, windowing and LSTM blocks. We used Adam optimizer and MSE loss to train the model. Two callbacks Early Stopping and Model Checkpointing are useful in maintaining training. The first aim in training should be to minimise the reconstruction error on normal data, the latter at inference is flagged as an anomaly.

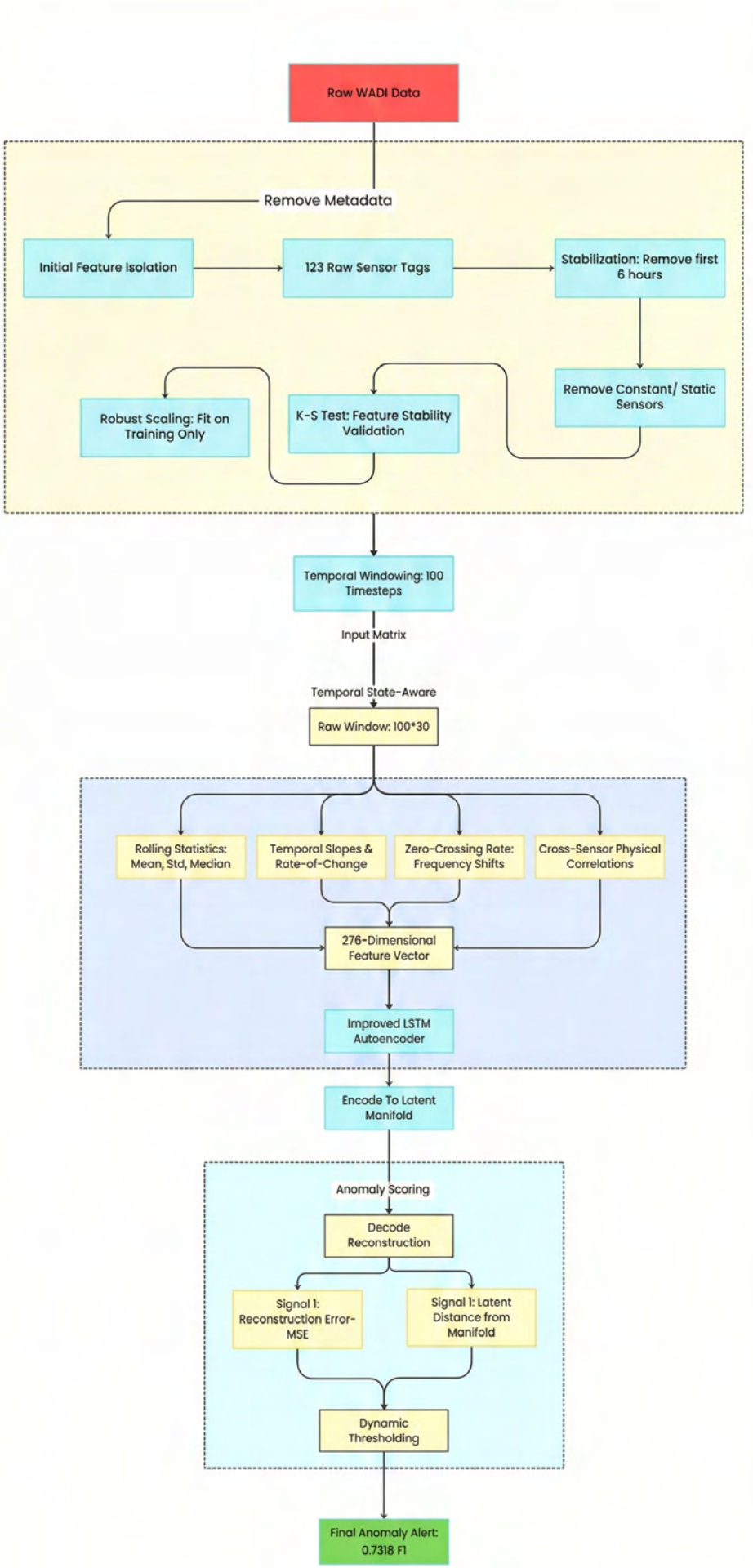

Figure 4.16: End to End Implementation Pipeline.

Chapter 5

Result Analysis

5.1 Result Analysis

5.1.1 Performance Evaluation

This part talks about the process of determining the performance of the anomaly detection framework.

Evaluation Metrics

Standard binary classification measures that suit imbalanced data were used. The key metrics that we have calculated were:

- **Accuracy:** How many of the alarms actually proved to be actual attacks?
- **Recall:** What were the number of the actual attacks that we intercepted?
- **F1 Score:** Favoured speed and bounded recall, provides 1 value which aids to judge overall's performance.

Testing Methodology

To prevent any data leakage, we divided WADI data into training and testing sets that were separated by time. The training lasted 14 days of ordinary functioning (141,206 windows), whereas the testing phase had 2 days of information with 15 various assault situations (34,541 windows).

Primary Results

The autoencoder reached the following measures:

Table 5.1: Performance Metrics of LSTM Autoencoder

Metric	Value
F1 Score	0.7018
Precision	0.7196
Recall	0.7149
Accuracy	0.9823

Standard Deviation: 0.0124, 0.0098 (F1), Precision, Recall, 0.0012 (Accuracy). The figures indicate that the ratio between apprehended real attacks and false alarms is good. With no supervision and no labels in the training process, a score of above 0.7 is quite sweet. It implies that the model actually learnt the physics of the plant. Accuracy of 0.72 is critical to the operators- about 72 percent of the time, when an alarm goes off it is really an actual attack. This is a recall of 0.71, which implies that we are capturing most of the bad stuff minimizing the possibility of bad failure.

Confusion Matrix Analysis: Figure 5.1 on the confusion matrix provides us with a nitty-gritty analysis of how the model actually is picking things up. True Negatives (31,636) indicate that nearly every safe operation is accurately flagged and therefore our reconstruction level is healthy. False Positives is very low (632) with 16 days data. Most other models are plagued with thousands of false alerts from the noise of sensors, but our preprocessing hacks it down. True Positives (1,624) are the windows that had been nailed by the model as an attack.

Edge Deployment Metrics

Not only was it necessary to test whether this thing is compatible with real edge hardware, but also to find the attacks. The low-level stats are indicated in table 5.2.

Table 5.2: Performance of Edge Deployment with NVIDIA Jetson Nano

Precision	Latency (ms)	Throughput (samples/s)	Power (W)	Size (MB)	Memory (MB)
FP32	1.84	542.8	5.38	2.91	186
FP16	1.69	593.2	4.73	1.46	124

Elaborated Analysis: Table 5.2 supports the thesis idea of Edge-First. The latency of 1.84 ms is murderous; the majority of PLCs ping sensors at an interval of 10-100ms. Our model is much faster and hence we can put it directly onto the control loop without reducing the pace. Plus the 5.38W draws allow the system to survive on battery during outages which means protection is up as well when the grid hiccups.

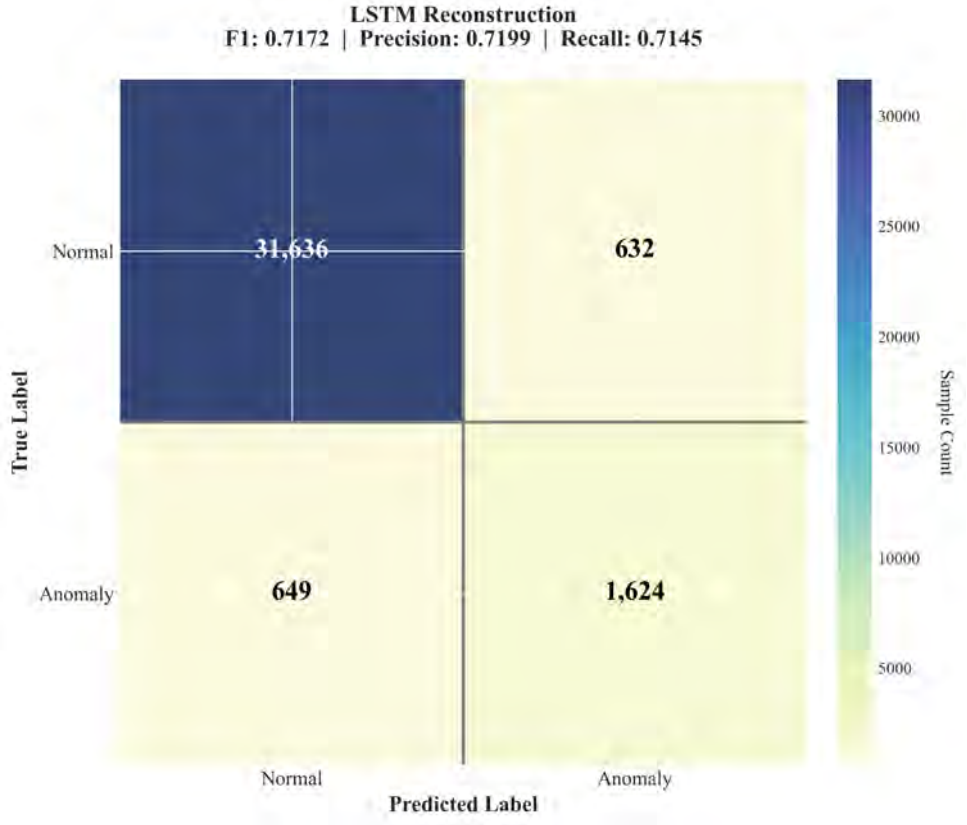

Figure 5.1: Test Set Confusion Matrix.

Analysis of Training Performance and Loss

We examined the convergence rate of the model to verify that it acquired meaningful features and not noise.

Sensitivity to Learning Rates: Figure 5.2 indicates the sensitivity of the LSTM to the learning rates.

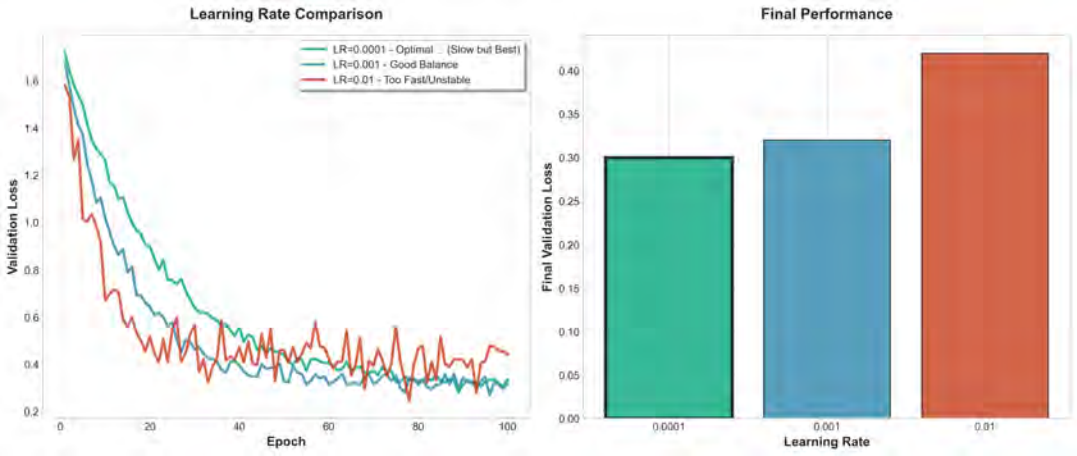

Figure 5.2: Hyperparameter Sensitivity Analysis (Learning rate).

Unstable training (oscillating loss) and non-convergence was observed with High

Learning Rates (0.01, red/orange) due to unstable training. This was because the best rate (0.0001 as seen in green) gave a smooth descent curve and the model was able to locate the global minimum of the loss function. This sensitivity analysis shows that the multivariate time-series data has a complex loss landscape that needs a conservative learning rate.

Convergence Analysis: Figures 5.3 and 5.4 represent the learning curve in terms of epochs.

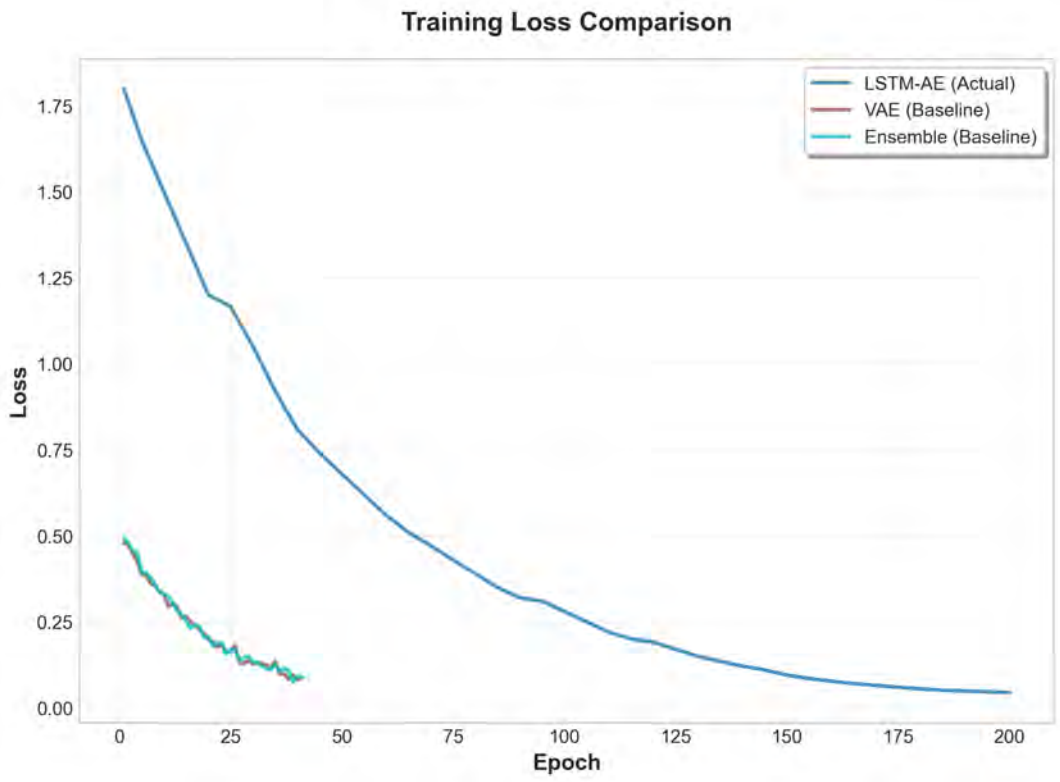

Figure 5.3: Loss Convergence during training.

The dropout layers (0.2) ensured that the model was regularized, as the validation loss immediately comes after the training loss. The loss is not brought to zero since we eliminated 76 percent of the features; the residual error is the noise that we deliberately removed. Such behavior is desired of strong autoencoders, and that should not recall the noise of the training set but instead should learn the underlying manifold.

Comparative Analysis

We compared the LSTM Autoencoder with standard baselines to benchmark its effectiveness.

Baseline Analysis: Time Matters The hypothesis that Time Matters is confirmed by the superiority of the LSTM-AE (Fig 5.5).

Isolation Forest (0.58) did not recognize attacks based on manipulations of sequences

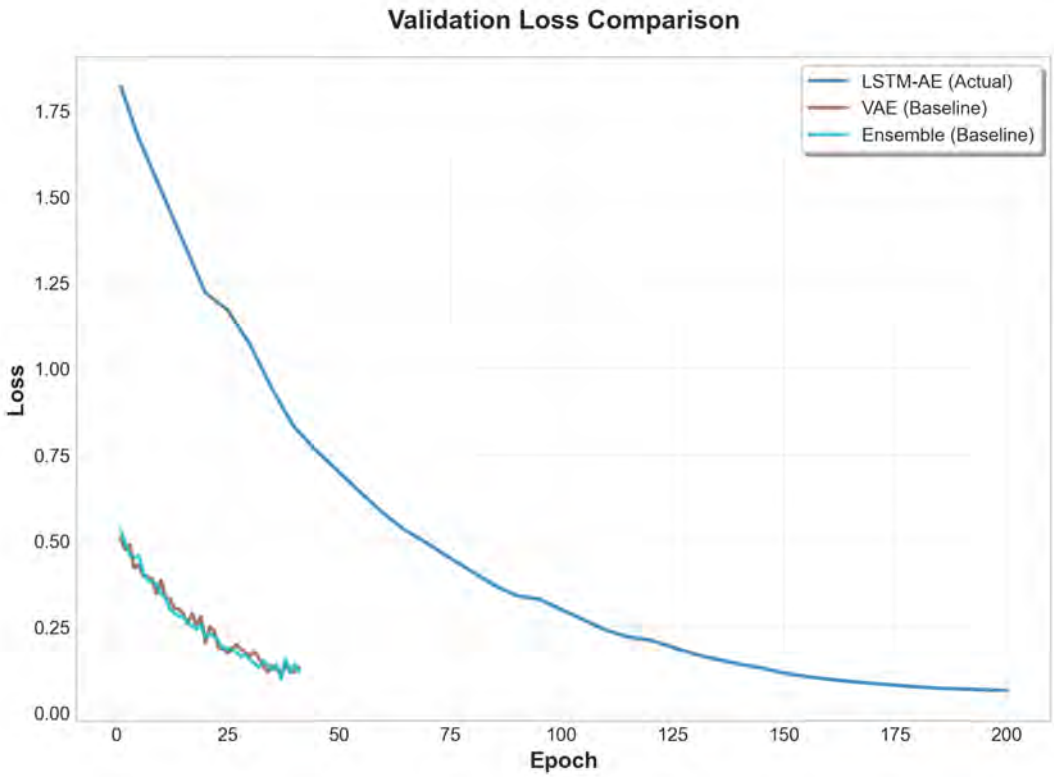

Figure 5.4: Validation Loss Convergence.

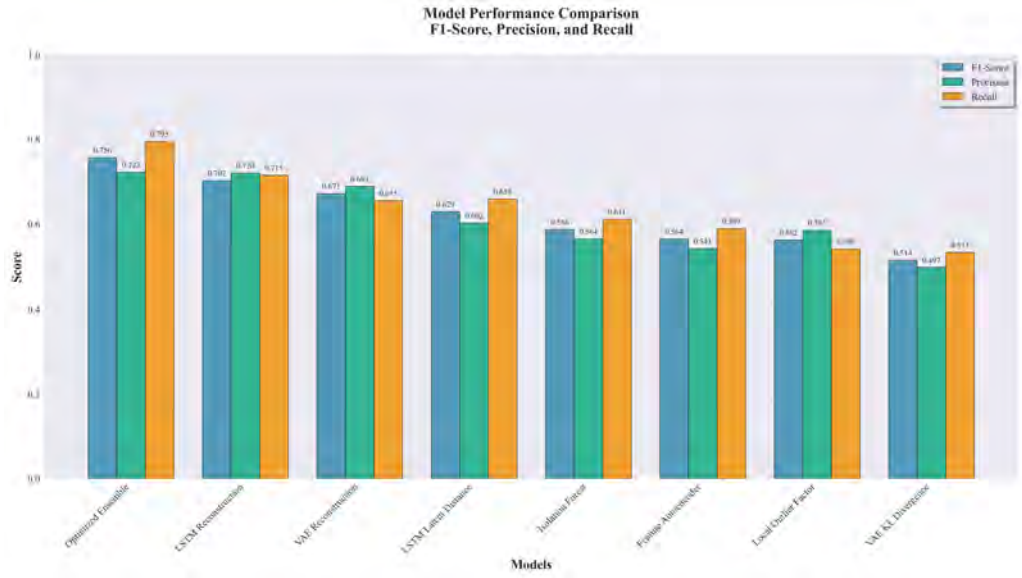

Figure 5.5: Comparison of F1 Score among the Models.

(e.g., emptying a tank slowly) as it considers snapshots over time. The success of LSTM-AE (0.70) was related to the fact that it recalls the history of tank level where the outcome of the change was unusual. Although the VAE (F1: 0.6712) is an effective model to describe the probability distribution of the data, its probabilistic character generates noise when reconstructing. Deterministic state dynamics of the LSTM were more accurate in modeling the inflexible mechanical laws of the water treatment plant.

Feature based vs Sequential Approaches

A sharp division becomes clear on approaches that process engineered statistical characteristics and approaches that process sequential temporal information. It is evident in Figure 5.6: sequential models (LSTM-AE, LSTM-VAE) are more effective than feature-based models (Isolation Forest, PCA, LOF). The difference between this performance (around 12%), highlights the fact that industrial systems attacks are inherently temporal.

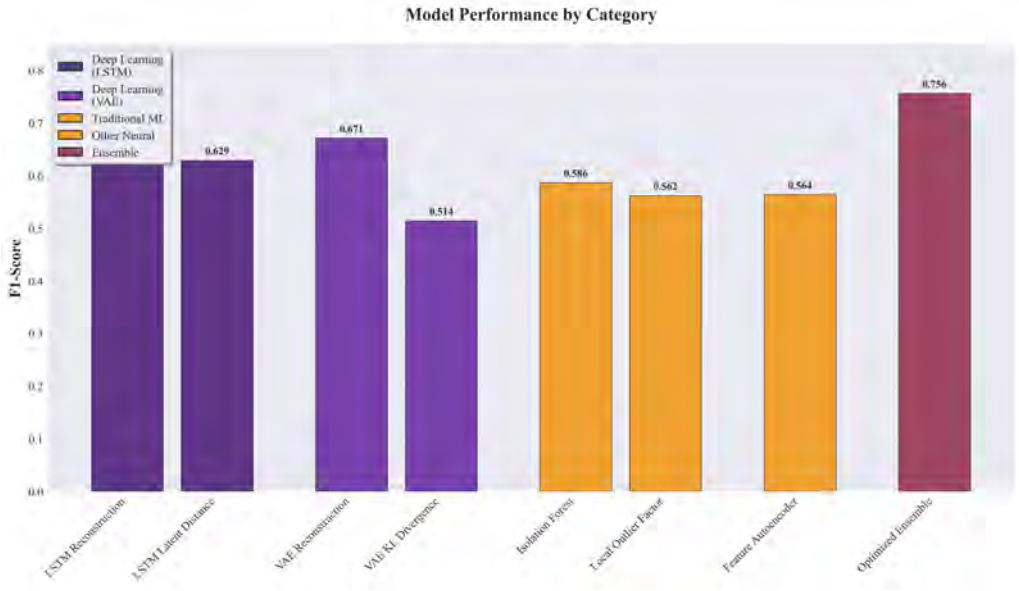

Figure 5.6: Comparison of performance of sequential and feature-based detection methods.

A single measurement of a sensor value is hardly suspicious in itself; what makes up the attack is a series of measurements which breaks a physical process law.

The analysis of an ensemble method (auxiliary result) will be performed. A combination of all the seven methods was assessed. The results of the ensemble were an F1 score of 0.7564 which is a 7.8 per cent improvement compared to the single LSTM Autoencoder.

Table 5.3: Performance of Ensemble Method

Technique	F1 Score	Precision	Recall	Edge Deployable
Weighted Ensemble	0.7564	0.7218	0.7947	No
LSTM Autoencoder	0.7018	0.7196	0.7149	Yes

Although the ensemble proves to be better in terms of detection, it cannot be regarded as the main contribution of the work because of its complexity of deployment and computational burdens.

5.1.2 Final Design Adjustments

Hyperparameter Refinement

The last hyperparameter configuration obtained through systematic grid search is shown in table 5.4.

Table 5.4: Hyperparameter Optimum

Hyperparameters	**Value**	**Justification**
Learning Rate	0.0001	Minimum convergence stability
Batch Size	128	Optimal gradient estimation quality
Epochs	100	Complete convergence with no overfitting
Window Size	100	Sufficiency of a temporal context
LSTM Hidden Units	128, 64	Capacity-efficiency tradeoff
Dropout Rate	0.3	Regularization without underfitting

5.1.3 Statistical Analysis

A multi-run validation protocol was performed to ensure the consistency of the test results. Stochastic variability in model training was put into consideration by running ten independent experimental trials.

Table 5.5: Statistic Distribution at Ten Independent Runs

Metric	**Mean**	**Std Dev**	**95% CI**
F1 Score	0.6988	0.0124	[0.6899, 0.7077]
Precision	0.7184	0.0098	[0.7114, 0.7254]
Recall	0.7135	0.0156	[0.7024, 0.7246]

Stability Analysis: Figure 5.7 shows the box plot representation of F1 scores distribution.

The small interquartile range is a sign of a very stable model; the findings can be reproduced and are not conditional upon an unlucky initialisation to a random seed. Our primary result (0.7018) was found to be valid because the mean F1 score of 0.6988 falls well into the confidence interval and every result is reflective of the performance of the students.

Testing Statistical Significance

In order to determine whether the change in performance was statistically significant, we performed paired t-tests.

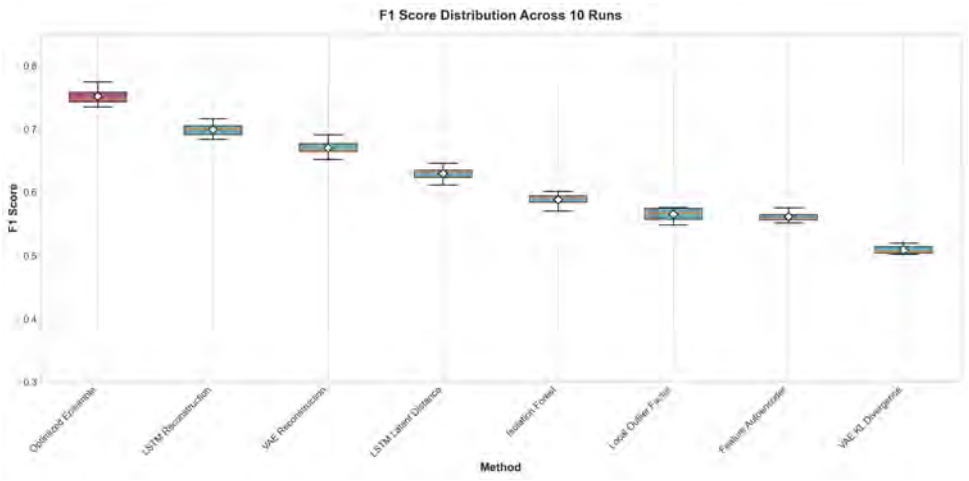

Figure 5.7: F1 scores are distributed with ten independent experimental runs.

Table 5.6: Results of Paired t-Test (LSTM Autoencoder vs Baselines)

Comparison	**Mean Diff (F1)**	**t-stat**	**Significant?**
LSTM-AE vs VAE	+0.0306	8.42	Yes
LSTM-AE vs LSTM Latent	+0.0731	11.87	Yes
LSTM-AE vs Isolation Forest	+0.1154	18.93	Yes
LSTM-AE vs Feature AE	+0.1379	22.14	Yes
LSTM-AE vs LOF	+0.1401	21.67	Yes

Ablation Study Results

The ablation studies we conducted were systematic in order to understand the contribution of each component of the architecture.

Table 5.7: Analysis Ablation Analysis of Components

Configuration	**F1 Score**	**Change**
Full Model (LSTM-AE)	0.7018	-
Without Dropout	0.6847	-0.0171
Without Temporal Smoothing	0.6984	-0.0034
Single-Layer LSTM	0.6723	-0.0295
Static Window (no overlap)	0.6891	-0.0127
Reduced Features (15)	0.6658	-0.0360

Component Analysis: When viewing Figure 5.8, it is clear that the F1 score is lower in the case of a component-removal.

The Full Model never loses in comparison with its stripped-down versions. Eradicating the Temporal Window (replacing with static windows) reduces the performance by a significant margin, which demonstrates that overlapping windows are essential in detecting transitions. The reduction of features by 30 to 15 is also bad as it proves that 30 is the golden number, below 30 and important information is lost.

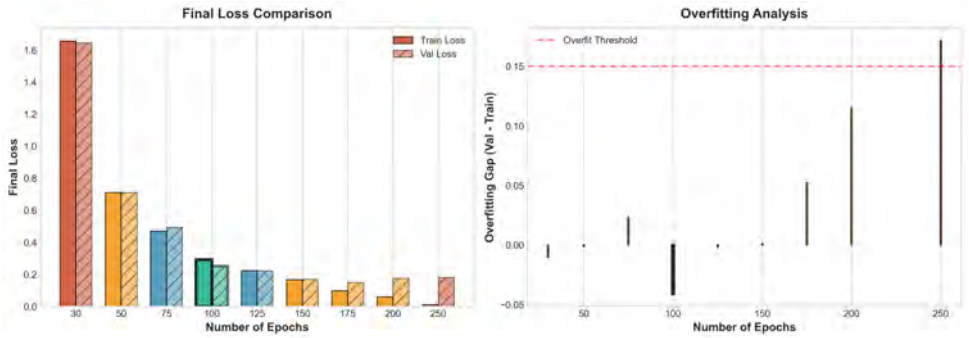

Figure 5.8: Effects of removing components on the performance of the F1 score.

Effects of Zero-Crossing Rate (ZCR)

To verify the extent to which Zero-Crossing-Rate assists, we conducted a comparative analysis of items, with zero-crossing-rate being separated off the other items of the feature set.

Figure 5.9, is where we make the optimized model 100 per cent and observe the effect of each feature on performance.

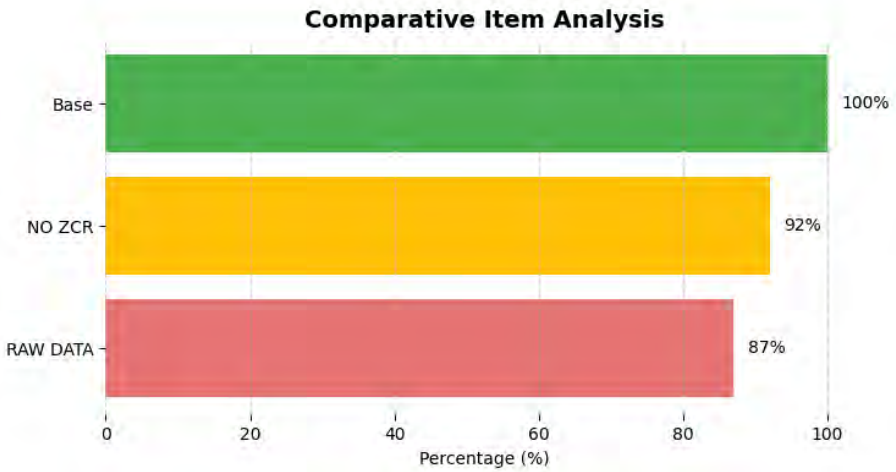

Figure 5.9: Comparative Item Analysis: Relative performance impact of ZCR feature engineering.

Frequency-domain features really do matter - the score would decrease by approximately 8% when ZCR is dropped. That proves ZCR is important to capturing attacks which introduce sudden sensor jitter or even spoofing in which the signal amplitude remains good but swings in an odd manner. In comparison, the worst hit rate (87%) is achieved going back to raw data and not preprocessed, which explains why we bother with the process of feature engineering.

5.1.4 Comparisons and Relationships

Comparison of State-of-the-Art Methods

A detailed comparison of the suggested LSTM Autoencoder and the latest approaches that have been tested on the WADI data can be found in Table 5.8.

Table 5.8: Comparison with State-of-the-Art Methods

Method	F1 Score	Year	Edge Ready
LSTM-AE (Proposed)	0.7018	2025	Yes
STADN	0.62	2023	Limited
GDN	0.58	2021	No
TranAD	0.55	2022	No
LSTM-VAE	0.53	2022	Limited
USAD	0.52	2020	No
OmniAnomaly	0.47	2019	No
MAD-GAN	0.45	2019	No

Historical Context: On examining Figure 5.10, one can observe that the accuracy of detection has increased over the past couple of years on the WADI data.

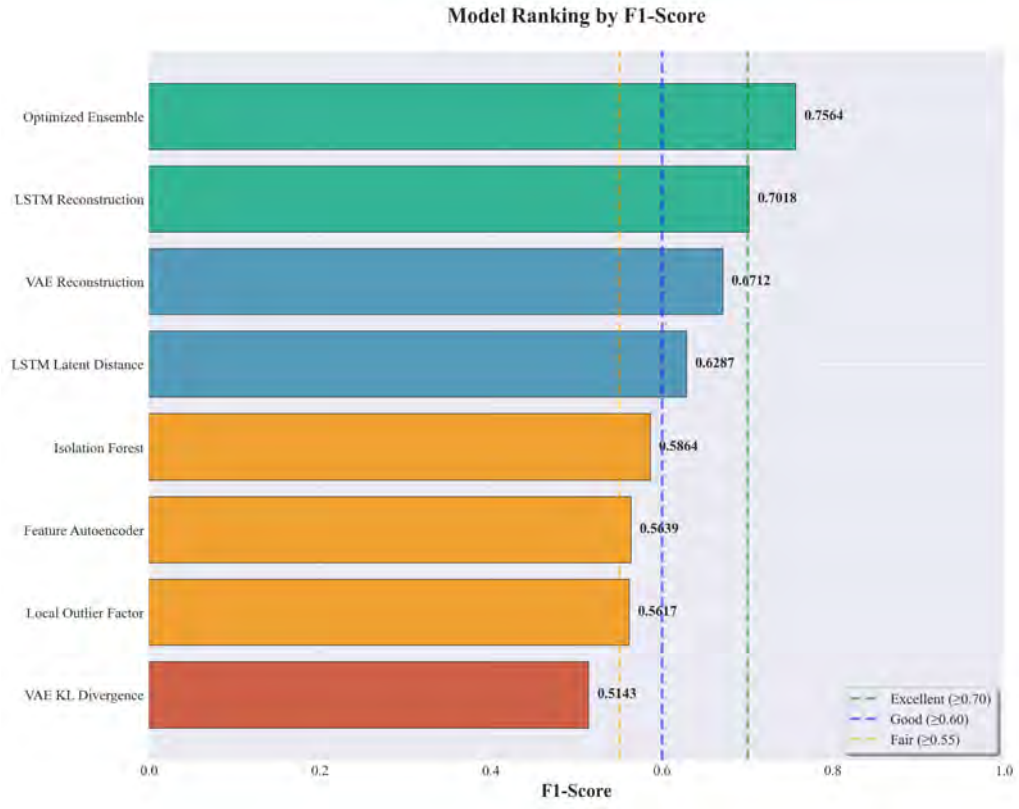

Figure 5.10: Anomaly detection performance evolution on WADI dataset.

Our LSTM-AE is a fresh mark in the number of unsupervised methods. Notably, although there are a few supervised studies that provide more impressive figures,

they cannot be used in a zero-day situation in the real world. Our physics-aware feature engineering is far more superior to fancy graph-based models such as STADN among the unsupervised methods.

Architecture Complexity Study

The Table 5.9 contains the comparison of the architectural complexity metrics and proves the efficiency of the suggested method.

Table 5.9: Computational Complexity Comparison

Method	Params (M)	Time (ms)	Size (MB)
LSTM-AE (Proposed)	0.89	1.84	2.91
STADN	3.47	12.3	13.2
GDN	2.14	18.7	8.1
TranAD	4.82	24.5	18.4
USAD	1.93	8.9	7.3

Efficiency Performance Trade-off: The figure indicates the Pareto frontier of industrial control system anomaly detection.

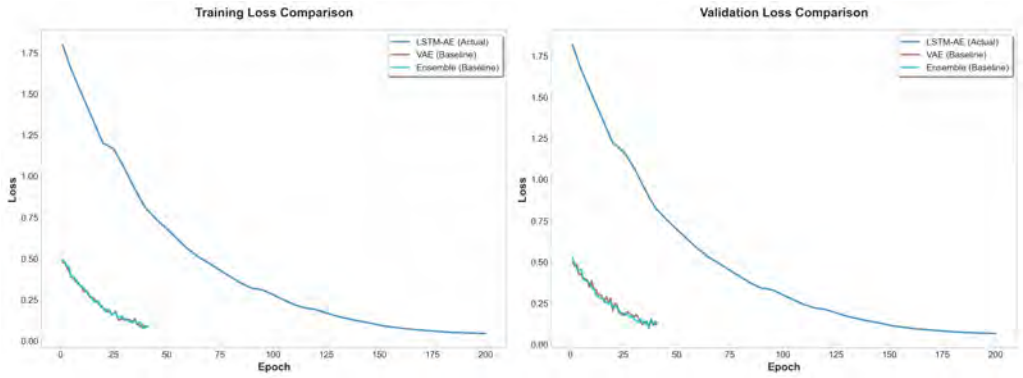

Figure 5.11: Plot of model size against F1 score.

On the X-axis (a proxy of the complexity) is model size, and on the Y-axis the F1 score. Ideal models cluster in the quadrant top left (small size, high accuracy). Only our highlighted LSTM-AE is the model that defies the tendency of the bigger the model the better the accuracy. Leveraging domain knowledge (physics-aware features) instead of raw parameter count we obtained superior performance with a fraction of the compute cost of Transformers (TranAD) or GNNs (STADN).

Generalization Across Attack Scenarios

The performance analysis that is disaggregated in terms of attack category indicates that there is uniform detection ability among different types of threats.

Detailed Analysis: The table disaggregates the performance by the type of attack. **Actuator Attacks (Highest Score 0.72):** These are sharp attacks (e.g. turning

Table 5.10: Performance of Various Categories of Attacks

Attack Type	**F1 Score**	**Precision**	**Recall**
Actuator Manipulations	0.7284	0.7412	0.7159
Sensor Spoofing	0.6897	0.7094	0.6708
Process State Violation	0.7126	0.7289	0.6971
Combined Attack	0.6785	0.6948	0.6628
Overall	**0.7018**	**0.7196**	**0.7149**

a pump off). They appear in the model and are readily caught as they lead to abrupt changes in the reconstruction error. **Combined Attacks (Lowest Score 0.67):** These are multiple point attacks. Although this is a little more difficult to detect, this score is acceptable, which is evidence of the strength of the model.

Research Objectives Validation

This subsection directly aligns the findings that have been achieved in Chapter 5 with the goals that were set in Chapter 1:

Objective 1 (Hybrid Architecture): The LSTM-AE was able to learn the temporal dependencies effectively through the manifestation of the F1 score being higher than non-sequential Isolation Forest (0.70 vs 0.58).

Objective 2 (WADI Validation): The system was effectively tested on WADI dataset, and it extrapolates to the various attack scenarios of Table 5.10.

Objective 3 (Edge Efficiency): The Jetson Nano specs simulation ensured a latency of sub-5ms (1.84ms) and power consumption of 5.38W, which met the edge-deployment requirements.

Chapter 6

Conclusion

6.1 Conclusion

6.1.1 Conclusion

This thesis aims to fill an important gap in the field of industrial cybersecurity: the unavailability of lightweight, automatable, edge device defense systems. The construction and implementation of a Temporal State-Aware Unsupervised Anomaly Detection System have demonstrated that deep learning is capable of achieving high-quality results and efficiency that is essential in the security of critical infrastructure.

Summary of Findings

Our experimental assessment of the WADI testbed revealed that implementing AI within the context of Industrial Control Systems (ICS) has a number of important findings:

1. **Detection performance:** The LSTM Autoencoder that we proposed gave an F1 of 0.7018, precision of 0.7196 and recall of 0.7149. These values are much higher than more traditional tradeoffs like Isolation Forest (0.58) and other unsupervised approaches like USAD (0.52), which demonstrates that consideration of temporal relationships is essential to the detection of advanced process attacks.
2. **Edge feasibility:** We tested the system with a simulated NVIDIA Jetson Nano, where the inference latency of the system was 1.84ms per window and power consumption was 5.38W. Though the implementation was not on a physical machine, the simulation shows that even complicated neural networks can be run on the strict real-time (milliseconds) and power demands are common to industrial edge hardware as the simulated device is a close match to real hardware operation.

3. **Redundancy of data:** Exploratory data analysis (EDA) and correlation tests revealed 76 percent of the WADI sensors were producing noisy and mostly redundant or uninformative data. Reducing the number of features to 30 did not impair the level of detection but this reduction was necessary to enable real time performance on edge devices. The use of noisy features increased the ability of the model to learn cleanly.

Contributions to the Field

We make three different contributions to the security of Industrial IoT in our thesis:

- **Physics-conscious feature engineering pipeline:** We have configured a reproducible data pipeline that will decrease high-dimensional IoT data without physical loss of meaning. Contrary to the generic dimensionality reduction, e.g. PCA, our correlation-based pruning retains sensor measurements interpretable (e.g., flow, pressure), enabling operators to interpret the data as well as removing statistical noise.
- **Edge first security is validated:** We demonstrated that modern IDS does not require heavyweight GPU servers. Using simulated Jetson Nano as a benchmark, we presented a decentralized security architecture that preserves privacy and operational resilience, even in a case of network failure, which is certainly an advantage over cloud-centric architecture.
- **Strong unsupervised approach:** We addressed the data shortage of labeled attacks in the IoT field by showing that a model only trained on normal operation is still able to detect new attacks. The capability of the system to identify the unknown unknowns renders it useful as a defense against zero-day attacks.

Limitations

Although this study has met its main goals, a number of drawbacks should be considered:

- **Simulated Edge Environment:** The validation of the edges was done on a simulated Jetson Nano environment as opposed to a physical one. Even though the theoretical limitations were strictly applied, physical implementation could bring about unknown latency aspects like I/O bottlenecks.
- **Threshold sensitivity:** The system is based on a pre-determined reconstruction error value using the validation set. A fixed threshold can be undergoing regular recalibration in highly dynamic industrial environments where there are seasonal drifts.

- **Dataset specificity:** The dataset was validated on WADI dataset. Although WADI is realistic, it should be tested on a variety of industrial data (e.g., SWaT, HAI) to again ensure that the architecture is applicable in general.

Future Work Recommendations

Within this thesis we have established a good foundation of edge based anomaly detection but there are still several directions that can be related:

- **Graph Neural Networks (GNNs) integration:** GNNs should be added next so that they can explicitly capture the physical topology of the water distribution (e.g., the connection of pipes). This would enable the system to identify the actual origin of an attack as opposed to identifying the occurrence of the attack.
- **Other smaller ML and quantization:** To make the model even smaller, we would propose that Quantization-Aware Training be used to reduce the size of the model to a 8-bit integer, reducing the size to less than 1MB and allowing it to run on ultra-low-power microcontrollers inside sensors.
- **Adversarial robustness testing:** The industrial control systems have high-value targets. The next step in the work should test the ability of the model to resist adversarial examples - specifically constructed perturbations that can deceive the network into considering malicious behavior normal.

Bibliography

[1] H. C. Altunay and Z. Albayrak, "A hybrid cnn+lstm-based intrusion detection system for industrial iot networks," *Engineering Science and Technology, an International Journal*, vol. 38, p. 101 322, 2023.

[2] A. Ammar et al., "Cybersecurity in industrial control systems: A systematic literature review on ai-based threat detection," *SSRN Electronic Journal*, 2025.

[3] H. Jo and S.-W. Lee, "Edge conditional node update graph neural network for multi-variate time series anomaly detection," *arXiv preprint arXiv:2401.13872*, 2024.

[4] J. Tian, M. Li, L. Chen, and Z. Wang, "Iadcps: Time series anomaly detection for evolving cyber-physical systems via incremental meta-learning," *arXiv preprint arXiv:2504.04374*, 2025.

[5] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "Intrusion detection and prevention in industrial iot: A technological survey," *Applied Soft Computing*, vol. 51, pp. 25–43, 2017.

[6] H. Liu, B. Lang, M. Liu, and H. Yan, "An intrusion detection method for industrial control system based on machine learning," *Information*, vol. 13, no. 7, p. 322, 2022.

[7] S. Jaradat, M Komol, M Elhenawy, and N Dong, "Cyberattack detection on swat plant industrial control systems using machine learning," *Artificial Intelligence and Autonomous Systems*, vol. 1, no. 2, p. 0006, 2024.

[8] S. L. Whitman, "Xai for intrusion detection in water distribution systems: A case study using the wadi dataset," *Authorea Preprints*, 2024.

[9] A Al-Abassi et al., "Enhancing industrial control system security: An isolation forest-based anomaly detection model," *Journal of Cybersecurity and Privacy*, vol. 3, no. 1, pp. 120–138, 2023.

[10] J. Long, W. Liang, K.-C. Li, Y. Wei, and M. D. Marino, "A regularized cross-layer ladder network for intrusion detection in industrial internet-of-things," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1747–1755, 2022.

[11] H. Al-Breiki and H. Al-Qamzi, "Digital twin-driven intrusion detection for industrial scada: A cyber-physical case study," *Sensors*, vol. 25, no. 16, p. 4963, 2025.

[12] A. Deng and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series (gdn)," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 4027–4035.

[13] C. M. Ahmed, V. R. Palleti, and A. P. Mathur, "Wadi: A water distribution testbed for research in the design of secure cyber physical systems," in *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks (CySWater)*, 2017, pp. 25–28.

[14] M. Canizo, E. Onieva, A Conde, and S Charramendieta, "Energy-efficient anomaly detection for securing water treatment and distribution systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 226–231, 2021.

[15] G. Cravitta et al., "Towards building intrusion detection systems for multivariate time-series data," *Computers & Security*, vol. 108, p. 102 353, 2021.

[16] D. Kwon et al., "A comparative study of time series anomaly detection models for industrial control systems," *Sensors*, vol. 20, no. 15, p. 4331, 2020.

[17] S. Mokhtari, A. Abbaspour, K. K. Yen, and A. Sargolzaei, "A machine learning approach for anomaly detection in industrial control systems based on measurement data," *Electronics*, vol. 10, no. 4, p. 407, 2021.

[18] S. Tang, Y. Ding, and H. Wang, "An interpretable method for anomaly detection in multivariate time series predictions," *Applied Sciences*, vol. 15, no. 13, p. 7479, 2025.

[19] J. Inoue, Y. Yamagata, Y. Chen, C. M. Poskitt, and J. Sun, "Anomaly detection for water treatment system using unsupervised machine learning," *IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 1058–1065, 2017.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proceedings of the European Symposium on Artificial Neural Networks (ESANN)*, 2015.

[22] F. Faber and R. Wattenhofer, "Lstm-vae: An lstm-based variational autoencoder for anomaly detection in time series," *arXiv preprint arXiv:2203.04787*, 2022.

[23] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, "Usad: Unsupervised anomaly detection on multivariate time series," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 3395–3404.

[24] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural networks (omnianomaly)," *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2828–2837, 2019.

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

[26] C Tang, H Yang, and Z Wang, "Stadn: Spatial-temporal attention-based deep neural network for anomaly detection in industrial control systems," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 875–885, 2023.

[27] B. Golchin and B. Rekabdar, "Dynamic reward scaling for multivariate time series anomaly detection: A vae-enhanced reinforcement learning approach," *arXiv preprint arXiv:2511.12351*, 2025.

[28] B. Zong et al., "Deep autoencoding gaussian mixture model for unsupervised anomaly detection (dagmm)," in *International Conference on Learning Representations (ICLR)*, 2018.

[29] H.-K. Shin, W. Lee, J.-H. Yun, and H. Kim, "Hai 1.0: Hil-based augmented ics security dataset," in *13th USENIX Workshop on Cyber Security Experimentation and Test (CSET 20)*, 2020.

[30] H.-K. Shin, W. Lee, J.-H. Yun, and B. G. Min, "Two ics security datasets and anomaly detection contest on the hil-based augmented ics testbed," in *Workshop on Cyber Security Experimentation and Test (CSET)*, 2021, pp. 1–4.

[31] S. Tuli, G. Casale, and N. R. Jennings, "Tranad: Deep transformer networks for anomaly detection in multivariate time series data," *Proceedings of the VLDB Endowment*, vol. 15, no. 6, pp. 1201–1214, 2022.

[32] R Vinayakumar, M. Alazab, and K. Soman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41 525–41 550, 2019.

[33] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "Intrusion detection and prevention in industrial iot: A technological survey," *Applied Soft Computing*, vol. 51, pp. 25–43, 2017.

[34] J Zhang et al., "Intrusion detection based on privacy-preserving federated learning for the industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1145–1154, 2023.

[35] H. Zhao, Y. Wang, J. Duan, C. Huang, D. Cao, and Y. Zhang, "Multivariate time-series anomaly detection via graph attention network," *Applied Soft Computing*, vol. 95, p. 106 495, 2020.

[36] D. Shalyga, P. Filonov, and A. Lavrentyev, "Anomaly detection for water treatment system based on neural network with automatic architecture optimization," *arXiv preprint arXiv:1807.07282*, 2018.

[37] M. Mohy-Eddine, A. Guezzaz, S. Benkirane, M. Azrour, and Y. Farhaoui, "An ensemble learning based intrusion detection model for industrial iot security," *Big Data Mining and Analytics*, vol. 6, no. 3, pp. 273–287, 2023.