Dynamic time warping based anomaly detection for industrial control system

Albert Zariqi, Souad Asroubi

December 2023

1 Introduction

In this research, we aim to build a model for detecting anomalies in Industrial Control Systems. Industrial Control Systems (ICS) find extensive application across various industries, commonly incorporating diverse sensors and embedded systems within a dedicated network for operation.

ICS have transitioned from being isolated systems to interconnected frameworks that leverage contemporary communication technologies and protocols. This shift aims to enhance efficiency, reduce operational costs, and further enhance an organization's support model. These solutions, however, face the risk of cyber-physical attacks on their network. Although system security is a thoroughly researched and documented field, there is still a possibility for attacks to go undetected. Anomaly detection is the identification of rare, suspicious deviations from standard patterns in data. These anomalies can be referred to as outliers, noise, novelties, or exceptions. It plays a crucial role in identifying issues like hacking, fraud, equipment malfunctions, and errors.

There are three main classes of anomaly detection techniques: unsupervised, semi-supervised, and supervised. The choice of method depends on the availability of labeled data. Supervised techniques require a dataset with labeled 'normal' and 'abnormal' instances, involving training a classifier, but handling the class imbalance can be challenging. Semi-supervised methods use labeled data to create a model of normal behavior and assess anomalies based on the model's likelihood. Unsupervised methods detect anomalies in unlabeled data, assuming the majority of instances are normal, and identifying the least congruent instances. Anomaly detection is essential for identifying rare and suspicious occurrences, and the choice of method depends on the dataset's labeling. [1][4][6]

Anomalies can be classified in several ways: **Network anomalies, Application performance anomalies and Web application security anomalies**. Network anomalies - These deviations from normal network behavior require continuous monitoring for unexpected trends or events to detect and address. Application performance anomalies - Detected through end-to-end application

performance monitoring, these anomalies trigger rate limiting and alert administrators to the source of issues. Web application security anomalies - These encompass anomalous or suspicious web application behaviors like CSS attacks or DDoS attacks that can impact security. [1]

Anomalies exhibit various types based on their characteristics [8]. They can be categorized into three common types: point anomalies, contextual anomalies, and collective anomalies. Point anomalies signify individual data points significantly differing from the dataset norm, contextual anomalies represent data points unusual within specific contexts, and collective anomalies emerge as groups of data points deviating when considered together. However, complexities arise when anomalies exhibit characteristics of multiple types, challenging precise identification.

Moreover, in this research, we focus on leveraging Dynamic Time Warping (DTW) as a crucial component in the anomaly detection process. DTW, a distance measure algorithm, is particularly adept at comparing and aligning time series data with varying lengths or temporal distortions. By integrating DTW into our proposed methodology, we aim to enhance the model's ability to capture complex temporal patterns and irregularities present in Industrial Control Systems data. DTW's capability to account for temporal shifts and variations in sensor readings makes it a valuable tool in distinguishing normal operational behavior from anomalies, contributing significantly to the efficacy of our anomaly detection framework. [3]

2 Related work

Considerable research has been conducted on the security concerns associated with Industrial Control Systems (ICS). Of the numerous methods available for detecting abnormal data, the Long Short Term Memory (LSTM) network stands out as a prominent and widely adopted approach.

This paper [13] introduces a composite autoencoder neural network model. The primary objective of this model is to capture the patterns in ICS operational data when operating under normal conditions. The model accomplishes this by working with data exclusively from normal operating conditions. It conducts simultaneous prediction and reconstruction of the original input time series, and then applies the error derived from both processes to classify observations of multi-variable time series as either anomalous or normal. In cases of anomalies, the change ratio is utilized to identify the variables that have been affected by an attack. The main architecture used consists of LSTM autoencoder which is divided in two parts: the encoder and the decoder using LSTM as building blocks.

The authors in [18] propose two unsupervised machine learning methods, Deep Neural Network (DNN) and Support Vector Machine (SVM), for anomaly detection in a water treatment system [18]. The authors use the SWaT dataset,

which consists of network traffic, sensor data, and actuator data collected over 11 days of continuous operation. The DNN approach includes a layer of Long Short-Term Memory (LSTM) architecture followed by feedforward layers, while the SVM is a commonly used technique for anomaly detection. The performance of the proposed methods is evaluated in terms of precision and recall scores of detected anomalies in the attack log. The DNN approach shows slightly better overall F-measure and precision, while the SVM has slightly better recall.

In this paper [17], the authors introduce a novel hybrid algorithm known as VQ-OCSVM, which combines Vector Quantization (VQ) and the One-Class Support Vector Machine (OCSVM) method to enhance anomaly detection. Another method [6] uses DTW to compare the distance between control plane and data plane, and if it falls within certain threshold is considered as normal, otherwise is considered as anomaly.

Recent advancements in industrial control system security include a study focusing on a dual-isolation-forests-based (DIF) attack detection framework [19]. This approach, applied to the SWaT and WADI testbeds, not only enhanced attack detection efficiency but also achieved a notable increase in F1-score, particularly for the SWaT testbed. However, it encountered challenges in precision-recall trade-offs and was less effective in complex adversarial attack scenarios, with some attacks remaining undetected.

In parallel, another research initiative utilized unsupervised machine learning, with an emphasis on deep learning techniques like Neural Networks and RNNs [20], to detect anomalies in water treatment systems. Despite its effective pattern recognition capabilities, this method faced limitations, such as overfitting and a susceptibility to adversarial attacks, which could hinder its practical application in real-world industrial control system environments.

In [23], the author present a significant contribution to the field of cybersecurity for Industrial Internet of Everything (IIoE) systems. As the fourth industrial revolution propels industrial control systems (ICSs) into the realm of IIoE, the authors address the escalating sophistication of cyber threats, including attacks that adeptly mimic normal network traffic. Their proposed solution, the Autoencoder-based Payload Anomaly Detection (APAD) method, harnesses the power of deep learning and autoencoders to discern between normal and abnormal behaviors, with a specific focus on the vulnerabilities associated with low-performance field devices. Through a meticulous classification of the extended reference model RAMI 4.0 into operative and product process management levels, the authors tailor the APAD method to each level's unique characteristics. Evaluation using the SWaT dataset showcases the effectiveness of APAD, surpassing other examined methods and offering a promising avenue for real-time anomaly detection in ICS networks.

Given the identified limitations and critical drawbacks in existing approaches, our approach, integrating Dynamic Time Warping (DTW) with autoencoders (Autoencoder and LSTM Autoencoder), aims to overcome limitations in prior

anomaly detection methods. By leveraging DTW's ability to compare time-series sequences and autoencoders' pattern learning capabilities, we aim to enhance anomaly detection in Industrial Control Systems (ICS). This fusion seeks to create a robust framework tailored for ICS, capable of accurately identifying anomalies amidst complex data structures. Our goal is to fortify the security of critical infrastructure by offering a more effective solution for anomaly detection in ICS environments.

2.1 Time Series Anomaly Detection

Time series data refers to a sequence of data points that are collected over time intervals. Anomaly detection in time series data has attracted much attention due to its wide applications in various fields, including finance, healthcare, and manufacturing. [10] There are several techniques that can be used for anomaly detection in time series data, including statistical methods, machine learning algorithms, and deep learning models. Statistical methods, such as moving average, exponential smoothing, and ARIMA, are commonly used for anomaly detection in time series data. Machine learning algorithms, such as decision trees, random forests, and support vector machines (SVMs), have also been applied for anomaly detection in time series data. These algorithms typically require a set of labeled data to train the model and detect anomalies. Recently, deep learning models, such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs), have shown promising results for anomaly detection in time series data. These models can capture complex patterns in the time series data and identify anomalies with high accuracy. [11]

2.2 Uni and Multivariate Time Series Anomaly Detection

Time series data is a common form of data in various fields such as finance, weather forecasting, and industry. Anomaly detection in time series data can be categorized into two main types: univariate and multivariate. Univariate anomaly detection is the process of identifying anomalies in a single time series data, while multivariate anomaly detection aims to detect anomalies in multiple time series data. There are various techniques used for both univariate and multivariate time series anomaly detection, including statistical methods, machine learning algorithms, and deep learning models. Each of these techniques has its advantages and limitations, depending on the characteristics of the time series data and the specific anomaly detection task.[12]

3 Dataset

We used the Secure Water Treatment (SWaT) testbed dataset[15], which is a real-world dataset collected from a water treatment plant located in Singapore. The dataset was collected on December 22, 2015, and includes measurements from 51 sensors and actuators for a total of 11 days. The SWaT testbed was

specifically designed to simulate cyber-physical attacks on a water treatment plant, making it an ideal dataset for evaluating anomaly detection methods. The system operated non-stop from its "empty" state to fully operational state for a total of 11 days. During the first seven days, the system operated normally without any attacks or faults. In the remaining days, cyber and physical attacks were launched on the SWaT while data collection continued. The dataset includes both normal and attack scenarios, where the attack scenarios simulate various cyber-physical attacks on the water treatment plant. The attacks were carefully designed to be realistic and to mimic the behavior of real-world attackers. The dataset is publicly available and has been used extensively in the research community for evaluating anomaly detection methods. [5] In total there are 36 attacks, and they are divided into four types:

- Single Stage Single Point (SSSP): A Single Stage Single Point attack focuses on exactly one point in a CPS. (26 attacks)
- Single Stage Multi Point (SSMP): A Single Stage Multiple Point attack focuses on two or more attack points in a CPS but on only one stage. In this case set, P consists of more than one element in a CPS selected from any one stage. (4 attacks)
- Multi Stage Single Point (MSSP): A Multi Stage Single Point attack is similar to an SSMP attack except that now the SSMP attack is performed on multiple stages. (2 attacks)
- Multi Stage Multi Point (MSMP): A Multi Stage Multi Point attack is an SSMP attack performed in two or more stages of the CPS. (4 attacks)

4 Baseline

4.1 Time Series Anomaly Detection

In [4], the authors detect cyber-attacks in Cyber-Physical Systems (CPS) through unsupervised learning, employing Long Short-Term Memory Recurrent Neural Networks (LSTM-RNN) and Cumulative Sum (CUSUM) methods. They propose a model using LSTM-RNN to predict sensor data based on historical data. Deviations between actual and predicted data are calculated, and CUSUM is used for anomaly detection. The evaluation is on a dataset from a Secure Water Treatment (SWaT) testbed, revealing the method's ability to detect 9 out of 10 attacks, with some false positives and unexplained anomalies. Techniques for both include statistical methods, machine learning, and deep learning models. Each technique has its strengths and limitations, depending on the data's characteristics and the specific detection task. The author concentrates on utilizing machine learning and deep learning models for both univariate and multivariate time series anomaly detection [12].

4.2 Baselines

In this section, we thoroughly examine the effectiveness of three baseline methodologies ,Isolation Forest, implemented as detailed in [19] , LSTM, as demonstrated in [13], and Autoencoder, applied as outlined in [23] for detecting anomalies in the SWaT dataset. These methodologies, including Isolation Forest, LSTM, and Autoencoder, function as essential benchmarks, offering a foundational basis to assess the performance of our proposed approach.

4.2.1 Isolation Forest

Isolation Forest, introduced by Fei Tony Liu in 2008, represents a significant advancement in the field of anomaly detection. The algorithm's inception was driven by the need for an efficient and effective method to identify anomalies within datasets. [22]

In the mathematical realm, Isolation Forest leverages fundamental concepts from decision trees and randomization to achieve efficient anomaly detection. The algorithm's strength lies in its ability to identify anomalies by isolating them with fewer partitions in a recursive, randomized process. The mathematical foundation involves constructing isolation trees, where each split is based on randomly selected attributes. The length of paths within these trees serves as a crucial metric for scoring anomalies.

Notably, the algorithm's linear time complexity and low memory requirement make it well-suited for high-dimensional datasets. However, it may struggle with structured data, anomalies sharing similarities with normal data, and can face challenges in handling imbalanced data. The approach demonstrates effectiveness in sub-sampling and is less affected by the presence of anomalies in training data. .The provided pseudocode for Isolation Forest encapsulates the algorithmic steps 1, offering a concise reference for its implementation and showcasing the elegance of its mathematical formulation.

Algorithm 1: iForest(X, t, ψ)

Input: X - input data, t - number of trees, ψ - subsampling size
Output: a set of t iTrees

```
Initialize Forest;
Set height limit l = ⌈log_2 ψ⌉;
for i = 1 to t do
    X_0 ← sample(X, ψ);
    Forest ← Forest ∪ iTree(X_0, 0, l);
end
return Forest;
```

The Isolation Forest algorithm was employed as a baseline for anomaly detection. We utilized grid search to optimally tune the model, ultimately setting the contamination factor at 0.01. The model, trained on the training set, was

then used to predict anomalies on the test set. Predictions were translated into a binary format, indicating anomalies.

4.2.2 Autoencoder

The autoencoder is a neural network architecture employed for unsupervised learning tasks, particularly in dimensionality reduction and feature learning. Its mathematical foundation involves an encoder and decoder structure. The encoder maps input data into a lower-dimensional representation, typically using linear transformations followed by non-linear activation functions. The decoder reconstructs the input from this compressed representation, minimizing the reconstruction error through the training process. The optimization objective often involves minimizing a chosen cost function, such as Mean Square Error (MSE). Autoencoders leverage linear algebra concepts, including matrix rank and transformations, where the encoder and decoder weight matrices play a crucial role. Training involves adjusting these weights to achieve a compact representation that captures essential features of the input data, making autoencoders versatile tools for capturing intrinsic data patterns in an unsupervised manner.

What sets the autoencoder apart is its adaptability to various data types and robustness in capturing intricate relationships within the input data. This makes it a versatile tool for unsupervised learning, providing a nuanced understanding of data patterns and enhancing anomaly detection across diverse datasets. In summary, the traditional autoencoder's strength lies in its capacity for unsupervised learning, enabling it to uncover meaningful representations of data and effectively identify anomalies based on reconstruction errors.

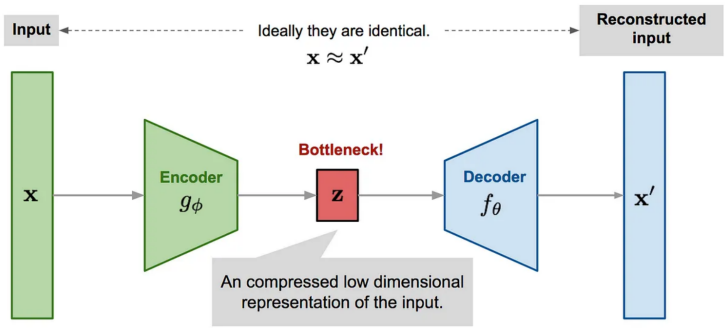

Figure 1: Architecture of AutoEncoder

4.2.3 LSTM

Long Short Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber (1997), and were refined and popularized by many

people in following work.1 They work tremendously well on a large variety of problems, and are now widely used. LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn! All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer. [21]

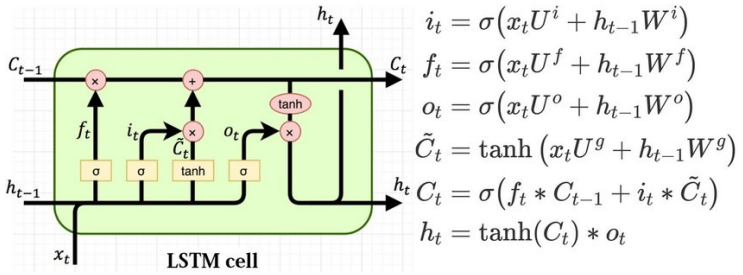

Figure 2: Architecture of LSTM cell

LSTM is a type of RNN through which the vanishing gradient problem is sufficiently reduced to make recurrent neural networks more useful for tasks requiring long-term memory. This is achieved by creating an internal memory state, which is added to the processed input, which reduces the multiplier effect of small gradients. Unlike traditional neural networks, which contains neurons, LSTM contains memory cells that are connected between layers. LSTM cells manage two state vectors (c(t) and h(t)), where for performance reasons these vectors are kept separate in the general case. C(t) is known as the long-term state, while h(t) is known as the short-term state. The idea is that the network learns what to keep in the long term, what to remove, and what to read from it. As seen from Fig. 2, the long-term state c(t-1) traverses the network from left to right, it is seen that it first passes through the forget gate, removing some data from memory, and then adds new data through the addition operation (which adds the data that is selected by the input gate). [21]

Considering the benefits of the LSTM model, we were focused on the paper [13] that implements LSTM autoencoder. The autoencoder contained encoder and decoder. The encoder part has two LSTM layers, first layer with 64 neurons and second layer with 32 neurons using ReLU as activation function. The decoder part also has two LSTM layers, where first layer with 32 neurons and second layer with 64 neurons using ReLU as activation function. Adam optimizer was employed with 5 epochs.

4.3 Proposed methodology

In this paper, we propose enhancing the baseline LSTM implementation through the integration of Dynamic Time Warping (DTW). The initial phase involves training the LSTM model on the dataset, establishing a baseline for comparison. Subsequently, we introduce DTW as a complementary technique to capture temporal dependencies more effectively. The DTW algorithm will be applied to align and measure the similarity between time series sequences, providing an additional layer of refinement to the LSTM's predictive capabilities. The combined model is then fine-tuned through an iterative process to optimize its performance. The proposed methodology aims to leverage the strengths of both LSTM and DTW, offering a more robust and accurate solution for time series analysis. Architecture of the system is shown below:

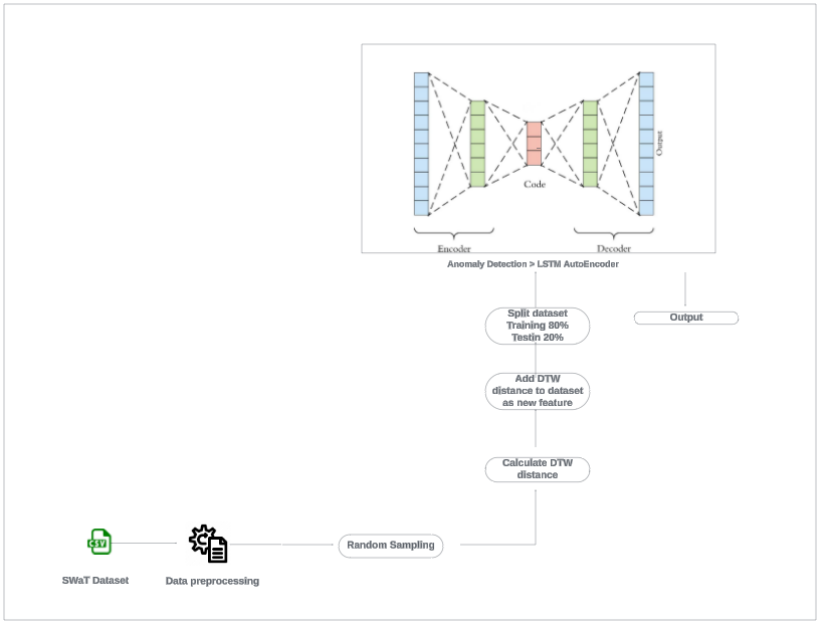

Figure 3: System architecture (LSTM)

As a secondary approach, an alternative technique involves integrating Dynamic Time Warping (DTW) into the initial stages of a traditional autoencoder's preprocessing phase. DTW plays a pivotal role in refining the representation of sequential data before it enters the autoencoder model. This collaboration of DTW with the preprocessing phase serves as a critical step in enhancing the model's ability to capture anomalies within the SWaT dataset. Leveraging DTW's capability to handle time-series data of varying lengths and temporal distortions, this method aims to enhance the autoencoder's proficiency in anomaly detection. The incorporation of DTW facilitates the alignment and transformation of sequential data, allowing the autoencoder to capture subtle

irregularities or temporal variations that may indicate anomalous patterns in the industrial system data. This fusion of DTW into the initial stages of preprocessing aims to fortify the autoencoder's performance in identifying anomalies by providing a more robust and enriched representation of the sequential data.

Furthermore, we undertook a comparable analysis integrating Isolation Forest alongside DTW, following the same architecture outlined below. This evaluation aimed to assess the combined performance of Isolation Forest and DTW and to measure their impact on anomaly detection within the SWaT dataset. We specifically compared this combined approach with our baseline, which involves using Isolation Forest alone.

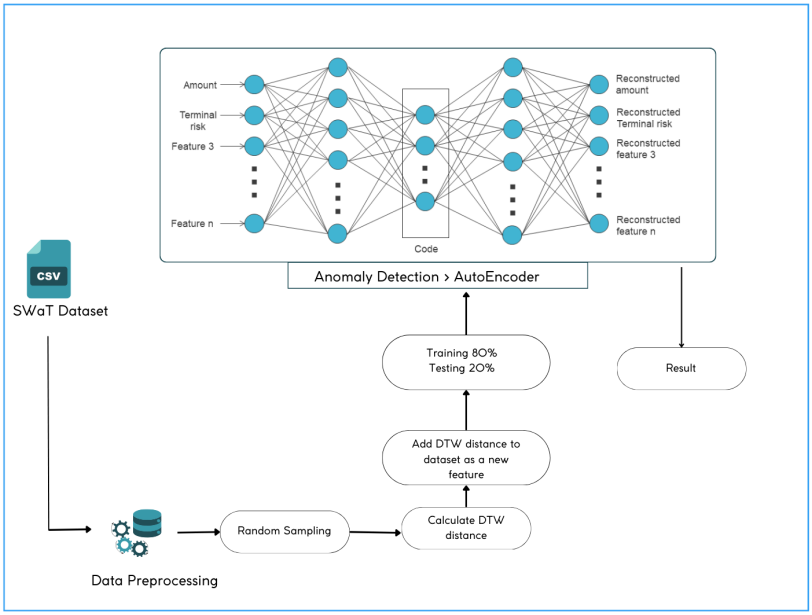

Figure 4: System architecture (AutoEncoder)

Below is the implementation of the basic Dynamic Time Warping (DTW) algorithm. This algorithm computes the DTW distance between two time series arrays, s and t, effectively capturing temporal dependencies. The pseudocode for the DTW algorithm is provided in Algorithm 2. The distance is calculated by aligning the sequences and considering the minimum cumulative cost. The application of DTW enhances the model's ability to analyze time series data, providing a foundation for subsequent discussions on its integration into other algorithms

Algorithm 2: Dynamic Time Warping (DTW) Distance Calculation

```
function DTWDistance(s: array [1..n], t: array [1..m])
    Data: Arrays s and t of lengths n and m
    Result: DTW distance between s and t
    DTW := array[0..n, 0..m];
    for i := 0 to n do
        for j := 0 to m do
            DTW[i, j] := ∞;
    DTW[0, 0] := 0;
    for i := 1 to n do
        for j := 1 to m do
            cost := d(s[i], t[j]);
            DTW[i, j] := cost + minimum(DTW[i − 1, j], DTW[i, j −
              1], DTW[i − 1, j − 1]);
    return DTW[n, m];
```

4.4 Data Preprocessing

In the process of preparing the SWaT dataset for anomaly detection, several crucial data preprocessing steps were applied, incorporating both feature engineering and data transformation to ensure the dataset's suitability for robust analysis. These steps aimed to enhance the dataset's quality, handle missing values, eliminate irrelevant information, and standardize the data. The SWaT dataset contains categorical data, such as equipment identifiers.

To incorporate these categorical features 2 into the analysis, we employed one-hot encoding. This technique converts categorical variables into binary (0 or 1) vectors, allowing the algorithms to interpret these features effectively. To maintain uniformity in the dataset and ensure that all numerical features 1 have the same scale, min-max scaling was applied. This process scales the values of numerical features to a specific range, usually between 0 and 1. Normalization enhances the performance of the machine learning model by preventing features with larger scales from dominating the analysis.

Numerical Features
FIT101, LIT101, AIT201, AIT202, AIT203, FIT201, DPIT301, FIT301, LIT301, AIT401, AIT402, FIT401, LIT401, AIT501, AIT502, AIT503, AIT504, FIT501, FIT502, FIT503, FIT504, PIT501, PIT502, PIT503, FIT601

Table 1: Numerical Features

Features lacking variation across the dataset were eliminated from consideration as they did not contribute valuable information to the anomaly detection

Categorical Features
MV101, P101, P102, MV201, P201, P203, P204, P205, P206, MV301, MV302, MV303, MV304, P301, P302, P402, P403, UV401, P501, P602

Table 2: Categorical Features

task. Furthermore, to prevent redundancy within the dataset, any duplicate features were also removed. The following features were eliminated: 'P202', 'P401', 'P404', 'P502', 'P601', 'P603'.

In some instances, textual data might contain trailing or leading spaces. These spaces were removed to standardize the textual information, preventing issues arising from inconsistencies in data representation. The timestamps in the dataset were converted into datetime format, allowing for more intuitive temporal analysis. This conversion enabled us to work with time-based features, which can be crucial for anomaly detection in time series data.

4.4.1 Data Splitting

The preprocessed dataset was divided into training and testing sets to facilitate model evaluation. The splitting was performed to reserve a portion of the data for training the anomaly detection model and another portion for testing its performance. The ratio of data allocated to training and testing was determined to ensure an appropriate balance between model training and evaluation. In our approach 80% of data were used for training and 20% of data were used for testing.

4.5 Evaluation Metrics

The assessment of anomaly detection models is a crucial aspect of ensuring their effectiveness in identifying deviations from normal behavior within the SWaT dataset. Various evaluation metrics were employed to comprehensively gauge the performance of the Dynamic Time Warping (DTW) algorithm and other baseline models. These metrics provide insights into different aspects of model performance, addressing both the ability to correctly identify anomalies and the capacity to avoid false alarms.

4.5.1 Recall (Sensitivity):

Recall, or sensitivity, is a fundamental metric for evaluating the ability of the model to correctly identify anomalies within the dataset. It is calculated as the ratio of true positives to the sum of true positives and false negatives. In the context of industrial control systems, high recall is imperative as it signifies the model's capability to identify all actual positive instances (anomalies), minimizing the risk of overlooking critical deviations.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{1}$$

4.5.2 Precision:

Precision measures the accuracy of the model in identifying true anomalies among the instances predicted as anomalies. It is calculated as the ratio of true positives to the sum of true positives and false positives. Precision provides insights into the reliability of the model's predictions.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{2}$$

4.5.3 F1-Score:

The F1-Score is the harmonic mean of precision and recall. It strikes a balance between these two metrics and is particularly useful in situations where there is an inherent trade-off between false positives and false negatives. In industrial control systems, achieving a high F1-Score is crucial, as it ensures a model that not only minimizes false alarms but also captures a significant proportion of anomalies.

$$\text{F1-Score} = 2 \times \left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \right) \tag{3}$$

4.5.4 Area Under the Receiver Operating Characteristic Curve (AUC-ROC):

The AUC-ROC is a performance metric that evaluates the model's ability to discriminate between normal and abnormal instances across different thresholds. A higher AUC-ROC indicates better discrimination performance. This metric is particularly useful for assessing the overall performance of the model across various operating points.

$$\text{AUC-ROC} = \int_0^1 \text{TPR}\, d\text{FPR} \tag{4}$$

4.5.5 MSE - Mean Squared Error

The MSE is a common loss function used for regression tasks, which measures the mean squared difference between the predicted and actual values.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{5}$$

Where:

- n represents the number of data points

- yi represents the actual or observed value for the i-th data point
- $\hat{y}i$) represents the predicted value for the i-th data point
- $\sum_{i=1}^{n}$ represents the summation symbol, which means to sum the squared differences for all data points from i = 1 to n.

4.5.6 MAE - Mean Absolute Error

The Mean Absolute Error (MAE) formula is used to calculate the average of the absolute differences between the actual (observed) values and the predicted values. It is a common metric for measuring the accuracy of a model.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{6}$$

Where:

- n is the total number of data points.
- yi represents the actual (observed) value for the i-th data point.
- $\hat{y}i$) represents the predicted (estimated) value for the i-th data point.
- $|x|$ denotes the absolute value of x.
- $\sum_{i=1}^{n}$ represents the summation over all data points from i = 1 to n.

4.6 Modeling

In the first approach, we define our LSTM model, we use two LSTM layers where the first layer contains 64 neurons, followed by a Dropout layer of 20% to prevent overfitting and the second layer contains 32 neurons. At the end, we add a dense layer with 1 unit. In each LSTM layer, we are using tanh as activation function and dropout layers with 20%. The model is compiled using the Adam optimizer with learning rate 0.0005 and the mean squared error (MSE) loss function.

In our second approach, we employed the Isolation Forest algorithm to identify unusual patterns within the dataset. The Isolation Forest is a tree-based ensemble model designed for efficient detection of anomalies by isolating them in the feature space. Our Isolation Forest model, was configured with 100 estimators, each consisting of randomly sampled subsets of the data. The model was trained with a contamination level of 1%, reflecting the assumed proportion of anomalies in the dataset. Notably, the Isolation Forest leverages the inherent properties of anomalies, such as being sparse and having distinct feature values, to efficiently isolate them within the constructed trees. The architecture also ensures scalability by employing randomization and parallelization strategies. The model was fitted to the training data using the fit method, allowing it

to learn the underlying patterns of normal behavior and subsequently identify instances deviating from these patterns during the detection phase.

Finally, we implemented an Autoencoder model for unsupervised feature learning and reconstruction. The autoencoder architecture consists of an encoding layer with 32 neurons, ReLU activation, batch normalization, and dropout (0.5) for enhanced robustness. The decoding layer reconstructs the input using sigmoid activation. Trained to minimize reconstruction error, the Autoencoder captures normal patterns in the data, providing a compressed representation. Anomalies, characterized by deviations from the learned normal behavior, are identified through this model. The autoencoder model serves as a powerful tool for efficiently detecting anomalies within the dataset.

5 Results

The results obtained from the experiments are as follows:

	F1 Score	**Recall**	**Precision**
LSTM	0.85	0.92	0.80
Autoencoder	0.85	0.77	0.95
Isolation forest	0.91	0.75	0.83
LSTM with DTW	0.88	0.93	0.83

Table 3: Performance Metrics of the Models

Upon reviewing the obtained results, it becomes apparent that incorporating the calculated Dynamic Time Warping (DTW) distances as an additional feature in the preprocessed dataset, and subsequently feeding it into our LSTM model, Isolation Forest, and Autoencoder, provided valuable observations. The LSTM model exhibited a slight increase in accuracy, reaching 91%, though the difference compared to the model without DTW distances was not substantial. It is essential to acknowledge that the random sampling of 50000 features and the selective consideration of only sensors from stage I and stage II during DTW distance calculation may contribute to this observed effect.

Furthermore, extending this analysis to the Isolation Forest and Autoencoder models adds valuable context to the impact of DTW distances on different anomaly detection methodologies. For both the Isolation Forest and Autoencoder, the F1 score, recall, and precision experienced slight improvements, rendering them worthy enhancements that contribute to the overall effectiveness of these anomaly detection techniques.

6 Conclusion

Based on the experimental results, it can be observed that the proposed method, "Dynamic time warping based anomaly detection for industrial control system using LSTM and AutoEncoder" gave better results than the baseline methods. In conclusion, comparing the Baseline Method and the Proposed Method, it is evident that the proposed method gave better results than the baseline in terms of several evaluation metrics. The proposed method (LSTM with DTW) achieves an accuracy of 0.91, precision of 0.83, recall of 0.93 and F1 score of 0.88. These metrics indicate that the proposed method has improved performance across multiple aspects compared to the baseline method.

Integrating Dynamic Time Warping (DTW) distances as an additional feature in both Isolation Forest-based model and Autoencoder-based approach resulted in a minor yet notable improvement over their respective baseline configurations. The inclusion of DTW in the anomaly detection process not only heightened the accuracy of both models but also demonstrated a slight enhancement in precision, recall, and F1 score. This emphasizes the efficacy of incorporating time-series similarity measures, such as DTW, within the Isolation Forest and Autoencoder frameworks. The flexible application of DTW across these distinct anomaly detection methodologies suggests its valuable role in capturing temporal intricacies for a slightly improved model performance.

It is worth noting that these results were obtained using only 50,000 samples to calculate DTW distances and exclusively considering sensor features from stage I and stage II. The outcomes could potentially be further improved by calculating DTW distances for the entire dataset and incorporating all features. However, due to limitations in machine resources, we were compelled to use dataset sampling and restrict the analysis to stage I and stage II sensor features for DTW distance calculations. Future research and experimentation can delve into exploring additional variations and enhancements to the proposed method, specifically addressing challenges in anomaly detection.

References

[1] Avi Networks. (2023). What is Anomaly Detection? Definition & FAQs — Avi Networks. Available at: *https://avinetworks.com/glossary/anomaly-detection (accessed: 10.10.2023)*

[2] AWS. (2023). What is Anomaly Detection? - Anomaly Detection in ML Explained - AWS. Available at: https://aws.amazon.com/what-is/anomaly-detection/ (accessed: 11.10.2023)

[3] Bhan, S. (2022). What is Dynamic Time Warping?. Available at: *https://medium.com/mlearning-ai/what-is-dynamic-time-warping-253a6880ad12* (accessed: 11.10.2023)

[4] Mathur, A., Tippenhauer, N. (2016). SWaT: A Water Treatment Testbed for Research and Training on ICS Security, *IEEE Conference Publication — IEEE Xplore*. Vienna, 11-11 April 2016. Vienna: IEEE, pp. 31-36

[5] iTrust (2015). *Secure Water Treatment (SWaT) Testbed*. Available at: https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/

[6] Diab, D., AsSadhan, B., Binsalleeh, H., Lambotharan, S., Kyriakopoulos, K., Ghafir, I. (2019). Anomaly Detection Using Dynamic Time Warping, *IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*. New York, 11-11 April 2016. New York: IEEE, pp. 193-198

[7] Tiyani (2023). *Understand how Dynamic Time Warping (DTW) works step by step*. Available at: *https://medium.com/@tianyi.datascience/understand-how-dynamic-time-warping-dtw-works-step-by-step-fc3c60237479*

[8] IBM, "Data Labeling." https://www.ibm.com/topics/data-labeling. (accessed: 31.10.2023)

[9] A. Networks, "Anomaly detection." https://avinetworks.com/glossary/anomaly-detection. (accessed: 31.10.2023)

[10] Dataloop AI, "Data labeling challenges." https://dataloop.ai/blog/data-labeling-challenges/, 2021. (accessed: 01.11.2023)

[11] M. Kuchta, "Anomaly detection in time series," 2021. (accessed: 01.11.2023)

[12] Z. Liu, Y. Yu, Y. Ma, W. Wang, and J. Tang, "Deep learning for multi-view multi-instance multi-label classification with incomplete labels," Machine Learning, pp. 1–37, 2022.

[13] Wang, C.; Wang, B.; Liu, H.; Qu, H. Anomaly Detection for Industrial Control System Based on Autoencoder Neural Network.Wirel. Commun. Mob. Comput. 2020, 2020, 8897926

[14] Diab DM, AsSadhan B, Binsalleeh H, Lambotharan S, Kyriakopoulos KG, Ghafir I. Denial of service detection using dynamic time warping.Int J Network Mgmt. 2021;31(6):e2159.

[15] Kim J, Yun J, Kim H. Anomaly Detection for Industrial Control Systems Using Sequence-to-Sequence Neural Networks. 2019.

[16] Hoang NX, Hoang NV, Houng TT, Du NH. Explainable anomaly detection for Industrial Control System CyberSecurity. 2022.

[17] Pang J, Pu X, Li Ch. A Hybrid Algorithm Incorporating Vector Quantization and One-Class Support Vector Machine for Industrial Anomaly Detection. 2022.

[18] Yuqi Chen Christopher M. Poskitt Jun Inoue Yoriyuki Yamagata and Jun Sun, eds. Anomaly Detection for a Water Treatment System Using Unsupervised Machine Learning. Ikeda, Japan Singapore, Singapore, 2017.

[19] M. Elnour, N. Meskin, K. Khan, R. Jain, *A dual-isolation-forests-based attack detection framework for industrial control systems*, IEEE Access, 8 (2020) 36639–36651.

[20] J. Inoue, Y. Yamagata, Y. Chen, C. M. Poskitt, J. Sun, *Anomaly detection for a water treatment system using unsupervised machine learning*, in: Proceedings of the IEEE International Conference on Data Mining Workshops (ICDMW), 2017, pp. 1058–1065.

[21] Geron, A. (2017). *Hands-on Machine Learning with Scikit-Learn and TensorFlow*. 1st ed. USA: O'Reilly Media, Inc

[22] Liu, Fei Tony, Ting, Kai Zhou, Zhi-Hua. (2009). *Isolation Forest*. 413 - 422. 10.1109/ICDM.2008.17.

[23] Kim, SungJin, WooYeon Jo, and Taeshik Shon. *"APAD: Autoencoder-based payload anomaly detection for industrial IoE."* Applied Soft Computing 88 (2020): 106017.