Knowledge-Augmented Anomaly Detection in Small Lot Production for Semantic Temporal Process Data

Jianjie Lin, Markus Rickert, Long Wen, Fengjunjie Pan, Alois Knoll

***Abstract*— In order to prevent unplanned downtime caused by potential errors in robot systems used for industrial automation, we propose a new approach to anomaly detection that employs a Transformer-based reconstruction network to identify anomalies in skill-based manufacturing. Based on a semantic description of processes, products, and resources, a semantic manufacturing execution system synthesizes a suitable robot program and executes the process. Our method utilizes these descriptions to segment and automatically label relevant process data, which allows for automated configuration of the detection pipeline. To address data scarcity, we use a sliding window for data augmentation and leverage the Transformer's attention mechanism to effectively extract semantic dependencies from the time series data. By analyzing the residuals of the reconstructed time series data, we can detect process-related anomalies. We conducted experiments on a real robot workcell and demonstrate that our approach outperforms other competitive concepts.**

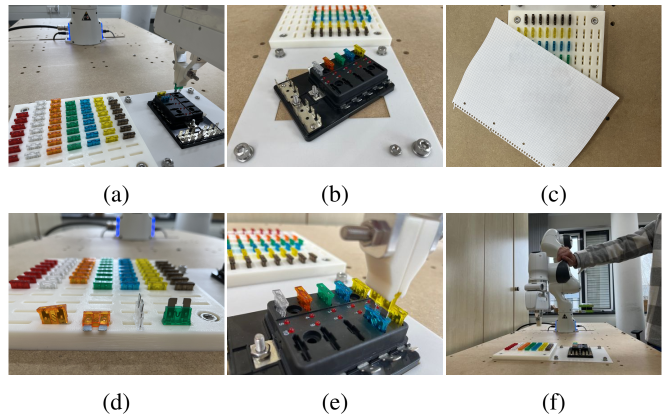

Fig. 1: Automotive fuse box assembly task with (a) normal execution and various anomaly cases: (b) location-induced anomaly due to incorrect fuse box location, (c) cluttered workspace, (d) improper placement of the fuse, (e) programming error, and (f) human interruption.

I. Introduction

The industrial manufacturing industry is fiercely competitive, with operational costs playing a significant role in the total cost of owning robot-based production systems. These costs include the labor and resources required to set up, reconfigure, and operate the systems. To meet the rising demand for tailored products while reducing operational costs, the level of automation in manufacturing systems needs to be increased [1]. This goes beyond the basic programming of control logic of robots and their tools. It also involves configuring analytics mechanisms that monitor production processes and collecting this data. Even the most advanced technical systems are not immune to errors, therefore a smart production environment must be able to identify and address issues automatically. A continuous monitoring of process parameters and sensor data would enable the implementation of corrective measures that boost the production system's ability to withstand unforeseen circumstances and external factors.

As cognitive robot-based manufacturing systems become increasingly capable of independently assessing their production objectives and capacities, traditional methods of developing and refining anomaly detection techniques are no longer viable. As the precise sequence of activities within these systems is automatically generated based on product and process specifications, manual approaches to anomaly detection can no longer be reliably employed. To detect anomalies in small-batch and high-variation manufacturing, solutions must be continuously trained using data from only a few production cycles. However, collected data must first be labeled to assign significance to raw sensor readings when using supervised or semi-supervised machine learning techniques. Unfortunately, in many cases, this manual labeling process is still time-consuming. Furthermore, anomaly detection in these environments presents unique challenges not found in traditional predictive maintenance applications. Small-volume and high-variety manufacturing environments are frequently unstructured, making it critical for anomaly detection solutions to operate in such environments.

In this paper, we combine semantic knowledge with the strength of recent advances in deep learning, in particular with Transformer-based models [2], for the monitoring and anomaly detection in skill-based manufacturing. The Transformer is currently widely used in natural language processing due to its attention mechanism that allows the model to directly query the state at an earlier point in the time series sequence and has the capability to extract the semantic affinity within a data. The proposed network is a semi-supervised learning approach, where only labeled normal operation data is used for training, and the evaluation is performed on an unlabeled data set containing both normal and anomalous operations. Unlike supervised learning approaches that classify events using their labels, we use the reconstruction error to determine whether an operation is

Jianjie Lin, Markus Rickert, Long Wen, Fengjunjie Pan, Alois Knoll are with Robotics, Artificial Intelligence and Real-time Systems, School of Computation, Information and Technology, Technische Universität München, Munich, Germany {jianjie.lin, rickert, wenl, f.pan, knoll}@in.tum.de

normal or abnormal. A robust and dependable model should generate samples with small errors compared to their real counterparts in the absence of anomalies. Furthermore, the reconstruction error should be smaller than a predefined threshold value. In the presence of anomalies, the model should produce a more significant reconstruction error than the threshold value. We determine the threshold value by computing the means of the reconstruction errors while evaluating the model with the training data set.

In our paper, we make four main contributions. Firstly, we propose a method for segmenting and labeling process data based on semantic descriptions of the manufacturing process and operational environment. This approach enables us to identify relevant data points and assign them to their respective components and skills. Secondly, we introduce a skill based Transformer-based reconstruction network for detecting anomalies in the manufacturing process. This approach utilizes the self-attention mechanism to capture long-range dependencies and variable-sized receptive fields. By computing the reconstruction error between the input and output data, we can detect abnormal behavior in the manufacturing process. Thirdly, we combine our semantic knowledge with the anomaly detection system in a monitoring pipeline. By utilizing the labeled data and the semantic descriptions of the manufacturing process, we can improve the accuracy of our anomaly detection system and reduce false alarms. Finally, we validate our approach with real-world data sets collected from a skill-based manufacturing system. Our results show that our approach can accurately detect anomalies in the manufacturing process.

II. Related Work

Anomaly detection has been widely studied and various methods have been proposed. Among the most popular are frequentist change-point detection methods, such as sequential probability ratio test (SPRT), cumulative sum (CUSUM), and Generalized Likelihood Ratio (GLR), due to their suitability for addressing online problems. These algorithms monitor the logarithm of the likelihood ratio between two consecutive intervals of the same time series and declare a change-point when the statistical properties of the intervals differ. However, their effectiveness is limited by their dependence on knowledge of the probability distribution functions of the data. Other approaches, such as spectral-based methods, maximum likelihood estimation, and subspace identification, also have limitations because of their reliance on pre-specified thresholds that are difficult to set beforehand. Bayesian approaches have been proposed as an alternative, but most have been used offline for retrospective studies that require the entire dataset before computing the probability of a change-point. To address these challenges, online Bayesian change-point detection (OBCPD) methods were introduced in [3]. OBCPD recursively determines the run length, which is the time since the last change-point, by using the sufficient statistics of the data. However, a drawback of OBCPD is that it requires knowledge of the sufficient statistics before updating them.

Recurrent Neural Networks (RNNs), and the recent advancements in deep learning, are increasingly benefiting anomaly detection approaches. They have proven to be efficient in dealing with both univariate and multivariate time series data. In fact, several studies, such as Zhang et al. [4], Maia et al. [5], and Yadav et al. [6], have explored the use of RNNs for multivariate anomaly detection. For instance, Nanduri et al. [7] used LSTM and GRU in conjunction with aircraft flight data, and their approach demonstrated effectiveness in detecting anomalies. Hundman et al. [8] investigated anomaly detection methods using LSTM and spacecraft telemetry, where they employed residuals from prior batches and domain experts' knowledge to determine the anomaly detection thresholds needed in subsequent batches. Additionally, Ebrahimzadeh et al. [9] utilized CNNs with wavelet transform to detect anomalies in synthetic data. Their method's strength lies in its ability to detect gradual drifts over time.

In order to enhance anomaly detection in time series data, various approaches combining LSTM with autoencoder have been examined. Malhotra et al. [10] explored a semi-supervised learning method that utilizes LSTM with autoencoder and univariate time series to compute the reconstruction errors of the autoencoder and use it as an anomaly detector. Kim et al. [11] exploited Convolutional Variational Autoencoders (CNN-VAE) to reduce the size, complexity, and training cost of the autoencoder without altering its ability to detect anomalies. Kieu et al. [12] proposed approaches using LSTM-based autoencoders and convolutional autoencoders with data enrichment during the pre-processing phase, demonstrating strengthened autoencoder modeling capabilities and anomaly detection performance.

The development of generative adversarial network (GAN) in the computer vision community has paved the way for its use in anomaly detection of time series data. The state-of-the-art approaches in GAN-based image anomaly detection were surveyed in Di et al. [13], where a new GAN structure was proposed in Akcay et al. [14] utilizing CycleGAN to consider the notion of consistency relationship. However, these surveyed methods are only applicable to image-based anomaly detection and are not suitable for time series data. Li et al. [15] developed an unsupervised GAN, utilizing LSTM to capture the temporal characteristics of the system in the Generator and Discriminator of the network, which was successfully used for detecting anomalies.

For a comprehensive discussion of anomaly detection in time series, Cook and Wolf [16] is a valuable resource. In this work, unlike the aforementioned approaches, a Transformer is employed to process the time series data set. The Transformer shows many benefits compared to LSTM-based approaches by utilizing the attention mechanism [2].

III. Semantic Process Description

Processes in industrial robotics and industrial automation are traditionally specified via imperative programming paradigms. A programmer manually defines the sequence of individual functions that need to be executed in order to

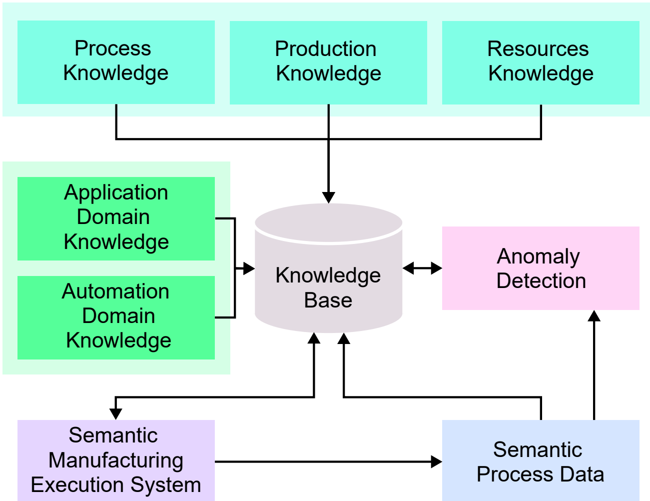

Fig. 2: Overview of knowledge-augmented anomaly detection architecture.

achieve the desired process result. This has to be manually adapted to new variants or even completely redesigned for a different process. In contrast to this, approaches in service robotics are typically based on a declarative programming paradigm, where the intended goal is specified and a suitable order of tasks is generated and executed in order to achieve this goal.

Our approach is based on such a declarative paradigm, where the intended goal product and its process is specified and the system automatically synthesizes a suitable task sequence based on the capabilities of its resources [17]. Here, a process represents a sequence of tasks on an object-centric level that can be assigned to individual actors such as a robot or a human, e.g., *pick up fuse A*. A task is then implemented via a number of skills with specific parameters such as the fuse's position and orientation or the proper insertion force. The knowledge base in our architecture (Fig. 2) uses a graph database with an inference engine and is based on the Web Ontology Language (OWL). It stores semantic knowledge of the process, product, and resources together with application-specific and automation-related domain knowledge. This includes detailed semantic models of objects, e.g., boundary representation of CAD data and geometric constraints for assembly [18], [19], and the system's resources, e.g., robots, grippers, objects, and their layout (Fig. 4). The resource models incorporate semantic knowledge of the OPC UA middleware's information model to represent individual elements of a resource such as function calls or data, e.g., joint positions, joint torques [20].

The process used in our evaluation assembles an automotive fuse box based on a vehicle's individual configuration as ordered by the customer and therefore exhibits a large number of variations, e.g., radio type, air conditioning, driver assistance functions. This assembly is currently performed manually, as automation via traditional imperative programming is economically unviable [21]. In our system, all process and product-related knowledge is available and can be triggered by the specification of the vendor's product ID. This is received by the system's semantic manufacturing execution system (sMES) that synthesizes a suitable robot program for the current product (Fig. 3). The generated sequence of tasks is executed and monitored, while the process-relevant semantic data is collected and used as input for the anomaly detection module. This data is semantically annoted (e.g., *joint torque data*) and is connected to the currently executed task (e.g., *pick up fuse A*) and its related information (e.g., *object location*).

The semantic digital twin of all manufacturing resources provides rich contextual information that allows sensor data generated during production runs to be automatically annotated with relevant information. This includes relating a measured force to the involved robot, tool, target object, and task description, among other factors. Machine learning-based anomaly detectors can benefit from automatically labeled data and contextual information during training. Once trained, these detectors can evaluate new sensor data samples and determine whether the current skill execution is normal or an anomaly. To accomplish this, semantic models are used to segment process data into individual skill executions within the system, and label time series data for the anomaly detector to use automatically.

IV. Anomaly Detection Algorithm

The Transformer [2] is widely applied in natural language processing (NLP) to handle sequential data, such as generating translations or text summaries, without regard for the order in which the sequences are presented. In addition, the Transformer leverages attention mechanisms to learn interdependencies within sequences. As with most Seq2Seq models, the Transformer is structured as an encoder-decoder architecture and can be used for input reconstruction.

A. Problem Formulation

Anomaly detection is a semi-supervised machine learning technique that relies on labeled normal and non-anomalous occurrences to train a model. The problem involves a training dataset, $X \in \mathbb{R}^{M\times n}$, and a testing dataset, $Y \in \mathbb{R}^{N\times n}$, where M and N are the sizes of the training and testing datasets, respectively, and n represents the size of the selected features obtained from robot movement. Each data sample, X_i, in X, Y corresponds to a high-dimensional piece of information, $X_i = x_0, \ldots, x_{n-1}$, that is time-stamped and derived from the status messages of executable robot system skills. Normally, the normal data should cover all areas of the data space accessed during regular task completion, while anomalous data exist in unknown areas of the data space. One way to distinguish between anomalous and nominal skill execution is to check whether the queried data point lies in the normal data space by computing its distance to all known normal data samples. However, this method is impractical since it is not feasible to cover all possible measurements during data collection. Alternatively, standard approaches employ kernel functions to approximate the normal data space, such as Support Vector Machines (SVM) or Gaussian

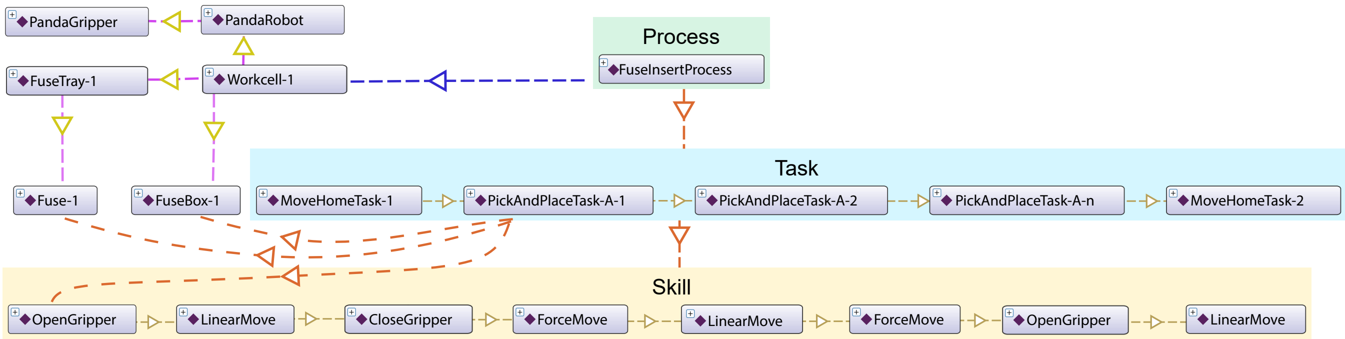

Fig. 3: Semantic Description of Pick-and-Place Skill in OWL-Based Representation. The skill involves a combination of linear and force movements, as well as open and close gripper operations.

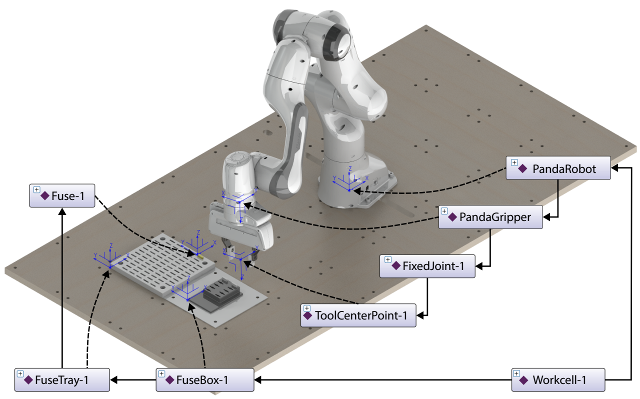

Fig. 4: The resources used in the assembly process and their semantic layout description.

Mixture Models (GMM). These approaches assign each point x in the data space a function value that corresponds to the probability of belonging to the trained data, but they have been shown to be less effective than deep learning-based methods. To address this issue, we have employed a Transformer-based network for anomaly detection. By analyzing the relationships between sequential data points, the Transformer can identify anomalous behavior that deviates from normal patterns within the data. Additionally, the Transformer's attention mechanisms allow it to focus on specific parts of the input sequence that may be relevant to detecting anomalies. This makes it a powerful tool for identifying subtle deviations or abnormalities within complex datasets, making it a popular choice for anomaly detection.

B. *Transformer-based Anomaly Detection Structure*

Fig. 5 depicts an overview of the overall structure of the transformer-based anomaly detection concept. Individual parts of this structure are explained in the following subsections.

1) Transformer-Encoder: Fig. 6 depicts Encoder, which consist of an embedding layer, a positional encoding layer, and $N\times$ stacked encoder layers. In this work, we set $N = 4$. The structure is illustrated in Fig. 6a (left). The embedding layer maps time series data to a d_{model}-dimensional vector using a fully connected network. We utilize a positional encoding layer with sine and cosine functions to label the sequential position information, which is crucial for interpreting the meaning of time series data. The resulting hidden feature is then fed to the encoder layer, which comprises two sub-layers: self-attention and a positional, fully connected network. Self-attention means that no external information flows, which is the central difference between attention mechanisms in the encoder and decoder. Each sub-layer uses a residual net-like structure together with a normalization layer. The extracted feature is then considered as the input to the reconstruction network. The encoder layer at lth layer can be formulated as

$$Z^l = \text{LayerNorm}(\text{Attention}(X^{l-1}) + X^{l-1}) \quad \text{(1a)}$$

$$X^l = \text{LayerNorm}(\text{FFN}(Z^l) + Z^l) \quad \text{(1b)}$$

where $X^l \in \mathcal{R}^{N\times d_{\text{model}}}, l \in 1, \dots, N$ indicates the output of the l-th layer, and Z^l is the hidden state. The attention function is used to compute the association relationships insides a time series.

2) Attention Mechanism: Attention is the core component of Transformer-based seq2seq networks. In the context of sequential data, using a convolutional kernel, such as Stacked dilated convolutions, as proposed in [22], is a common approach to capture relationships between positions in the sequence. However, such a design cannot fully guarantee to capture all the dependencies. On the other hand, (self-)attention computes the dot product between each position in the sequence and establishes a variable-sized receptive field. Additionally, the multiplication operator enables crisp error propagation. Fig. 6 illustrates the attention component of the network and its vicinity. The attention mechanism assigns a weight to each position in the sequence and computes the weighted sum of values to produce the output. The weights are calculated based on the dot product between the query vector and the key vector of each position in the sequence. The output of the attention mechanism is a linear combination of the values, where the weights determine the importance of each position in the sequence. By using multiple heads in the attention mechanism, the network can attend to different relationships and capture more complex dependencies in the sequence.

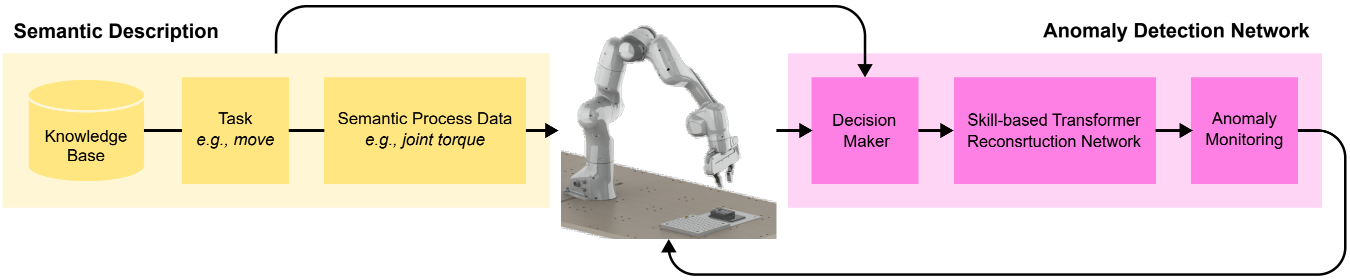

Fig. 5: The knowledge-augmented neural network structure for anomaly detection. The process begins with semantic description in the first block, where task-specific information for anomaly detection is extracted and provided as input to the transformer-based anomaly detection network. The network identifies anomalies and annotates them with reconstruction error, which is then fed back to the robot. Based on this feedback, the robot or human can take appropriate actions, such as monitoring or stopping the process.

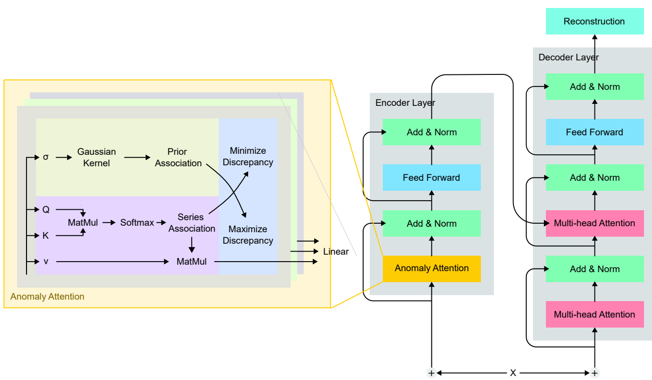

Fig. 6: Illustration of the transformer-based anomaly detection architecture, featuring the incorporation of prior-association and series association mechanisms in lieu of the conventional attention mechanism, for enhanced semantic representation.

In this study, we enhance the attention mechanism by incorporating an anomaly attention block that simultaneously takes into account prior-association and series association. To represent the prior association, we employ a learnable Gaussian kernel that calculates the prior based on the temporal distance. The Gaussian kernel, which has a learnable $\sigma_i = X^{l-1}W^l_\sigma$, is interpreted as.

$$P^l = \text{Rescale}\Big([\frac{1}{\sqrt{2\pi\sigma_i}}\exp(-\frac{|j-i|^2}{2\sigma_i})]_{i,j\in(1,\ldots,N)}\Big) \quad (2)$$

The series association is computed as the normal way as stated in the original paper.

$$S^l = \text{Softmax}(\frac{QK^T}{\sqrt{d_{\text{model}}}}) \quad (3)$$

where $Q = X^{l-1}W^l_Q$ and $K = X^{l-1}W^l_K$.

3) Reconstruction network: At the first stage, the Transformer encoder extracts a global feature $\mathbf{F}_{\text{TE}}$ to represent the robot raw data. We then feed the extracted features into a reconstruction network to reconstruct the robot data. The reconstruction network can be structured as a standard auto-encoder, such as the transformer decoder The reconstruction network can be mathematically formulated as

$$Y = \text{Decoder}(\mathbf{F}_{\text{TE}}) \quad (4)$$

a) Loss Function of anomaly detection Network: Our network is classified as a semi-supervised learning model, with only normal execution being utilized for training. The reconstruction loss is the primary objective for optimizing the model, and we have replaced the standard attention mechanism with the anomaly attention block [23]. This modification has resulted in the introduction of additional losses, such as the Association Discrepancy loss, which is defined as follows:

$$L_{\text{AssDiss}(P,S,X)} = [\frac{1}{L}\sum_{l=1}^{L}\big(\text{KL}(P^l_{i,:}||S^l_{i,:}) + \text{KL}(S^l_{i,:}||P^l_{i,:})\big)]_{i\in 1,\ldots,N} \quad (5)$$

The Association Discrepancy loss ($L_{\text{AssDiss}(P,S,X)}$) is determined by computing the discrete distribution between P and S using the KL divergence (KL). This value is then utilized to determine the presence of anomalies, whereby anomalies are characterized by having a smaller Association Discrepancy value than in normal situations [23]. On the other hand, the reconstruction loss (L_{recon}) is computed to determine the distance between a given normal data sample and its corresponding reconstructed version. This distance is computed as

$$L_{\text{recon}} = ||X - \hat{X}||_F\,. \quad (6)$$

where $\hat{X} \in \mathcal{R}^{N\times d}$ is the reconstructed version of X, and the operator $\|.\|_F$ is the Frobenius matrix norm. The unimodal nature of the prior-association results in the discrepancy loss driving the series-association to prioritize the non-adjacent area. This, in turn, poses a greater difficulty in reconstructing anomalies and enhances their identification. Therefore, we can combine the reconstruction loss together with association discrepancy to form the final loss function

$$L_{\text{final}} = L_{\text{recon}} - \lambda L_{\text{AssDiss}(P,S,X)} \quad (7)$$

where the hyper-parameter λ serves to regulate the impact of the Association Discrepancy loss. The Minimax Strategy

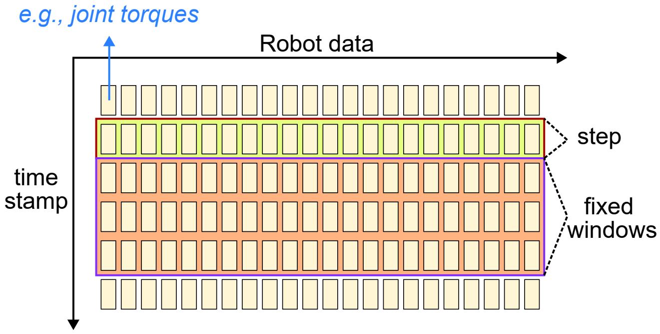

Fig. 7: Data augmentation with predefined window size and shifting window step. Each row indicates the robot data at a time, and column is the time stamp.

proposed in [23] is utilized to train the network. During the inference stage, only the reconstruction loss is used as a criterion, with the training value acting as a reference value to identify anomalies.

V. Experiments

The proposed concept for anomaly detection was implemented and evaluated in a real robot workcell, as depicted in Fig. 8. PyTorch was used to implement the anomaly detection algorithm, and the models were optimized using an Adam optimizer in conjunction with a CosineAnnealingLR scheduler. Each encoder was set to 3 with 8 heads for the multi-head attention mechanism. Additionally, the values of $d_{\text{model}} = 256$ and $d_{\text{ff}} = 512$ were set in the positional feed forward network. The network was trained on a Linux system with an Intel i9-12900K processor with eight 3.2 GHz cores, 32 GB of RAM, and one Nvidia RTX3080 GPU.

A. Skill-Based Semantic Data Acquisition and Processing

The data used in this study was collected from a Franka Panda robot that had a parallel gripper attached, as depicted in Fig. 8. As explained in Section IV-A, each time series data point is represented as a vector that contains various features such as joint positions, joint velocities, end joint force, all of which are labeled according to the semantic skill descriptions of the robot. To enhance the quality of the machine learning model, data augmentation is used, which plays an important role, especially when large data sets are not available in industrial applications of small and medium-sized enterprises (SMEs). To train and evaluate the approach, we collected data from 100 normal pick and place skill executions with fuse insertion task, as well as 80 anomalous pick and place runs representing 5 different failure types (Fig. 1). To split the data set into subsets, we use a sliding window with a size of 300 time stamps, and each two sliding windows is shifted with 30 time stamps to remain the continuities. Our anomaly detection structure is trained on each subset of data samples, and during the inference stage, we apply the same strategy to obtain the reconstruction error $l_{\text{recon},i}$ at piece $0 \leq i \leq N$. We compare $l_{\text{recon},i}$ with the maximum normal reconstruction error represented by $l_{\text{recon,normal,max}}$ and assign a predicted probability P_i of 1 for $l_{\text{recon},i} \leq$ $l_{\text{recon,normal,max}}$ and 0 otherwise. We then compute the mean probability for N pieces with $P = \sum_i^N \frac{P_i}{N}$. If an execution has $P \geq 0.5$, it is considered a normal operation; otherwise, it is considered an anomalous operation.

Fig. 8: Robot cell used to conduct experiments for anomaly detection in fuse insertion task.

B. Nominal and anomalous use case variants

The nominal task execution involves picking a fuse and placing it into the fuse box. To demonstrate the task, we tasked our robot demonstrator with inserting a set of fuses into holes within the tabletop's dimensions, as shown in Fig. 8. Consequently, the pick and place operations involved varying motions that differed in their coordinates and duration. We introduced several types of anomalies to evaluate the system's performance. The first anomaly involved the fuse box's location by placing it in an incorrect position, as depicted in Fig. 1b. This led to a completely wrong fuse hole position. To simulate a situation where an object was mistakenly placed on the fuse supplier, we placed a paper on top of the fuse supplier, as shown in Fig. 1c. The third anomaly was caused by improper placement of the fuse in the fuse supplier, as depicted in Fig. 1d. Additionally, we simulated a scenario where the robot placed the fuse in an already occupied position due to programming error. The last type of anomaly was caused by human interaction during the motion, as shown in Fig. 1e.

C. Evaluation Metric

The following metrics are used to evaluate the performance of our proposed approach: Accuracy (Accu), Precision (Pre), F_1 score,

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \qquad \text{Pre} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{8}$$

$$\text{Rec} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \qquad F_1 = 2 \cdot \frac{\text{Pre} \cdot \text{Rec}}{\text{Pre} + \text{Rec}} \tag{9}$$

where True Positive (TP) indicates the detection of normal operation, while True Negative (TN) is the detection of anomalies. False Positive (FP) is the falsely detected normal while False Negative (FN) is the falsely detected anomalies.

D. *Evaluation of Anomaly Detection Models*

Industrial tasks are often repetitive in nature, such as pick-and-place operations and peg insertion manipulation. In this section, we will examine anomaly detection in different scenarios. Our model is first trained on a nominal dataset and then evaluated on a dataset that includes both nominal and anomalous runs. The training dataset includes a partial view of the nominal grasp dataset. This scenario simulates the conditions of repetitive work in a real-world application. In a more strict evaluation, we trained our neural network with normal pick-and-place executions and evaluated it with anomalous and normal runs that were not known before. This scenario covers the situation in which a manufacturer deploys a new process pipeline for handling the same object. We will compare our approach with three other competitive autoencoder methods. The results are summarized in Table I. It can be seen that our proposed approach can predict TP with an accuracy of 87.1% and FN of 12.8%, while TN is 96.3% and FP is 3.7%. By computing the mean values of TP and TN, we can obtain an overall anomaly detection accuracy of 94.1%.

Furthermore, we compared the proposed approach with three different types of autoencoder approaches. To ensure a fair comparison, we used the same optimization strategy and batch size to train all three models. Additionally, we evaluated different network structures for each model by adjusting the number of layers in the encoder and decoder to extract better latent features and changing the dropout value to avoid overfitting. We selected the best-performing solutions as our comparison results. The LSTM-based autoencoder used two stacked LSTM layers in the encoder and decoder, respectively. The result in the first row demonstrates that the LSTM autoencoder cannot reconstruct the normal sample well enough and has a TP value of only 50%. It has a 30% probability of predicting anomalous events as normal operations. The MLP-based autoencoder had a stack of linear layers in the sizes of $[29, 64, 64, 128, 256]$, while the CNN-based autoencoder used the same choice of layer sizes in Conv2d combined with a kernel size of 3. From the results in Table I, we can conclude that the MLP-based autoencoder and CNN-based autoencoder show similar performance and outperform the LSTM-based autoencoder. However, they still show a significant gap compared to our approach. From the results, we can conclude that our proposed approach can build a better normal operation space compared to the other three methods.

To further analyze the performance of our proposed model for anomalous events, we separately evaluated the anomaly in each skill. The results are summarized in Table II. It is interesting to see that in all skills, the anomaly detection in terms of true negative rate (TNR) can achieve more than 90%. Here, we need to point out that in the evaluation dataset, the normal execution has only 12 set data, but the anomaly execution is roughly 85 set. Therefore this is an imbalance dataset. But the whole accuracy executed more than 90%. It can proven that the proposed approach is efficient to detect the anomaly in the real world.

TABLE I: Comparison of results of different anomaly detection approaches for experiments in a fully-unknown scenario. Note: values are given as a percentage.

Ground Truth	TPR ↑	FNR↓	FPR↓	TNR↑	Accu↑	Pre↑	F_1↑
LSTM-AE	50.0	50.0	30.0	70.0	63.0	62.5	55.6
CNN-AE	80.0	20.0	50.0	50.0	65.0	61.5	69.6
MLP-AE	60.0	40.0	25.0	75.0	68.0	70.6	64.9
Our approach	87.1	12.8	3.7	96.3	94.1	95.6	91.3

TABLE II: Individual anomaly detection results for each skilled of introduced anomaly. Note: values are given as a percentage.

Skills	TPR↑	FNR↓	FPR↓	TNR↑	Accu↑
Lift skill	100.0	0.0	0.0	100.0	100.0
Pick skill	83.3	16.6	8.6	91.3	90.4
Place skill	81.8	18.1	0.0	100.0	93.5
Plugin skill	83.3	16.7	6.1	93.9	92.6

E. *Visualization of Model Reconstruction Results*

Figure 9 displays the results of the model reconstruction experiment. Our analysis involves reconstructing the seven joint torque values of the robot for both normal and anomalous events. To facilitate training and evaluation, all values were normalized between the range of 0 and 1. In the case of normal events, as depicted in Fig. 9a through Fig. 9g, the reconstruction output is similar to the input values, resulting in a small reconstruction loss. However, for anomalous events, as shown in Fig. 9h through Fig. 9n, the reconstruction loss is considerably larger. Therefore, we can easily differentiate between normal and anomalous events.

The visualization indicates that our proposed network can map all input values through an encoder to a normal operation space and reconstruct normal operations through this hidden state. When an anomalous event occurs, the decoder rebuilds the output from the normal space, which differs from the anomalous space. Hence, we can use this approach to distinguish between normal and anomalous skill executions.

VI. Conclusion

We developed a new approach to detecting anomalies in manufacturing scenarios by integrating semantic knowledge with a reconstruction network based on Transformers. By leveraging our semantic descriptions of manufacturing resources, we can automatically gather relevant training and testing data for specific skills, allowing us to analyze the manufacturing process at a skill level. This enables our skill-driven production system to monitor individual skill executions and evaluate their normal conditions. The Transformer-based reconstruction network effectively captures higher-level features in the generated time series data using its self-attention mechanism.

To evaluate our method, we collected a dataset of normal and anomalous task executions on a physical robot workcell.

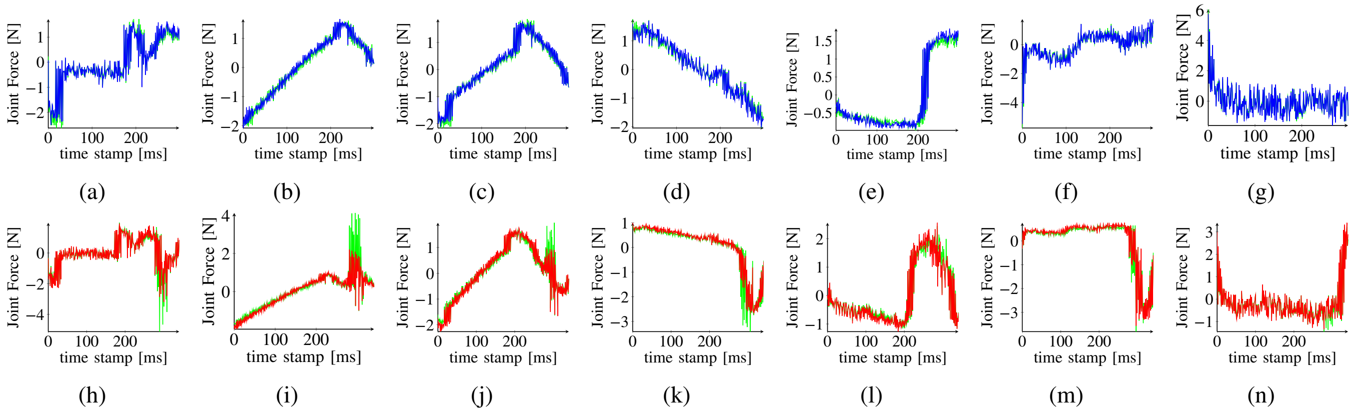

Fig. 9: This visualization displays an excerpt of reconstructed output generated by our proposed anomaly detection network for nominal and anomalous operation. The input data is represented by the green line, while the blue/red line shows the corresponding reconstructed output. The first row presents the results for a normal execution run, specifically the normalized joint force of the Franka robot (a)–(g). In contrast, the second row showcases the corresponding plots in case of an anomaly (h)–(n).

Our approach outperformed other state-of-the-art methods, yielding promising results. As a future extension, the system could either alert an operator to inspect the anomaly or handle it on its own.

References

[1] A. Perzylo, M. Rickert, B. Kahl, N. Somani, C. Lehmann, A. Kuss, S. Profanter, A. B. Beck, M. Haage, M. R. Hansen, M. T. Nibe, M. A. Roa, O. Sörnmo, S. G. Robertz, U. Thomas, G. Veiga, E. A. Topp, I. Kessler, and M. Danzer, "SMErobotics: Smart robots for flexible manufacturing," *IEEE Robotics & Automation Magazine*, vol. 26, no. 1, pp. 78–90, Mar. 2019.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017.

[3] R. P. Adams and D. J. MacKay, "Bayesian online changepoint detection," *arXiv preprint arXiv:0710.3742*, 2007.

[4] W. Zhang, W. Guo, X. Liu, Y. Liu, J. Zhou, B. Li, Q. Lu, and S. Yang, "LSTM-based analysis of industrial IoT equipment," *IEEE Access*, vol. 6, pp. 23 551–23 560, 2018.

[5] R. Maia, "Multivariate temporal data analysis for vessels behavior anomaly detection," in *Proceedings of the EuroSys Doctoral Workshop, European Conference on Computer Systems*, 2018.

[6] M. Yadav, P. Malhotra, L. Vig, K. Sriram, and G. Shroff, "ODE-augmented training improves anomaly detection in sensor data from machines," *arXiv preprint arXiv:1605.01534*, 2016.

[7] A. Nanduri and L. Sherry, "Anomaly detection in aircraft data using Recurrent Neural Networks (RNN)," in *Proceedings of the Integrated Communications Navigation and Surveillance Conference (ICNS)*, 2016, pp. 5C2–1.

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

[9] Z. Ebrahimzadeh and S. Kleinberg, "Multi-scale change point detection in multivariate time series," in *NIPS Time Series Workshop*, 2017.

[10] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "LSTM-based encoder-decoder for multi-sensor anomaly detection," *arXiv preprint arXiv:1607.00148*, 2016.

[11] D. Kim, H. Yang, M. Chung, S. Cho, H. Kim, M. Kim, K. Kim, and E. Kim, "Squeezed convolutional variational autoencoder for unsupervised anomaly detection in edge device industrial internet of things," in *Proceedings of the International Conference on Information and Computer Technologies (ICICT)*, 2018, pp. 67–71.

[12] T. Kieu, B. Yang, and C. S. Jensen, "Outlier detection for multidimensional time series using deep neural networks," in *Proceedings of the IEEE International Conference on Mobile Data Management (MDM)*, 2018, pp. 125–134.

[13] F. Di Mattia, P. Galeone, M. De Simoni, and E. Ghelfi, "A survey on gans for anomaly detection," *arXiv preprint arXiv:1906.11632*, 2019.

[14] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "GANomaly: Semi-supervised anomaly detection via adversarial training," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 622–637.

[15] D. Li, D. Chen, J. Goh, and S.-k. Ng, "Anomaly detection with Generative Adversarial Networks for multivariate time series," *arXiv preprint arXiv:1809.04758*, 2018.

[16] A. A. Cook, G. Mısırlı, and Z. Fan, "Anomaly detection for IoT time-series data: A survey," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6481–6494, 2019.

[17] A. Perzylo, N. Somani, S. Profanter, I. Kessler, M. Rickert, and A. Knoll, "Intuitive instruction of industrial robots: Semantic process descriptions for small lot production," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, Republic of Korea, Oct. 2016, pp. 2293–2300.

[18] A. Perzylo, N. Somani, M. Rickert, and A. Knoll, "An ontology for CAD data and geometric constraints as a link between product models and semantic robot task descriptions," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, Sept. 2015, pp. 4197–4203.

[19] N. Somani, A. Gaschler, M. Rickert, A. Perzylo, and A. Knoll, "Constraint-based task programming with CAD semantics: From intuitive specification to real-time control," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, Sept. 2015, pp. 2854–2859.

[20] A. Perzylo, S. Profanter, M. Rickert, and A. Knoll, "OPC UA nodeset ontologies as a pillar of representing semantic digital twins of manufacturing resources," in *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Zaragoza, Spain, Sept. 2019, pp. 1085–1092.

[21] A. Perzylo, I. Kessler, S. Profanter, and M. Rickert, "Toward a knowledge-based data backbone for seamless digital engineering in smart factories," in *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Vienna, Austria, Sept. 2020, pp. 164–171.

[22] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[23] J. Xu, H. Wu, J. Wang, and M. Long, "Anomaly transformer: Time series anomaly detection with association discrepancy," *arXiv preprint arXiv:2110.02642*, 2021.