Computers & Industrial Engineering 203 (2025) 111023

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Anomaly detection in manufacturing systems with temporal networks and unsupervised machine learning

Giulio Mattera [a],*, Raffaele Mattera [b], Silvestro Vespoli [a], Emma Salatiello [a]

[a] University of Naples "Federico II" Department of Chemical, Materials and Industrial production Engineering, Naples, 80125, Italy
[b] Department of Social and Economic Sciences, Sapienza University of Rome, Rome, Italy

ARTICLE INFO

Keywords:
Anomaly detection
Complex networks
Unsupervised learning
Manufacturing system
Machine learning

ABSTRACT

Traditional manufacturing systems face significant challenges in detecting operational anomalies due to the absence of advanced sensor networks and intelligent machinery commonly associated with Industry 4.0. Existing solutions often rely on sophisticated, interconnected infrastructures, which are not feasible in conventional settings. This paper introduces a novel methodology for anomaly detection tailored specifically for traditional manufacturing environments, addressing the gap in cost-effective monitoring solutions. The proposed approach models manufacturing systems as complex temporal networks, where each machine or process is represented as a node and job flows between machines form the network edges over time. The novelty of this method lies in the combination of dynamic network theory with unsupervised machine learning. Statistical features extracted from the temporal networks are processed through dimensionality reduction techniques, specifically Principal Component Analysis (PCA) and Deep Neural Autoencoders, to reduce feature complexity while preserving essential information. The reduced feature sets are then analysed using multiple unsupervised anomaly detection algorithms, including Isolation Forest, One-Class Support Vector Machine (OC-SVM), and Local Outlier Factor (LOF). This approach does not require significant infrastructure upgrades, making it suitable for traditional manufacturing plants while still aligning with Industry 4.0 paradigms. By using only normal job flow data, it provides a cost-effective solution where anomalous data is scarce. The results demonstrate that Local Outlier Factor and Isolation Forest, when combined with Autoencoder-based feature reduction, achieved an F1-score exceeding 84%, with precision close to 99% and recall at 74%. This strong performance underscores the methodology's potential for real-world manufacturing environments, bridging the gap between traditional settings and modern Industry 4.0 paradigms.

1. Introduction

Rapid advancement in technology and increasing complexity of manufacturing systems have ushered in a new era of industrial production. While Industry 4.0 and intelligent manufacturing solutions offer sophisticated monitoring capabilities through sensor networks and interconnected machinery (Kusiak, 2018; Tao, Qi, Liu, & Kusiak, 2018),recent advancements in process monitoring emphasize the integration of real-time monitoring technologies, such as digital twins, IoT, and AI, to enhance production accuracy and quality across various manufacturing processes, from traditional machining to additive manufacturing (Ani, Olu-lawal, Olajiga, Montero, & Adeleke, 2024; Bertocco, Bruno, Armentani, Esposito, & Perrella, 2022; Herzog, Brandt, Trinchi, Sola, & Molotnikov, 2024; Liu, Lang, Gui, Zhu and Laalej, 2024; Mattera, Polden and Norrish, 2024; Mattera, Vozza, Polden, Nele, & Pan, 2024). These advanced methods can be developed if high-frequency sample data can be obtained once sensors are integrated into the machinery. However, many manufacturing facilities still operate with traditional equipment that lacks these advanced features. In such environments, anomalies — whether due to machine failures, process inefficiencies, or unexpected delays — can lead to significant disruptions, financial losses, and safety hazards (Li, Tao et al., 2023).

This work introduces a novel anomaly detection framework that integrates dynamic network theory, dimensionality reduction techniques, and unsupervised machine learning, specifically designed for traditional manufacturing environments where advanced sensor networks may be limited or unavailable. Our methodology transforms standard operational metrics, such as job flow data between machines, into dynamic network models that capture both structural and temporal dynamics of the manufacturing system. By extracting statistical network features and combining them with dimensionality reduction techniques such as Principal Component Analysis (PCA) and deep neural

* Corresponding author.
E-mail address: giulio.mattera@unina.it (G. Mattera).

https://doi.org/10.1016/j.cie.2025.111023
Received 30 September 2024; Received in revised form 25 January 2025; Accepted 27 February 2025
Available online 15 March 2025

autoencoders, we effectively reduce feature complexity while preserving critical structural information. This enables the detection of complex process anomalies without requiring extensive sensor upgrades or high-frequency data acquisition.

Traditional monitoring methods often fall short in capturing the complex interdependencies and temporal dynamics present in these environments (Brundage, Bernstein, Morris, & Horst, 2017), while advanced solutions such as multiscale PCA-SDG and wavelet-entropy frameworks (Ali et al., 2022; Ali, Zhang, & Gao, 2023), distributed Canonical Correlation Analysis (CCA) (Ali, Safdar et al., 2024), and dynamic ICA-distributed CCA (DICA-DCCA) (Ali, Zhang et al., 2024), require significant computational resources and specialized tools. For instance, multiscale PCA-SDG depends on wavelet transforms for multiscale fault analysis (Ali et al., 2023), while DICA-DCCA integrates dynamic Independent Component Analysis (ICA) with distributed CCA to achieve high-accuracy fault detection (Ali, Zhang et al., 2024). Our approach bridges this gap by providing an innovative and computationally efficient solution that requires only standard operational data, making it accessible to facilities with limited digital infrastructure.

One promising avenue, yet unexplored in traditional manufacturing environments, is the application of complex network theory combined with unsupervised machine learning (Becker, Meyer, & Windt, 2014; Becker & Wagner, 2016). Complex networks have been effectively used in various fields to model dynamic interactions, demonstrating their versatility and effectiveness (Bassett, Zurn, & Gold, 2018). By representing machines and processes as nodes and the interactions between them as edges, we can capture the structural and temporal characteristics of the system. The novelty lies in harnessing dynamic network models to detect anomalies, even when data sources are limited to basic operational metrics. By applying dimensionality reduction and unsupervised learning, we can identify subtle deviations in system behaviour that may indicate inefficiencies or failures, enabling a data-driven, system-wide anomaly detection framework.

Integrating machine learning techniques, particularly unsupervised learning methods, with network analysis provides a powerful approach to anomaly detection without relying on extensive labelled datasets, which are often scarce or unavailable in industrial settings (Caggiano, Napolitano, Teti, Bonini, & Maradia, 2020; Dominguez-Monferrer, Guerra-Sancho, Caggiano, Nele, Miguélez, & Cantero, 2024; Mattera, Polden, Norrish, 2024). Unlike conventional statistical process control methods or supervised learning approaches, which require labelled data to distinguish between normal and anomalous states, unsupervised algorithms can identify the underlying patterns of normal operations and detect deviations that may indicate anomalies or inefficiencies.

A key innovation of our methodology is its ability to operate effectively in traditional manufacturing setups where sensor data is either sparse or non-existent. By leveraging dynamic network models derived from basic operational metrics and enriching these representations through dimensionality reduction techniques, we achieve an anomaly detection framework that is both scalable and adaptable to real-world manufacturing environments.

Building on this foundation, our approach combines complex network theory with unsupervised machine learning to model and detect anomalies in environments where advanced imaging or sensor technology is limited. Focusing specifically on traditional manufacturing environments lacking sophisticated sensor networks, our method enables effective anomaly detection even when data are limited to standard operational metrics. Many machine learning frameworks, including distributed CCA (Ali, Safdar et al., 2024), are effective for monitoring in facilities with advanced equipment. However, our network-based approach captures structural and temporal dynamics across the manufacturing system. By focusing on the system-wide network structure and integrating advanced data transformation techniques, our approach goes beyond conventional fault detection by identifying complex interdependencies and deviations at a macro scale, reducing reliance on individual fault indicators and enhancing the robustness of the anomaly detection process.

The central innovation of this work is the development of a comprehensive anomaly detection framework tailored for traditional manufacturing environments. By combining dynamic network models with unsupervised learning, we offer a novel, scalable, and data-efficient solution for identifying process anomalies. This framework provides a significant contribution by demonstrating how statistical features derived from network models can be integrated with dimensionality reduction techniques to uncover process anomalies without the need for advanced sensor data.

In the following sections, we dive deeper into these concepts. Section 1.1 introduces the fundamentals of dynamic networks, outlining how they capture evolving relationships in complex systems. Section 1.2 discusses the application of dynamic network theory to manufacturing systems, highlighting the potential benefits and challenges. In Section 1.3, we present our proposed methodology, detailing how we effectively combine statistical network features with unsupervised machine learning and dimensionality reduction techniques to detect anomalies in traditional manufacturing settings.

1.1. Dynamic networks

Complex networks are powerful tools for representing multiple objects and their relationships through nodes and edges. This modelling technique is employed in several fields, such as global financial networks, biological systems, power grids, and manufacturing systems (Bassett et al., 2018). In manufacturing, the dynamic nature of networks is particularly relevant as the operational state of the system evolves due to changes in production schedules, machine status, and job flows.

Dynamic networks are characterized by the evolution of node properties and edge configurations over time. For example, in a manufacturing context, nodes represent machines or processes, and edges represent the flow of jobs between them. The weights of the edges can change over time, reflecting variations in job quantities, processing times, or machine availability. An example of a dynamic network with four nodes is illustrated in Fig. 1. In the example shown in Fig. 1, the edges — which characterize the relationships between nodes — change in magnitude over time, either increasing or decreasing. Similarly, we can find that the edges between two nodes disappear, or new links can be created in the network at a given time.

The dynamic nature of these complex networks makes them a valuable tool for modelling and monitoring complex systems like those associated with manufacturing processes. In the particular case of manufacturing, as shown in Fig. 2, each machine or process can be modelled as a node, with edges representing the flow of jobs between machines necessary for the production of the final product in a given amount of time (Brundage et al., 2017; Manupati, Thakkar, Wong, & Tiwari, 2013; Weise, Benkhardt, & Mostaghim, 2018). Therefore, dynamic network-based approaches offer profound insight into operational dynamics and interactions within the manufacturing system (Becker et al., 2014; Becker & Wagner, 2016).

1.2. Dynamic networks in manufacturing systems

A general manufacturing system can be represented as a network, where each machine or process is modelled as a node, and the edges represent the flow of jobs necessary to produce the final product (Brundage et al., 2017; Manupati et al., 2013; Weise et al., 2018). Despite the increasing prevalence of Artificial Intelligence (AI) and the Industry 4.0 paradigm (Kusiak, 2018; Tao et al., 2018), many manufacturing plants have not fully transitioned to these advanced technologies. This means that while individual machines may be capable of detecting their operational states using Machine Learning (ML) algorithms and real-time sensor data, it is still expected to find plants

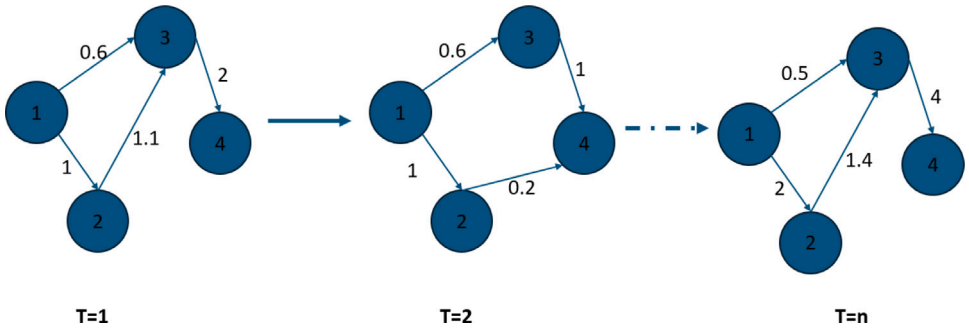


Fig. 1. Representation of a dynamic network with four nodes, where the structure of the network and its weights change over time.

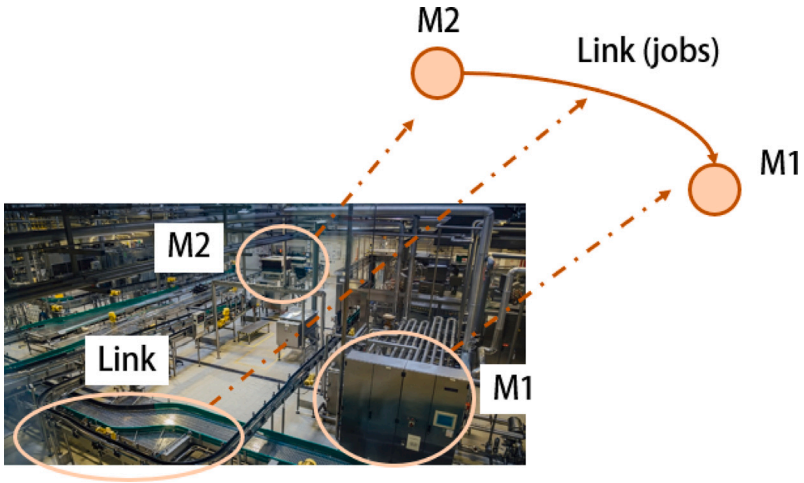


Fig. 2. A manufacturing system can be represented as a network, where each node represents a machine, and the links between nodes correspond to the number of jobs exchanged within a defined time period t.

where machines are not interconnected and cannot provide comprehensive information on the overall status of the plant (Caggiano et al., 2020; Dominguez-Monferrer et al., 2024; Mattera, Polden, Norrish, 2024). Since it is always possible to track the number of jobs exchanged between machines in a manufacturing plant, an anomaly detection network approach can be applied to provide rapid feedback on the production line without the need for human expert intervention.

In such environments, anomalies — due to unexpected delays, machine failures, or process inefficiencies — can significantly impact production outcomes. Therefore, detecting these anomalies is critical to maintaining high levels of operational efficiency, preventing costly downtime, and ensuring product quality. Traditional methods of monitoring production processes (Li, Tao et al., 2023) may struggle to capture the inherent complexity and dynamic interdependencies within the system. Instead, by representing manufacturing systems as dynamic complex networks, it becomes possible to model the flow of operations over time, monitor interactions between components, and detect unusual patterns or deviations from expected behaviour without using additional intelligent machine modules and sensors. Since anomalies can manifest as disruptions in the normal flow of jobs, unexpected changes in network structure or irregular patterns in node and edge activity can be detected using anomaly detection methods for networks. Despite this potential, no work on this approach can be found in the literature.

1.3. Proposal of the work

In this paper, we address the challenge of identifying anomalies in manufacturing systems by adopting a dynamic network perspective. By treating the manufacturing system as a complex dynamic network, we aim to leverage the network's evolving structure and behaviour to pinpoint anomalies, enabling more effective monitoring and management of production processes.

In more detail, the proposed methodology explores how dynamically extracted statistical network features can be used to identify anomalies in complex manufacturing plants through unsupervised machine-learning approaches. This unsupervised method is designed to align with traditional industrial scenarios, where the volume of data representing normal behaviour is significantly greater than that of anomalies. With this aim, we compared the results of different algorithms such as Isolation Forest (IF), Local Outlier Factor (LOF), and One-Class Support Vector Machine (OC-SVM). Furthermore, to handle the large amount of extracted statistical network features, we consider low-dimensional features as input of the aforementioned machine learning algorithms. Two approaches for dimensionality reduction are used for this aim: Principal Component Analysis and Autoencoders. This approach bridges the theoretical framework of complex dynamic networks with practical requirements, providing an effective solution for anomaly detection in conventional manufacturing settings without the need for intelligent and interconnected machinery. The key features of the proposed methodology are listed below:

- Develop a methodology that does not require significant infrastructure upgrades and can be applied to detect anomalies in line with the intelligent paradigm of Industry 4.0, even in traditional manufacturing plants.
- Innovative application of dynamic network theory to model manufacturing systems and implement anomaly detection within them.
- Utilize an unsupervised learning framework to address real industrial needs, as these datasets are often imbalanced due to the

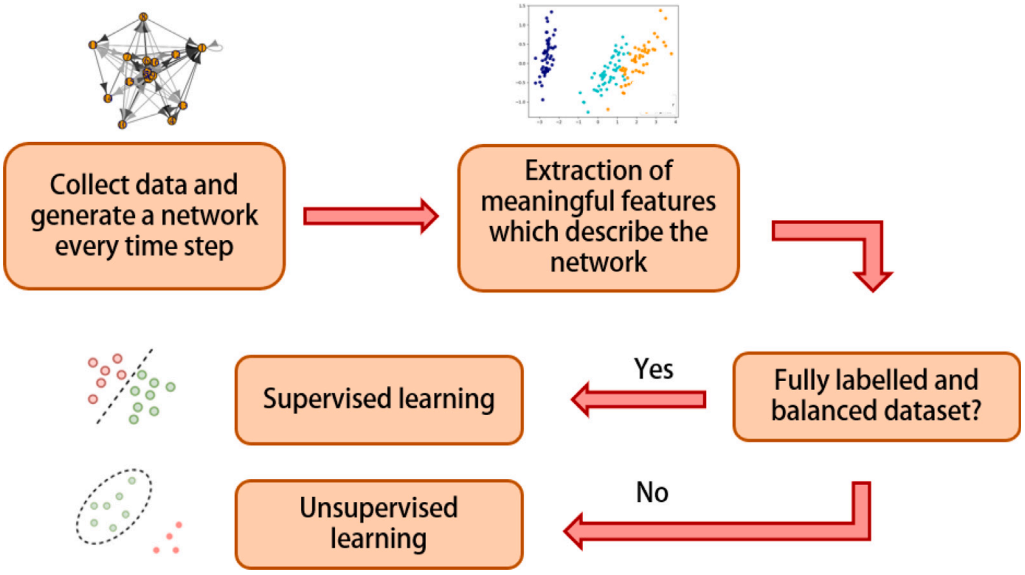


Fig. 3. Proposed procedure for anomaly detection based on Machine Learning methods.

scarcity of data associated with anomalous conditions compared to normal behaviour.

2. Related works

To better present the methodology in Section 3, we find it appropriate to two main strands of previous research. First, we briefly present related work on network anomaly detection using machine learning approaches. Second, we review papers dealing with manufacturing processes.

2.1. The role of machine learning in network anomaly detection

Anomaly detection is the process of identifying patterns in the data that deviate from the expected behaviour (Chandola, Banerjee, & Kumar, 2009). Detecting these unusual patterns is crucial in various scientific fields, as they can identify issues such as errors in manufacturing processes, suboptimal operating conditions of machines or systems of machines (Mattera, Caggiano, Caggiano et al., 2024), misconfiguration, fraud (Baesens, Höppner, & Verdonck, 2021), or other significant but rare events. Different methods have been used to develop anomaly detection tools for networked data.

Statistical methods have been widely used to define normal behaviour in data through various models, such as Principal Component Analysis (PCA) and mixture models (Ahmed, Mahmood, & Hu, 2016). In PCA, normal data inputs are transformed into a lower-dimensional space and then reconstructed; the reconstruction error is then used to evaluate anomalies. In cases involving high-dimensional data, Gaussian mixture models can be applied after extracting the principal components from the normal data (Yu et al., 2023). These models estimate the underlying distribution, allowing for the detection of principal components that deviate from the original distribution. In addition, machine learning methods can be utilized on historical data to identify patterns of normal behaviour. Deviations from these established patterns are marked as anomalies (Mattera, Polden, Norrish, 2024).

In complex networks, various types of anomalies can occur, each reflecting different aspects of network behaviour. These anomalies can be broadly categorized into three main types: vertex anomalies, edge anomalies, and, in the case of temporal network, anomalies resulting from a particular event (Ranshous et al., 2015).

An anomalous vertex refers to a node in the network that displays unusual variations in its community affiliation or role over time. For instance, if a machine begins swapping tasks/jobs with an unrelated or inappropriate machine, it could signal anomalous behaviour, such as an incorrect production sequence.

An anomalous edge is identified by unusual fluctuations or deviations in its weight or strength over time. The weight of an edge generally represents the frequency or intensity of interactions between two vertices. Abnormal variations in edge weight can indicate unexpected interactions or breakdowns within the network (Yeganeh, Chukhrova, Johannssen, & Fotuhi, 2023). For example, if a machine starts exchanging an unusually high or low number of jobs with other machines, it may suggest sub-optimal working conditions or potential malfunctions.

Finally, detecting changes or events involves identifying times in which significant deviations or alterations in network behaviour occur over time (Malinovskaya, Otto, & Peters, 2022). In this case, for the detection of abnormal behaviour, many powerful techniques based on statistical inference exist to perform network monitoring (Malinovskaya & Otto, 2021). While these deviations may not necessarily indicate a failure, they signal changes in the complex patterns of the network, potentially due to issues such as those previously mentioned.

Compared to outlier detection, which shares a similar aim, the anomaly detection task follows the philosophy of ML (Mozharovskyi, 2022). The comprehensive survey by Wang et al. (2021) highlights recent advancements in applying ML to network anomaly detection. Supervised and unsupervised learning techniques have been employed across various types of networks (Saran & Kesswani, 2023; Shahraki, Abbasi, & Haugen, 2020; Zoppi, Ceccarelli, Salani, & Bondavalli, 2020). However, deep learning methods are gaining increasing attention due to their strong generalization capabilities and high performance in tackling complex tasks, particularly those involving intricate network structures.

Regardless of whether the outlier rejection or the anomaly detection algorithm is based on statistics, supervised or unsupervised learning, feature engineering remains a crucial factor for success (Olszewski, Iwanowski, & Graniszewski, 2024). Networks are described by domain-dependent metrics that are used as input for ML algorithms. For example, Bhatia, Benno, Esteban, Lakshman, and Grogan (2019) use an Autoencoder (AE)-based classifier trained exclusively on normal traffic to detect Distributed Denial of Service (DDoS) attacks. They first extract a small set of features that represent the normal patterns of the network. Then, they compare the results of using PCA and AE to reconstruct these features when new data from the network are

Table 1
Anomaly detection related works.

Authors	Methods used	ML
Baesens et al. (2021)	Data-driven methods for fraud detection using binary classification and feature engineering, alongside robust statistics and multivariate outlier detection.	No
Ahmed et al. (2016)	Classification and statistical methods (chi-square, information theory) for mapping anomalies to attack types and clustering.	Yes
Yeganeh et al. (2023)	Machine learning-based control charts, Poisson regression, ANFIS, SVR, MLP, and ensemble methods for network surveillance and performance comparison with traditional control charts.	No
Jaramillo-Alcazar, Govea, and Villegas-Ch (2023)	Systematic review and configuration of machine learning algorithms for anomaly detection in smart manufacturing.	Yes
Latsou, Farsi, and Erkoyuncu (2023)	Hybrid agent-oriented and process-oriented methods using multi-agent simulations and UML State Machine diagrams for real-time performance monitoring.	Yes
Bhatia et al. (2019)	Unsupervised machine learning (Autoencoders, PCA) and supervised comparisons (SVM) for IoT traffic anomaly detection with retraining approaches.	No
Olszewski et al. (2024)	Utilizes Self-Organizing Maps (SOM), i-SNE, and Neighbourhood Retrieval Visualizer for data visualization in IoT cybersecurity.	Yes
Saran and Kesswani (2023)	Multiple ML classifiers (k-NN, SVM, NB, RF, DT, SGD) evaluated using the MQTT-IoT-IDS2020 dataset with confusion matrix for performance assessment.	Yes
Shahraki et al. (2020)	Boosting algorithms (Real AdaBoost, Gentle AdaBoost, Modest AdaBoost) for IDS performance enhancement using K-Fold cross-validation on benchmark datasets.	No
Mozharovskyi (2022)	Data depth methodology for multivariate anomaly detection with comparisons to existing methods.	Yes
Wang et al. (2021)	Reviews various ML techniques (SL, UL, SSL, RL) for anomaly detection across networks, discussing their strengths and limitations.	Yes

considered, thereby enhancing the capability for anomaly detection with AE. Other similar approaches involving unsupervised learning methods can be found, such as Isolation Forest (Liu, Ting, & Zhou, 2008) and the One-Class Support Vector Machine (SVM) (Li & Jung, 2023), which typically require only normal data or a mixture of standard and some uncommon cases. Also, in this case, features extracted by the network associated with normal behaviour can be used as input to these algorithms, which can then detect anomalies based on the new values of features extracted by the network. However, when a balanced dataset of both normal and anomalous cases is available, classification methods like neural networks can also be employed effectively to detect anomalies in complex networks (Jin et al., 2024).

The general framework for ML-based anomaly detection in complex networks is illustrated in Fig. 3. Once data are collected, feature extraction is essential for converting raw data into meaningful features that can be input into Machine Learning algorithms. Then, real-time data can be used to develop early warning systems that detect changes in the network before failures occur at a vertex or edge, enabling proactive maintenance.

2.2. Anomaly detection in manufacturing networks

To offer a more comprehensive comparison of the existing work in anomaly detection, we have carried out a deeper analysis, summarized in Table 1, situating our proposed methodology within the broader context of current approaches.

Baesens et al. (2021) focuses on data engineering for fraud detection, emphasizing model interpretability and proposing feature and instance engineering to improve models. Methods used include data-driven techniques for fraud detection, binary classification for suspicious transaction detection, feature engineering for performance improvement, robust statistics for detecting fraud in time series, and multivariate outlier detection using Mahalanobis distance. No machine learning applications are specified.

Ahmed et al. (2016) analyse four main anomaly detection techniques, mapping them to the main types of attacks. Methods used involve classification techniques, statistical approaches such as chi-square theory, and clustering-based methods for anomaly detection. Machine learning applications are present.

Yeganeh et al. (2023) applies control charts in retrospective and online detection phases, combining them with machine learning techniques for improved surveillance. Performance comparisons between machine learning-based and traditional statistical control charts are conducted. Methods used include machine learning-based control charts, Poisson regression for network monitoring, and techniques like ANFIS, SVR, MLP, and ensemble learning. No machine learning applications are specified.

Jaramillo-Alcazar et al. (2023) explores anomaly detection in smart manufacturing plants, focusing on IoT security and its impact on efficiency. The methods include a systematic review of IoT security and anomaly detection literature, as well as machine learning algorithm selection and configuration. Machine learning applications are present.

Latsou et al. (2023) develops a novel Digital Twin-Cyber Physical System (DT-CPS) for anomaly detection, introducing a "monitoring agent" for performance monitoring and decision-making in complex manufacturing systems. Methods used involve a hybrid agent-oriented and process-oriented approach, multi-agent simulation, UML State Machine diagram for agent behaviour, and real-time sensor data. Machine learning applications are present.

Bhatia et al. (2019) uses unsupervised machine learning to identify known and unknown anomalies, focusing on DDoS attack detection in IoT networks. Autoencoders are used to learn the normal behaviour of the IoT traffic. Methods include unsupervised learning classifiers (Autoencoders, PCA), supervised comparisons (e.g., SVM), traffic data collection from a simulated local area network, and re-training approaches for classifiers. No machine learning applications are specified.

Olszewski et al. (2024) addresses cybersecurity in IoT systems, presenting a comparative analysis of i-SNE and NeRV methods for data visualization of IoT network traffic. Methods include Self-Organizing Map (SOM), i-SNE, and Neighbourhood Retrieval Visualizer (NeRV). Machine learning applications are present.

Saran and Kesswani (2023) introduces an Intrusion Detection System (IDS) for IoT using multiple machine learning classifiers, achieving high accuracy in intrusion detection. Methods involve k-Nearest Neighbour (k-NN), Support Vector Machine (SVM), Naive Bayes (NB), Random Forest (RF), Decision Tree (DT), and Stochastic Gradient Descent

(SGD). Performance evaluation is done using the confusion matrix. Machine learning applications are present.

Shahraki et al. (2020) highlights boosting approaches for intrusion detection systems (IDS), comparing AdaBoost algorithms using five public benchmark datasets. Methods involve Real AdaBoost, Gentle AdaBoost, Modest AdaBoost, and K-fold cross-validation for performance evaluation. No machine learning applications are specified.

Mozharovskyi (2022) studies data depth for anomaly detection, discussing challenges and threshold selection for detecting anomalies. Methods include data depth techniques for multivariate anomaly detection and statistical depth values assigned to observations, with comparisons to existing methods. Machine learning applications are present.

Wang et al. (2021) presents a comprehensive survey on machine learning techniques for anomaly detection, addressing the advantages, disadvantages, and challenges of various methods. Methods include Supervised Learning (SL), Unsupervised Learning (UL), Semi-supervised Learning (SSL), and Reinforcement Learning (RL). Machine learning applications are present.

3. Methodology

3.1. System description

In formal terms, complex networks are presented as a graph $G = (V, E)$, consisting of a set of nodes $V = \{v_1, v_2, \dots, v_n\}$ and a set of edges $E = \{e_1, e_2, \dots, e_n\}$. A network can be mathematically represented through an adjacency matrix as in Eq. (1), which describes the structure and the relationship between nodes. For a network with n nodes, the adjacency matrix $\mathbf{A}$ is an $n \times n$ matrix where each element a_{ij} denotes the presence and weight of the edge between nodes i and j.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \tag{1}$$

In the case of binary representation, we have that $a_{ij} = 1$, indicates the presence of an edge between nodes i and j, while $a_{ij} = 0$ indicates no edge. This is often used for unweighted graphs. For the weighted representation, on the other hand, the value $a_{ij} \neq 0$ represents the weight of the edge between nodes i and j. If there is no edge, we simply have that $a_{ij} = 0$. This type of representation is used in systems where edges have varying strengths, such as transportation or communication networks. Finally, a special type of weighted representation is provided by count-based weights. In this representation, a_{ij} denotes the number of items exchanged between nodes i and j. This is commonly used in manufacturing systems to record the volume of item exchanges between machines. Networks can be described using various types of representations depending on their nature.

In the case of binary representation, we have that $a_{ij} = 1$, indicates the presence of an edge between nodes i and j, while $a_{ij} = 0$ indicates no edge. This is often used for unweighted graphs. For the weighted representation, on the other hand, the value $a_{ij} \neq 0$ represents the weight of the edge between nodes i and j. If there is no edge, we simply have that $a_{ij} = 0$. This type of representation is used in systems where edges have varying strengths, such as transportation or communication networks. Finally, a special type of weighted representation is provided by count-based weights. In this representation, a_{ij} denotes the number of items exchanged between nodes i and j. This is commonly used in manufacturing systems to record the volume of item exchanges between machines.

For many graphs, the matrix $\mathbf{A}$ is symmetric and has diagonal entries $a_{ii} = 0$, indicating no self-loops. An example of different networks and the associated adjacent matrix is illustrated in Fig. 4.

However, the matrix $\mathbf{A}$ does not need to be symmetric. This is the case of origin–destination matrices observed in manufacturing processes.

We remark that in this paper, we consider dynamic networks, that is, $\mathbf{A}_t$ for $t = 1, \dots, T$. For each point in time t, many features summarizing the characteristics of entire graphs can be computed from the adjacency matrices. Therefore, for complex networks, this is the main source for feature extraction. As mentioned in the introduction, the specific features extracted from these alternative representations are then used as input for anomaly detection algorithms. For large n networks, automatic feature extraction or dimensionality reduction techniques, such as PCA and autoencoders, may be employed to reduce dimensionality and use principal components or latent space representation as features to improve the performances of the machine learning methods.

3.2. Features extraction for networks

In network theory it is well known that nodes are not equivalent, that is, there exist nodes that are more central or important than other nodes. This information, contained in the so-called centrality measures, provides a useful synthesis of the network structure that can be used to identify anomalies. Therefore, we can use centrality measures to quantify the importance or influence of nodes within the network and to describe and synthesize the structure of the network. Eigenvector centrality, betweenness centrality, degree centrality, and closeness centrality are the most commonly used centrality measures. These metrics help identify the most important nodes within a network, based on the flow of information or interactions (Becker & Wagner, 2016).

The degree centrality is defined as in Eq. (2), where the degree $d(v)$ of a node v is the number of ties to other nodes.

$$C_d(v) = d(v) \tag{2}$$

Therefore, according to the degree centrality measure a node is important if it has many connections to other nodes. The degree centrality is usually normalized, so that value lies in the range from 0 to 1.

The eigenvector centrality represents the extension of the simple Degree-Centrality, which measures the influence of a node based not only on the number of direct connections it has but also on the importance of the nodes it is connected to. It is defined as the eigenvector associated with the largest eigenvalue of the adjacency matrix and is calculated using Eq. (3), where λ is a constant (eigenvalue), $\mathbf{A}$ is the adjacency matrix of the complex network, and e denotes the eigenvector.

$$e_v = \lambda^{-1} \sum_{j=1}^{n} a_{vj} e_j \tag{3}$$

Eigenvector Centrality provides a broad perspective on network influence, capturing the cumulative importance of a node based on both direct and indirect connections. The value lies in the range from 0 to 1. Furthermore, important nodes take large values.

The centrality of the interdependence, on the other hand, is calculated as reported in Eq. (4), where $g_{jk}(v)$ represents the number of shortest paths between nodes j and k which contains node v, and g_{jk} is the number of shortest paths between nodes j and k.

$$C_b(v) = \sum_{j<k} \frac{g_{jk}(v)}{g_{jk}}, \tag{4}$$

Therefore, the centrality of the networks quantifies how often a node acts as a bridge along the shortest path between two other nodes. It emphasizes the role of nodes as intermediaries, which is critical in understanding control over the flow of information or resources. Monitoring this can reveal strategic points for controlling the network, as well as potential weak points. Different to the degree of centrality, also indirect connections are considered.

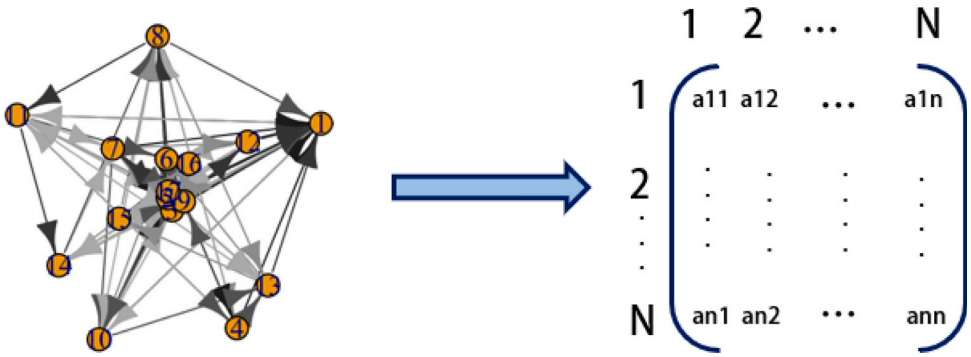


Fig. 4. An example of a network and its corresponding adjacency matrix.

Finally, the closeness centrality considers a node as important if it is connected to all other nodes on short geodesic distances, computed as in Eq. (5), here d_{vy} is the shortest path between the nodes v and y.

$$C_c(v) = \left[\sum_{y=1}^{n} d_{vy}\right]^{-1} \tag{5}$$

Also, the closeness is normalized to ensure the measures assume values between 0 and 1. The closeness Centrality reflects how easily a node can disseminate or access information throughout the network.

We stress that these measures are computed for each node; therefore, we can summarize a $n \times n$ adjacency matrix using these n features. Considering dynamic networks, the aforementioned centrality measures can be computed for each point in time. All these measures are computed at the node level for each point in time t. In other words, given an adjacency matrix at time t, $\mathbf{A}_t$, we summarize its information by computing n features and the centrality measures of the nodes at time t.

Monitoring the patterns of centrality measures over time in dynamic networks provides valuable insights into the changing roles and importance of nodes in the system. Each centrality measure captures different aspects of the network's structure, and variations in these measures can signal important shifts, including anomalies that may indicate instability or unusual events.

3.3. Dimensionality reduction techniques

The centrality measures for each node can be used as features to identify anomalies in a network. In the presence of large n networks, the number of features increases considerably, especially when many centrality measures are considered in the analysis. Dimensionality reduction techniques can, therefore, be used to reduce the number of features while maintaining essential information and structure. This approach offers several advantages in ML, such as improving performance and reducing computation time by eliminating unnecessary or redundant features, thus focusing on the most meaningful ones (Rashid et al., 2022; ur Rehman, Chang, Batool, & Wah, 2016; Zebari, Abdulazeez, Zeebaree, Zebari, & Saeed, 2020). Moreover, dimensionality reduction aids in data visualization and helps prevent overfitting in machine learning models (Bharadiya, 2023). PCA and Autoencoders architectures (Alghushairy et al., 2024; Shakya, Choudhary, & Singh, 2024) are two techniques widely used in dimensionality reduction.

On one side, the goal of PCA is to find the orthogonal transformation into lower dimension, the principal components, in the data that maximize variance while minimizing reconstruction error. It achieves this by solving an eigenvalue problem. A regularity condition for PCA is the stationarity of the data. We note, however, that under normal operating conditions, it is reasonable to assume that the manufacturing process is stationary, as the system is designed to operate under a stable production regime.

On the other hand, autoencoders (Wang, Yao, & Zhao, 2016) are a type of neural network used for learning a compressed representation (encoding) of the input data and then reconstructing the original input from this compressed representation. Autoencoders, illustrated in Fig. 5, consist of two main parts, namely the Encoder and the Decoder. The Encoder compresses the input data into a lower-dimensional representation, known as the latent space, using a non-linear function, while the Decoder is responsible for the reconstruction of the input data from the lower-dimensional representation using another non-linear transformation.

While PCA assumes that the data lies in a linear subspace, and the transformation is a linear projection of the data onto this subspace, autoencoders are non-linear and allow capture of more complex patterns in the data that cannot be represented in a linear subspace (Fan, Wang, & Zhang, 2017). Various studies emphasize that the decoder part of autoencoder architectures can work like a nonlinear PCA, which, despite being more complex to train than PCA, is better at identifying patterns in data, especially for complex systems (Caggiano, Mattera, & Nele, 2023; Hinton & Salakhutdinov, 2006).

Dimensionality reduction techniques are particularly useful for anomaly detection. In fact, once the new data is transformed using the same linear transformation fitted on normal behaviour data, the features may not be well represented in the new space, resulting in larger differences from the original data. In this scenario, dimensionality reduction can help emphasize these differences, improving anomaly detection performance (Camacho, Pérez-Villegas, García-Teodoro, & Maciá-Fernández, 2016; Harrou, Kadri, Chaabane, Tahon, & Sun, 2015; Maciá-Fernández, Camacho, García-Teodoro, & Rodríguez-Gómez, 2016).

Additionally, variants like Convolutional Autoencoders (Alsubai, Umer, Innab, Shiaeles, & Nappi, 2024; Yan, Shao, Xiao, Liu, & Wan, 2023) and Recurrent Autoencoders (Hong, Kang, Kim, & Baek, 2023; Li, Shang, Zhang, Gao and Qian, 2023) enable latent space decomposition of diverse data types, such as time series and images, allowing for more efficient extraction of spatio-temporal and spatial features from data.

3.4. Machine learning algorithms for anomaly detection

The problem studied in the paper is identifying points in time characterized by anomalies, considering the observed network features at a given point in time t. In other words, we evaluate at each point in time t if the observed network is in its target state or if it deviates from its target state (Malinovskaya & Otto, 2021) using ML approaches.

In what follows, we adopt an unsupervised learning approach to identify anomalies, which is particularly useful when labelled data is limited or unavailable. Furthermore, this method is particularly effective when the dataset contains significantly more instances of normal behaviour than anomalies or defects, making it well-suited for anomaly detection tasks. In this paper, we compare the performances of the following ML algorithms for anomaly detection: Local Outlier

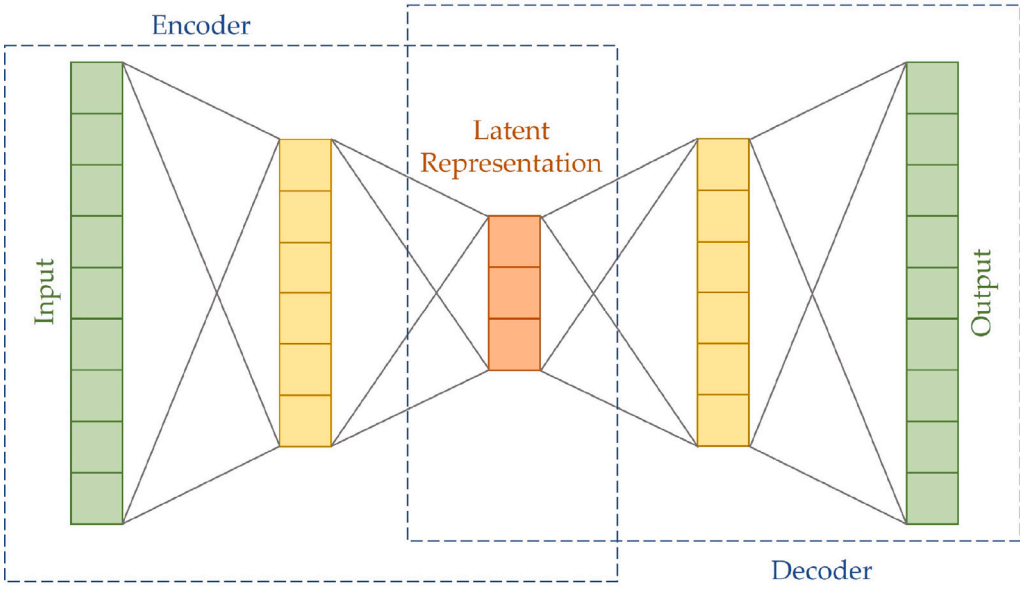


Fig. 5. A typical autoencoder architecture consists of an encoder, which extracts features into a latent space, and a decoder, which reconstructs the original input from this latent representation.

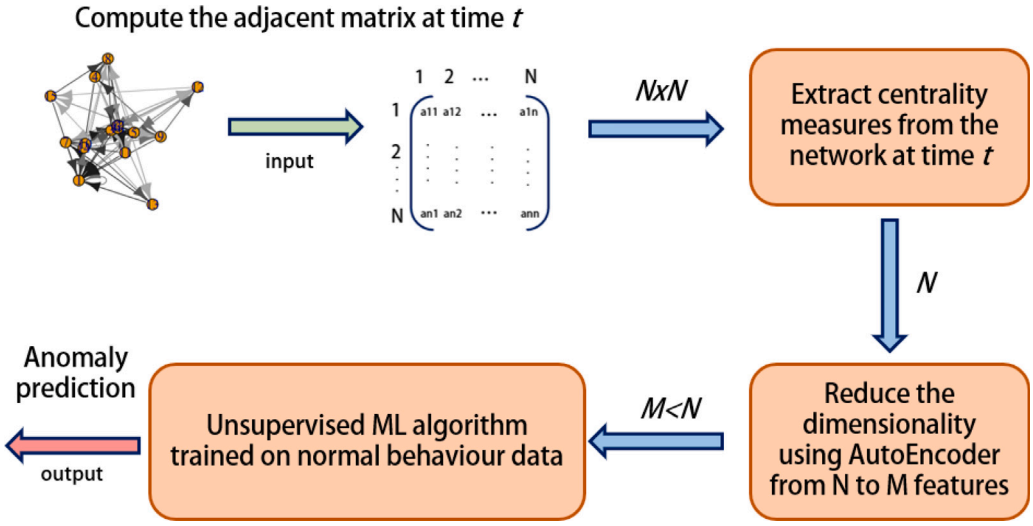


Fig. 6. Diagram of the proposed methodology: After extracting the adjacency matrix at time t, centrality measures are computed, reducing the input data from an N×N matrix to N features. These features are further reduced to M<N using dimensionality reduction techniques, which are then used as input to an unsupervised machine learning algorithm trained exclusively on normal behaviour data.

Factor (LOF), Isolation Forest (IF) and the One-Class Support Vector Machine (OC-SVM). The data workflow of the proposed methodology is illustrated in Fig. 6. In particular, once the adjacency matrix is obtained at time t, features related to centrality measures are extracted. These features are then reduced using either PCA or an autoencoder. After training the autoencoder, only the encoder part is used to reduce the number of features. These reduced features are then used as input to ML algorithms. In this study, we compared several approaches in which centrality measures are used or where the features are linearly reduced using PCA or non-linearly reduced using the encoder with different ML algorithms, as mentioned.

3.4.1. Local outlier factor

The Local Outlier Factor (LOF) (Breunig, Kriegel, Ng, & Sander, 2000) is a density-based unsupervised anomaly detection algorithm that identifies outliers based on their local density. The key concept of the LOF algorithm is that outliers exhibit significantly lower density compared to their neighbours, which is defined by a hyperparameter of the algorithm. This density is calculated based on the distances between a sample and its neighbours. The LOF score measures the degree of isolation of a data point relative to the local density of its neighbours. A score close to 1 indicates that the point's density is similar to that of its neighbours, while a score substantially greater than 1 suggests that the point is an outlier within its local neighbourhood. Although LOF provides a straightforward, density-based approach to anomaly detection, it can become computationally intensive, especially with large datasets or when the number of nearest neighbours (k) is increased

3.4.2. Isolation forest

The Isolation Forest algorithm (Liu, Ting, & Zhou, 2012) is an ensemble-based unsupervised anomaly detection technique that isolates anomalies by recursively partitioning the data. The key concept of this method is that anomalies are easier to isolate compared to normal data

points. The algorithm constructs a number of binary trees where, at each node of a tree, a feature is randomly selected, and a random split value is chosen for that feature, thereby partitioning the data into two subsets. This recursive partitioning continues until each data point is isolated in a leaf node, or a predefined tree depth is reached.

Unlike the LOF algorithm, which can benefit from data preprocessing such as standardization (which ensures that each feature contributes equally to the distance calculations), the Isolation Forest does not necessitate additional data manipulation. Its tree-based structure is inherently robust to variations in feature scales, making it particularly efficient for high-dimensional datasets without requiring explicit standardization or normalization of features.

3.4.3. One class support vector machine

The One-Class Support Vector Machine (OC-SVM) is an unsupervised learning algorithm primarily used for anomaly detection. This algorithm is designed to learn a decision boundary that best separates the majority of the data (normal instances) from the origin in the feature space, thus distinguishing between inliers (normal data) and outliers (anomalies). Unlike traditional SVMs used for binary classification, which aim to find a hyperplane separating two classes, OC-SVM focuses on separating data from the origin in high-dimensional space. Although OC-SVM can detect complex patterns in data through the use of non-linear kernels such as the RBF, its performance is highly dependent on the selection of ν and γ. These parameters often require meticulous tuning to achieve optimal results.

3.5. Evaulation metrics in anomaly detection

To assess the efficacy of anomaly detection models, a variety of metrics are often employed, including precision, recall, the F1-score, and the Precision–Recall curve. These metrics are particularly useful when dealing with datasets where anomalies are rare. Precision measures the accuracy of the model in predicting true anomalies. It is defined as the proportion of true positives (TP) relative to the sum of true positives and false positives (FP), reflecting how well the model identifies true anomalies among all flagged anomalies, that is,

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{6}$$

Recall, also referred to as sensitivity, evaluates the model's effectiveness in identifying all actual anomalies. It is the ratio of true positives to the total number of true positives and false negatives (FN), indicating the model's ability to capture all relevant anomalies,

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{7}$$

Finally, the F1-score provides a balanced measure that combines precision and recall into a single metric, offering a harmonic mean of the two. It is especially valuable when dealing with imbalanced datasets where either precision or recall might be prioritized. The F1-score is calculated as follows

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{8}$$

4. An example with simulated data

4.1. Design

In what follows, we model the industrial process network as an origin–destination matrix $\mathbf{A}_t$, where each element $a_{ij,t}$ represents the quantity exchanged from machine i to machine j at time t. The matrix $\mathbf{A}_t$ is of size $n \times n$, where n denotes the number of machines in the network. The network evolves over time, resulting in a series of matrices $\{\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_T\}$, where T is the total number of time steps considered in the simulation.

The element $a_{ij,t}$ is a count variable representing the number of units transferred from machine i to machine j during the time interval t. As such, each $a_{ij,t}$ is a non-negative integer. The origin–destination matrix thus captures the dynamic flow of materials or products across the network, reflecting the operational interactions between machines over time. By analysing the counts in the matrix $\mathbf{A}_t$, we gain insights into the volume and direction of exchanges within the system, which are crucial for optimizing industrial processes.

To model the dynamic exchange of quantities between machines in the network, we simulate a first-order Poisson autoregressive (PAR) process for each element $a_{ij,t}$ in the origin–destination matrix $\mathbf{A}_t$. This approach allows us to capture the temporal dependencies and inherent randomness in the transfer of units from machine i to machine j over time. The PAR(1) model can be expressed as (Fokianos, Rahbek, & Tjøstheim, 2009)

$$\begin{aligned} a_{ij,t} &\sim \text{Poisson}(\lambda_{ij,t}), \\ \lambda_{ij,t} &= \alpha_{ij} + \phi a_{ij,t-1}, \end{aligned} \tag{9}$$

where α_{ij} is the intercept term, representing the baseline rate of exchange between machines i and j and $\phi = 0.5$ is the autoregressive parameter, capturing the influence of the previous time step's quantity $a_{ij,t-1}$ on the current rate $\lambda_{ij,t}$. For each pair of machines (i, j), the intercept α_{ij} is simulated as $\alpha_{ij} \sim \mathcal{N}(5, 1)$. We highlight that the PAR model preserves stationarity and, therefore, all the elements of the matrix $\mathbf{A}_t$ are stationary.

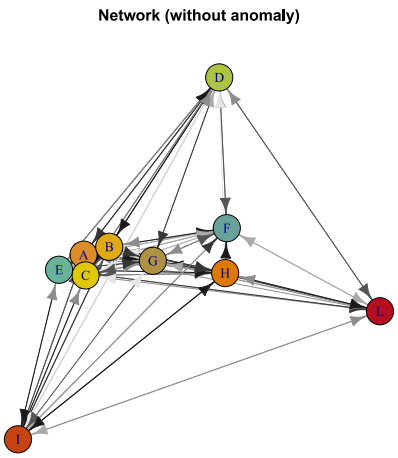
This approach ensures variability in the baseline exchange rates between different pairs of machines, reflecting the heterogeneity in their operational characteristics. The autoregressive parameter $\phi = 0.5$ introduces a temporal dependency, meaning that the quantity exchanged in the current time step is influenced by the quantity exchanged in the previous time step.

4.2. Benchmarks design

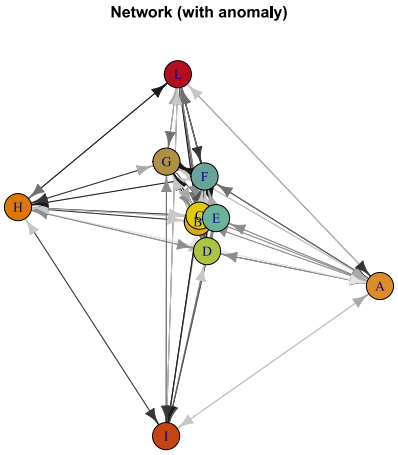
To illustrate the overall effectiveness of the methodology, we evaluated two benchmarks. The first benchmark involved a large network with $n = 50$ nodes, while the second benchmark was applied to a smaller network consisting of $n = 10$ nodes. In both cases, the simulation proceeds by iterating through time steps $t = 1, \dots, T$, where $T = 1000$. At $t = 1$, the initial quantities $a_{ij,1}$ are generated from a Poisson distribution with rate $\lambda_{ij,1} = \alpha_{ij}$, while for $t > 1$, the quantities $a_{ij,t}$ are generated using the autoregressive process with the previously simulated quantities $a_{ij,t-1}$. These simulations allow us to obtain data from only normal behaviour of the network, which are totally used to train the ML agents.

To evaluate the performance of anomaly detection, so for testing purpose, we introduce a deliberate anomaly at a specific time point, for example, $\tilde{t} = 700$. This anomaly is designed to simulate a sudden change in machine operational behaviour, reflecting potential scenarios such as malfunctions, abrupt changes in production rates, or unexpected workflow disruptions.

The anomaly is implemented by modifying the intercept parameter α_{ij} in the first-order autoregressive Poisson process. The simulation is divided into two distinct phases. In the first phase, from $t = 1$ to $t = \tilde{t}$, the quantities exchanged between machines are modelled using the standard PAR(1) process with the intercept parameter $\alpha_{ij} \sim \mathcal{N}(5, 1)$. This setup maintains the typical operational behaviour of the network. At time $\tilde{t}$, the anomaly is introduced by altering the intercept parameter. In the second phase, from $t = \tilde{t}$ to $t = T$, the intercept is $\alpha_{ij} \sim \mathcal{N}(3, 0.5)$. This change simulates a decrease in the baseline rate of exchange between machines, potentially indicating reduced efficiency or partial machine failure. By introducing this deliberate anomaly, we create a noticeable deviation from the normal operational behaviour, allowing us to assess the model's sensitivity to such changes. The goal is to test how the network responds to unexpected events and to develop methods for detecting similar anomalies in real-world industrial settings.



(a) Network structure at a given time t without anomaly



(b) Network structure at a given time $\tilde{t}$ with anomaly

Fig. 7. Example of network structures with and without anomaly.

The primary focus is on the results from the first benchmark; however, a comparative analysis of the same methodology applied to the smaller network is also provided, along with its results. Fig. 7 shows an example of two network structures at different times, for the case $n = 10$, where the edges values decrease and the network topology modifies considerably. In particular, Fig. 7(a) shows the case without anomaly and Fig. 7(b) the case with anomaly. Our aim is to identify the network structure in Fig. 7(b) as anomalous.

4.3. Results

Once the origin–destination matrix was generated as described above, various methodologies were compared, as outlined in the previous section. Specifically, three approaches are evaluated: using the extracted local statistical features of the network, the reduced features obtained applying linear dimensionality reduction via PCA, and the features obtained using nonlinear reduction through an autoencoder. These features were then used as inputs to unsupervised machine learning models, such as Isolation Forest, Local Outlier Factor, and One-Class Support Vector Machine. All the algorithms were trained using 90% of the normal behaviour data associated with the network, while the validation accuracy is evaluated on the remaining 10% of the normal behaviour data.

The autoencoder adopted for feature extraction extracts three features in the latent space that represent the associated input data. In both benchmarks, the autoencoder architecture consists of two hidden layers in the decoder, each using the ReLU activation function. The decoder follows a symmetric structure, where the first hidden layer contains half the number of features from the input layer, and the second hidden layer further reduces the dimensionality by half again. The architecture is illustrated in Fig. 8.

Similarly, PCA extracted three principal components. This reduction in the number of input features for subsequent machine learning algorithms helps to decrease computational time and hardware requirements.

4.3.1. Isolation forest

The Isolation Forest model was trained with 50 trees and a contamination rate of 0.05. It shows optimal performance when using features reduced by the autoencoder, achieving the highest F1-score of 82.35%. The model demonstrates a balanced precision of 70.0% and perfect recall of 100%, indicating its capability to detect all anomalies in the dataset, though it may have some false alarms. When using PCA-reduced features, the model's performance reduced slightly, with a precision of 95.0% and a recall of 70%. This suggests that while the model is less inclined to false alarms, its ability to detect anomalies is compromised. Overall, the autoencoder-based feature reduction offers the best performance by effectively capturing non-linear relationships in the data. Finally, using the locally extracted features significantly reduces the F1 score to 36.85%.

4.3.2. Local outlier factor

Local Outlier Factor (LOF) was trained using a Euclidean distance metric with 70 neighbours to compute the LOF score. Despite achieving perfect validation accuracy of 100% with local features, the model exhibits low precision of 45.0% and a recall of 70%, resulting in an F1-score of 54.78%. When using PCA and autoencoder-reduced features, LOF demonstrates an improved balance between precision and recall. Notably, precision increases to 85.0% with PCA-reduced features, and the F1-score improves to 76.77% with autoencoder-reduced features. Although dimensionality reduction enhances LOF's performance, it still lags behind the Isolation Forest model.

4.3.3. Once class support vector machine

OC-SVM was trained using a radial basis function kernel of order 2 with $\nu = 0.02$. The model performs poorly with local features, failing to detect any normal cases in the test dataset. However, its performance improves significantly with PCA and, especially, autoencoder features. With autoencoder features, OC-SVM achieves the highest precision of 80.0% and an F1-score of 74.6%. This demonstrates that OC-SVM benefits greatly from optimized feature extraction, with the F1-score increasing from 64.42% with PCA features to 74.6% with autoencoder features.

4.4. Summary of the results

In summary, Isolation Forest with autoencoder features offers the best overall performance for anomaly detection, excelling in F1-score, precision and recall, respectively 82.35%, 100% and 70%. LOF and OC-SVM also benefit from dimensionality reduction techniques, with

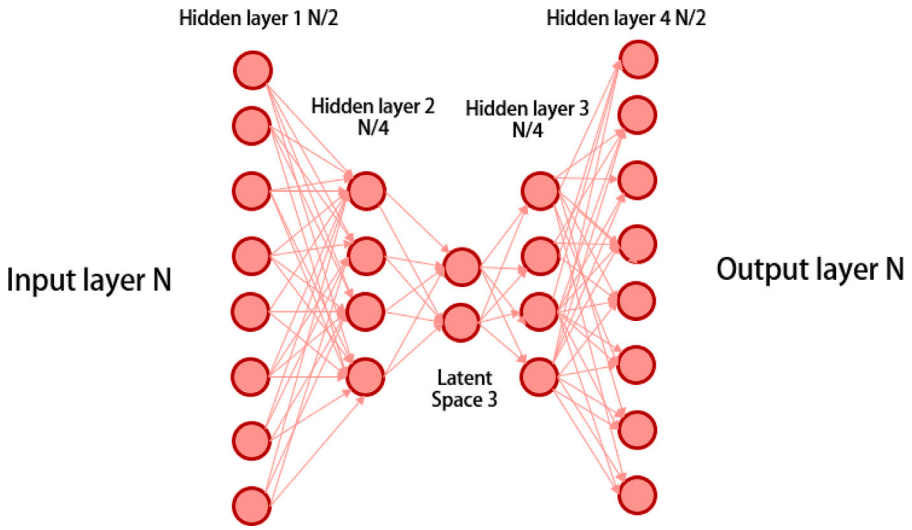


Fig. 8. The general architecture of the proposed autoencoder designed for feature extraction is outlined in this section.

Table 2
Comparison of performance metrics for different models and feature extraction techniques.

Model	Features used	Validation accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Isolation Forest	Local extracted features	94.4	70.0	25.0	36.84
	PCA	100	100.0	70.0	82.35
	Autoencoder	94.0	100.0	70.0	82.35
Local Outlier Factor	Local extracted features	100.0	70.0	45.0	54.78
	PCA	89.2	70.0	75.0	72.41
	Autoencoder	95.2	70.0	75.0	72.41
OC-SVM	Local extracted features	62.0	0.0	100.0	0.0
	PCA	54.7	45.0	70.0	54.78
	Autoencoder	92.8	70.0	84.0	76.36

the autoencoder showing the greatest improvement across models. The summary of the results is summarized in Table 2.

To investigate the reproducibility of the methodology, we reduced the dimensionality of the network used in the second benchmark, which consists of a network with 10 nodes, as discussed in previous sections. The results of the proposed methodology are summarized in Table 3, which demonstrates that the Isolation Forest method with autoencoder feature extraction outperforms the others, achieving an F1-score of 82.35% compared to 76.77% for LOF and 74.67% for OC-SVM. In both benchmarks, the statistical approach, which involves using centrality measures of nodes, results in poorer performance, underscoring the significance of dimensionality reduction for feature selection.

5. Application to job shop manufacturing system

5.1. Motivation

In traditional manufacturing environments, particularly in job-shop systems, monitoring and optimizing production processes present significant challenges (Grassi, Guizzi, Santillo, & Vespoli, 2020; Liu, Lv et al., 2024; Wang, Li, Jiao, & Ma, 2024). Job shops are characterized by a high degree of variability and complexity due to the customization of products and the flexible routing of jobs through different machines. Each job, in fact, may require a unique sequence of processing steps, leading to a complex web of interactions among machines and processes (Salatiello, Guizzi, Marchesano, & Santillo, 2022; Shi & Xiong, 2024).

Representing a job-shop manufacturing system as a dynamic complex network (see Fig. 9) provides an interesting framework for analysing these interactions and the proposed method. In this network, each machine or workstation is modelled as a node, and the flow of jobs between machines over time is represented by directed edges. The weights of these edges at any given time t, denoted as $a_{ij,t}$, quantify the number of jobs moving from machine i (the origin) to machine j (the destination). This formulation effectively constructs an origin–destination (OD) matrix $\mathbf{A_t}$ at each time point, capturing the real-time dynamics of the job flows within the system.

By analysing these OD matrices, insights into the operational patterns and dependencies within the job-shop can be obtained. Anomalies such as unexpected delays, machine failures, or bottlenecks manifest as deviations in the expected flow of jobs, which can be detected by monitoring changes in the network's structure and the evolution of the above-mentioned OD matrix. This network-based approach enables proactive identification of issues, facilitating timely interventions to maintain optimal production efficacy, without the need for human expert intervention. Furthermore, the proposed methodology is applicable also to traditional manufacturing plants, in which intelligent and interconnected machines are not present.

5.2. The role of discrete event simulation

Discrete Event Simulation (DES) is a valuable tool for modelling and analysing complex manufacturing systems (Marchesano, Guizzi, Vespoli, & Ferruzzi, 2023) like job shops. DES models the operation of a system as a chronological sequence of events, where each event occurs at a particular point in time and marks a change of state in the system. In the context of a job shop, events include the arrival of jobs, the completion of processing tasks, machine failures, and repairs. DES is widely used in manufacturing and logistics since acting as a Digital Twin (Nele, Mattera, Yap, Vozza, & Vespoli, 2024), it allows the analysis of the system's behaviour under various conditions and

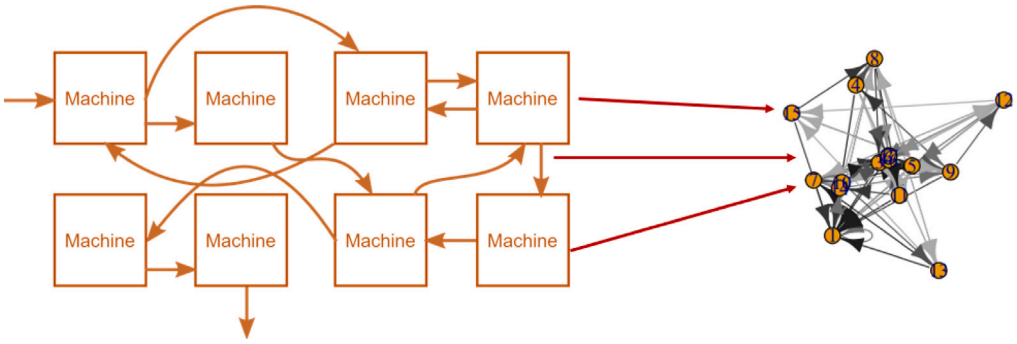


Fig. 9. Job shops are characterized by a variety of machines through which jobs can follow multiple pathways, making them well-suited for representation as networks using a node-edge approach, typically structured in an adjacency or origin–destination matrix.

Table 3
Comparison of performance metrics for different models and feature extraction techniques for the benchmark 2.

Model	Features used	Validation accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Isolation Forest	Local extracted features	94.4	70.00	39.0	50.09
	PCA	100.0	100.00	70.0	82.35
	Autoencoder	93.6	100.0	70.0	82.35
Local Outlier Factor	Local extracted features	100.0	70.00	45.0	54.78
	PCA	89.2	75.00	70.0	72.41
	Autoencoder	87.6	85.0	70.0	76.77
OC-SVM	Local extracted features	76.4	0.0	100.0	0.0
	PCA	63.2	70.00	45.0	54.78
	Autoencoder	93.6	80.0	70.0	74.67

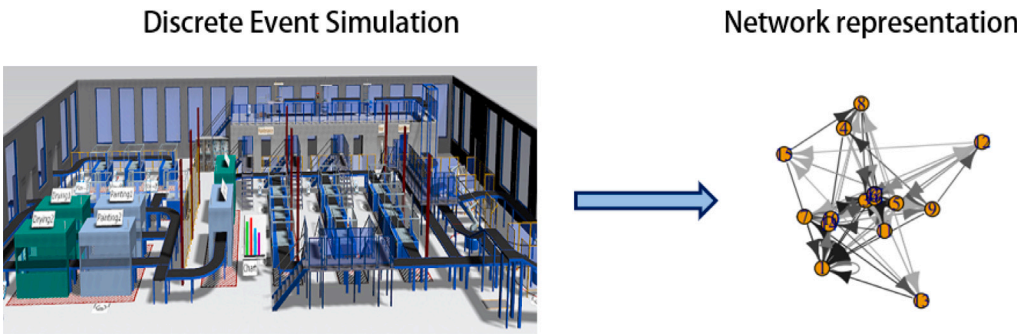


Fig. 10. A Digital Twin of a production plant can be analysed using Discrete Event Simulation and subsequently examined through complex network tools.

optimization processes by predicting system performance (Wang, Peng, Nassehi, & Tang, 2023).

Moreover, DES is useful to generate augmented data, referring to synthetic data generated through simulations to supplement real-world data. Since real-world data may not cover all possible scenarios, especially rare events like equipment failures or extreme workload conditions, augmented data can be generated in this sense, helping ML models learn to predict and respond to such situations more effectively. Moreover, Scenario simulations help test the performance of ML algorithms in different hypothetical situations. This ensures that the models can respond correctly to anomalies, unexpected workloads, or unusual operating conditions (Hummel & Schuhmacher, 2023; Monek & Fischer, 2023). Fig. 10 illustrates how a network representation can be derived from a DES model using an adjacency matrix, as discussed in the previous section. This allows augmented data to be used for analysing the performance and behaviour of a production plant, identifying anomalous conditions, and generating data to train machine learning algorithms.

By employing DES, we can create a virtual representation of the job shop that captures the stochastic nature of production processes, including variability in processing times, machine availability, and job routing. This simulation provides time-stamped data on job movements between machines, which can be aggregated into OD matrices $\mathbf{A}_t$ for each time interval.

In the proposed methodology, DES serves multiple purposes:

- **Data Generation**: It produces realistic operational data reflecting both typical and atypical conditions in the job shop, including anomalies that may be rare or difficult to observe in real-world settings.
- **Scenario Analysis**: We can introduce specific disruptions, such as machine failures or changes in job priorities, to study their impact on the system and test the robustness of our anomaly detection methods.

- **Validation**: The simulated data allows for controlled experimentation, enabling us to validate the effectiveness of our network-based anomaly detection approach before deployment in actual manufacturing environments.

5.3. Modelling the job shop as a dynamic network

In the proposed approach, we model the job shop as a time-evolving directed network. At each time interval t, we evaluate an adjacency matrix $\mathbf{A}_t$ where each element $a_{ij,t}$ represents the number of jobs transferred from machine i to machine j. This adjacency matrix functions as an OD matrix, capturing the flow of jobs between all pairs of machines within the system.

Mathematically, the adjacency matrix $\mathbf{A}_t$ is defined as:

$$\mathbf{A}_t = \begin{bmatrix} a_{11,t} & a_{12,t} & \cdots & a_{1n,t} \\ a_{21,t} & a_{22,t} & \cdots & a_{2n,t} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1,t} & a_{n2,t} & \cdots & a_{nn,t} \end{bmatrix},$$

where n is the number of machines in the job shop. Since self-loops (jobs processed consecutively on the same machine without transfer) may occur, the diagonal elements $a_{ij,t}$ can be non-zero.

This network representation allows us to capture temporal dynamics by constructing $\mathbf{A}_t$ for successive time intervals, creating a sequence of networks that reflect the evolving state of the job shop. The OD matrices reveal how jobs traverse the system, highlighting common pathways and identifying critical machines that serve as hubs or bottlenecks. As described in the methodology section, an unsupervised learning approach has been employed. This means that only the data associated with normal behaviour are used during the training phase, while anomalies are detected as deviations from complex hidden normality patterns.

5.4. Case study: Simulation of a job shop environment

To validate the proposed methodology, we conducted a case study using a simulated job-shop environment modelled through DES. The simulator, developed in AnyLogic, is designed to be highly generalizable and accommodate various configurations of job shops by adjusting key input parameters. This generality ensures that the methodology can be applied to a wide range of manufacturing systems, reflecting the diversity and complexity found in real-world production environments.

In this case study, we consider a job shop consisting of $M = 10$ machines and $J = 15$ different types of jobs. The selection of 10 machines and 15 job types strikes a balance between complexity and computational tractability, allowing us to capture intricate interactions present in a typical job shop while keeping the simulation manageable. The simulation requires three primary input parameters: the interarrival times of jobs, the processing times for each job on each machine, and the routing sequences of jobs through the machines.

The interarrival times of the jobs are represented by the vector $\lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_J\}$, where λ_j is the mean time between arrivals of jobs of type j. These values are critical in modelling the workload and flow of jobs into the system. For this case study, we set:

$$\lambda = \{15, 22, 18, 25, 30, 35, 28, 40, 32, 38, 20, 26, 24, 36, 34\} \text{ min.}$$

These interarrival times were chosen to reflect a variety of demand rates across different job types, simulating the diversity of products and orders that a real job shop might handle. By incorporating a range of interarrival times, we can model both high-frequency and low-frequency job arrivals, which is essential for testing the robustness of the anomaly detection methodology under varying load conditions.

Each job type has a specific sequence of machines it must visit, represented by the routing matrix $\mathbf{R}$, where each row $\mathbf{R}_j$ corresponds to job type j and lists the machines in the order they are visited. The routing sequences are designed to vary in length and composition, capturing the flexibility and customization inherent in job shop operations. The routing matrix is given by:

$$\mathbf{R} = \begin{bmatrix} 2 & 5 & 7 & 10 & \\ 3 & 6 & 2 & 9 & 4 \\ 1 & 4 & 8 & 5 & \\ 5 & 3 & 7 & 2 & 6 \\ 6 & 1 & 9 & 4 & 8 \\ 10 & 2 & 5 & 3 & \\ 7 & 4 & 1 & 8 & 6 \\ 2 & 6 & 10 & 3 & 9 \\ 9 & 5 & 2 & 7 & \\ 8 & 3 & 6 & 1 & 5 \\ 4 & 7 & 10 & 5 & 2 \\ 1 & 9 & 2 & 6 & 4 \\ 3 & 8 & 5 & 7 & \\ 2 & 7 & 4 & 9 & 1 \\ 5 & 1 & 3 & 6 & 8 \end{bmatrix}.$$

A graphical representation of the job shop routing network is shown in Fig. 11. This figure illustrates the flow of jobs from the "W-RM" (Warehouse for Raw Materials) through the sequence of machines, and finally to the "W-FP" (Warehouse for Finished Products), highlighting the progression of jobs through the system.

The processing times are represented by the matrix $\mathbf{P}$, where each row $\mathbf{P}_j$ contains the processing times for each operation (machine) in the routing of job type j. These processing times are given in minutes and were selected to introduce variability in the duration of operations, which is characteristic of manufacturing processes due to differences in task complexity and machine capabilities. The processing times are set as:

$$\mathbf{P} = \begin{bmatrix} 15 & 20 & 25 & 18 & \\ 22 & 18 & 20 & 15 & 25 \\ 20 & 25 & 18 & 22 & \\ 25 & 15 & 22 & 20 & 18 \\ 18 & 22 & 20 & 25 & 15 \\ 20 & 18 & 25 & 22 & \\ 22 & 20 & 18 & 25 & 15 \\ 15 & 25 & 20 & 18 & 22 \\ 25 & 18 & 22 & 20 & \\ 18 & 20 & 22 & 25 & 15 \\ 20 & 25 & 15 & 18 & 22 \\ 22 & 18 & 25 & 20 & 15 \\ 15 & 22 & 18 & 20 & \\ 20 & 25 & 22 & 18 & 15 \\ 25 & 20 & 15 & 22 & 18 \end{bmatrix}.$$

In the simulation, jobs of type j arrive according to a Poisson process with mean inter-arrival time λ_j, capturing the randomness of job arrivals in a real production setting. The processing times for each operation are modelled as exponentially distributed random variables with means given by $\mathbf{P}_j$, reflecting the stochastic nature of processing durations due to factors such as machine variability and operator efficiency. This stochastic modelling is essential to accurately represent the inherent uncertainties in manufacturing processes.

5.4.1. Analysing anomalous scenario via unsupervised learning

The simulation runs for a period of $T = 2$ years of simulated time, with data collected at regular intervals of $\Delta t = 4$ h. This results in a total of $N = \frac{2\times365\times24}{4} = 4380$ time intervals. At each time interval t, we construct the adjacency matrix $\mathbf{A}_t$, representing the number of jobs transferred between machines during that interval. This temporal granularity allows us to capture the dynamics of the system and observe the evolution of job flows over time, which is essential for detecting anomalies that may manifest as changes in the network's structure. In this first scenario, data associated with the normal behaviour of the network has been collected with the aim of training the unsupervised learning algorithms.

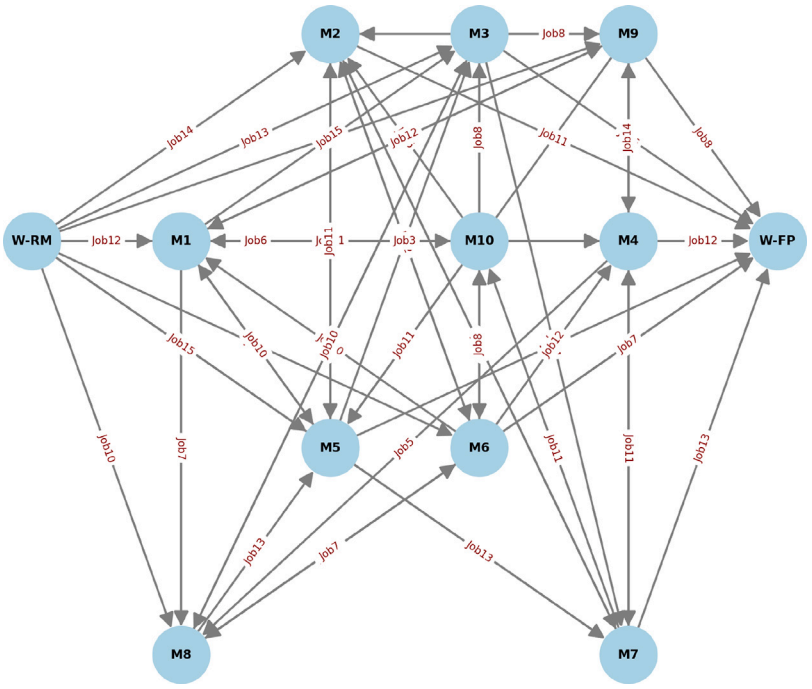


Fig. 11. Job Shop Routing Network with W-RM (Raw Material Warehouse) and W-FP (Finished Product Warehouse).

To test the anomaly detection capabilities of the proposed methodology, we introduced a second scenario in which anomalies at random times occur during the simulation. In this case, a machine failure or the transportation mechanism between machine failure (e.g. due to bridge crane or AGV breakage) are common issue in manufacturing, therefore this failures have been simulated. As a result, jobs requiring the specific machine are delayed, affecting the flow of jobs and altering the structure of the network. The impact of this anomaly is expected to propagate through the network, as failure of a single machine can create bottlenecks and affect subsequent operations. By incorporating them into the simulation, we can evaluate the effectiveness of anomaly detection methods in identifying and responding to various types of operational deviations. The anomalies affect different aspects of the system, such as equipment availability, demand levels, and process routing, providing a comprehensive test of the robustness of the methodology.

5.4.2. *Features extraction and hyperparameters*

As mentioned in the previous section, centrality measures have been collected as statistical network features. Then, dimensionality reduction is used to reduce the dimensionality. To select the number of features to reduce, a variance analysis has been performed, as shown in Fig. 12, in which it is shown that 5 principal components are sufficient to explain most of the variance of the network features.

Then the PCA algorithm extracts 5 PCs as well as the autoencoder, which is used to extract 5 features in the latent space. The autoencoder possesses an architecture of 36, 18, and 5 neurons in 3 levels for both encoder and decoder structure, which is symmetric, with ReLU activation functions. It has been trained with the Adam backpropagation algorithm with a learning rate of $\alpha = 0.001$. The following unsupervised learning methods employed to process the extracted features have the same hyperparameters employed for the simulated case study of the previous section.

5.5. *Results and discussion*

Using the collected data, we extracted network features at each time interval, focusing on centrality measures and other relevant statistics.

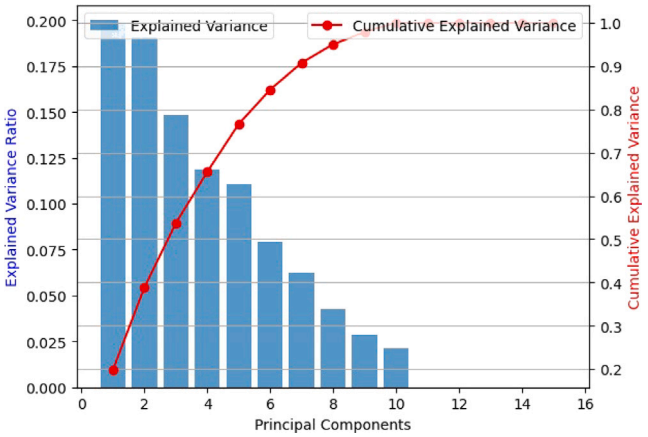


Fig. 12. Explained variance via Pareto diagram of the normal network behaviour.

We observed that under normal operating conditions, the network exhibits stable patterns with gradual fluctuations due to the inherent variability in the system.

When anomalies were introduced, significant deviations in the network features were detected, as illustrated in Fig. 13. The breakdown of a machine led to abrupt changes in the degree and centrality of the relationship between adjacent machines, as jobs were rerouted or delayed. Similar results are reached by the break of the transportation mechanism between two machines of the network. However, this modification in the network structure are emphasized using feature compression techniques, as presented in the previous sections.

For example, due to the machine breakage, a node can disappear from the network representation, whereas neighbouring machines showed modified values of job exchange rate due to the redistribution of workload. Additionally, potential misrouted jobs may create unexpected edges in the network, altering the adjacency matrices.

The application of dimensionality reduction techniques helped in managing the high-dimensional feature space and emphasized the differences between normal and anomalous states. The autoencoder, in

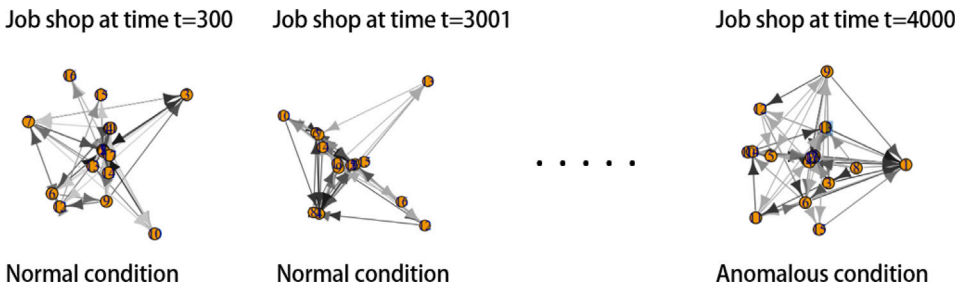


Fig. 13. A Digital Twin of a production plant can be analysed using Discrete Event Simulation and subsequently examined through complex network tools.

Table 4
Comparison of performance metrics for different models and feature extraction techniques for the Job Shop proposed in this study.

Model	Features used	Validation accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Isolation Forest	PCA	94.52	94.06	73.85	82.74
	Autoencoder	98.81	98.73	73.80	84.47
Local Outlier Factor	PCA	99.36	99.38	73.85	84.73
	Autoencoder	99.63	99.51	73.78	84.74
OC-SVM	PCA	97.17	96.88	73.75	83.74
	Autoencoder	95.53	94.56	73.82	82.91

particular, captured non-linear relationships and provided a compact representation of the data, given the complex hidden relationship.

Also, in this case, the performance of the models were evaluated using precision, recall and F1-score metrics, and the results are summarized in Table 4. The results indicate that the F1-score exceeds 84% when using the autoencoder for dimensionality reduction alongside the LOF or Isolation Forest algorithms. As noted in the previous section, the OC-SVM algorithm achieves the lowest performance, with an F1-score of 83.74%, although it remains comparable. Therefore, this study demonstrates that employing feature selection methodologies through dimensionality reduction combined with unsupervised learning models effectively identifies anomalies with good accuracy. Nonetheless, further improvements are needed to enhance performance, making this method more applicable in industrial contexts.

In general, the case study validates the proposed methodology, demonstrating its applicability to monitoring job-shop manufacturing systems and detecting anomalies that could impact production efficiency, also in traditional manufacturing plants without requiring interconnected and intelligent machines able to self-detection capabilities.

5.6. Summary, limitations and future perspective

In this study, we demonstrate that ML can be utilized effectively for anomaly detection in manufacturing systems, achieving approximately 84% performance through statistical feature extraction and dimensionality reduction techniques within an unsupervised framework. This approach offers several advantages:

- First, the proposed methodology does not require both anomalous and normal data; it suffices to have only normal data or a highly imbalanced dataset, thus reducing the costs associated with dataset generation and labelling.
- Second, this method is applicable to both traditional and non-intelligent manufacturing systems, where it is sufficient to monitor the number of jobs exchanged between machines during a specific work shift or portion of it.
- Therefore, this method has the potential to enhance traditional systems, enabling them to recognize anomalies occurring on the production line and notify operators for checks and predictive maintenance to address malfunctions.
- Finally, the information gathered can be leveraged for subsequent rescheduling algorithms based on innovative methods such as Reinforcement Learning, thereby increasing the volume of available data for informed decision-making.

While the results obtained are promising, further improvements are necessary to develop algorithms that can be trusted in real industrial environments. Achieving this could involve implementing various techniques, such as Convolutional AutoEncoders (CAE), which are capable of directly processing spatial information from the origin–destination matrix. This approach would eliminate the need for manually extracted features that may not adequately represent the network, thereby leveraging automatic feature extraction.

Additionally, rather than focusing solely on 1-time step delta matrices, it could be useful to consider more temporal relationships over time. Utilizing Recurrent Convolutional Networks (R-CNN) can facilitate the automatic extraction of features while taking into account preceding time steps.

Therefore, future analyses will explore these advanced techniques for analysing complex time series data within manufacturing systems, similar to the approach taken in this study.

6. Conclusions

This paper presented a novel methodology that bridges the gap between complex network theory and practical anomaly detection in traditional manufacturing systems. By modelling manufacturing processes as dynamic networks, we captured the temporal and structural nuances of job flows between machines. The extraction of statistical network features enabled us to quantify these dynamics effectively.

The proposed approach demonstrated that unsupervised machine learning techniques could be successfully applied in environments where intelligent and interconnected machinery is absent, and only normal operational data are available. The use of dimensionality reduction methods proved essential in enhancing the performance of anomaly detection algorithms. Specifically, we achieved an F1-score exceeding 84% when applying Isolation Forest and Local Outlier Factor algorithms to the reduced feature by the autoencoder. The high recall of 99% demonstrates that the model is highly effective at detecting nearly all anomalies in the data. However, the precision of 74% indicates the presence of some false positives among the identified anomalies. This trade-off can be beneficial, especially when the goal is to catch

anomalies as early as possible, which is often crucial in manufacturing systems.

The significance of this work lies in its applicability to conventional manufacturing settings, offering a cost-effective and efficient solution for anomaly detection without the need for additional sensors or advanced communication infrastructure. By providing a means to monitor and detect anomalies based on existing data, our methodology can help improve operational efficiency, reduce downtime, and prevent potential failures in manufacturing systems. Moreover, the integration of complex network modelling with machine learning opens new avenues for research and development in industrial analytics. It opens the door for more advanced applications, such as predictive maintenance, dynamic scheduling, and real-time optimization of manufacturing processes.

Future research will focus on extending this methodology to larger and more complex manufacturing systems, as well as exploring the integration of other machine learning techniques, such as deep learning models capable of capturing even more intricate patterns in the data (e.g. the automatic feature extraction capability of Convolutional AutoEncoders, which can act directly on the spatial and temporal structure of the origin–destination matrix).

In conclusion, the proposed methodology represents a significant step toward enhancing anomaly detection in manufacturing systems, contributing to the advancement of smart manufacturing and Industry 4.0 initiatives, even in environments that have yet to fully adopt these paradigms.

CRediT authorship contribution statement

Giulio Mattera: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Raffaele Mattera:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Silvestro Vespoli:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Emma Salatiello:** Writing – review & editing, Writing – original draft, Visualization, Methodology, Investigation, Data curation, Conceptualization.

Data availability

Data will be made available on request.

References

Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, *60*, 19–31.

Alghushairy, O., Alsini, R., Alhassan, Z., Alshdadi, A. A., Banjar, A., Yafoz, A., et al. (2024). An efficient support vector machine algorithm based network outlier detection system. *IEEE Access*.

Ali, H., Maulud, A. S., Zabiri, H., Nawaz, M., Suleman, H., & Taqvi, S. A. A. (2022). Multiscale principal component analysis-signed directed graph based process monitoring and fault diagnosis. *ACS Omega*, *7*(11), 9496–9512.

Ali, H., Safdar, R., Zhou, Y., Yao, Y., Yao, L., Zhang, Z., et al. (2024). Robust statistical industrial fault monitoring: A machine learning-based distributed CCA and low frequency control charts. *Chemical Engineering Science*, *299*, Article 120460.

Ali, H., Zhang, Z., & Gao, F. (2023). Multiscale monitoring of industrial chemical process using wavelet-entropy aided machine learning approach. *Process Safety and Environmental Protection*, *180*, 1053–1075.

Ali, H., Zhang, Z., Safdar, R., Rasool, M. H., Yao, Y., Yao, L., et al. (2024). Fault detection using machine learning based dynamic ICA-distributed CCA: Application to industrial chemical process. *Digital Chemical Engineering*, *11*, Article 100156.

Alsubai, S., Umer, M., Innab, N., Shiaeles, S., & Nappi, M. (2024). Multi-scale convolutional auto encoder for anomaly detection in 6g environment. *Computers & Industrial Engineering*, *194*, Article 110396.

Ani, E. C., Olu-lawal, K. A., Olajiga, O. K., Montero, D. J. P., & Adeleke, A. K. (2024). Intelligent monitoring systems in manufacturing: current state and future perspectives. *Engineering Science & Technology Journal*, *5*(3), 750–759.

Baesens, B., Höppner, S., & Verdonck, T. (2021). Data engineering for fraud detection. *Decision Support Systems*, *150*, Article 113492.

Bassett, D. S., Zurn, P., & Gold, J. I. (2018). On the nature and use of models in network neuroscience. *Nature Reviews. Neuroscience*, *19*(9), 566–578.

Becker, T., Meyer, M., & Windt, K. (2014). A manufacturing systems network model for the evaluation of complex manufacturing systems. *International Journal of Productivity and Performance Management*, *63*(3), 324–340.

Becker, T., & Wagner, D. (2016). Identification of key machines in complex production networks. *Procedia CIRP*, *41*, 69–74.

Bertocco, A., Bruno, M., Armentani, E., Esposito, L., & Perrella, M. (2022). Stress relaxation behavior of additively manufactured polylactic acid (PLA). *Materials*, *15*(10), 3509.

Bharadiya, J. P. (2023). A tutorial on principal component analysis for dimensionality reduction in machine learning. *International Journal of Innovative Science and Research Technology*, *8*(5), 2028–2032.

Bhatia, R., Benno, S., Esteban, J., Lakshman, T., & Grogan, J. (2019). Unsupervised machine learning for network-centric anomaly detection in IoT. In *Proceedings of the 3rd acm conext workshop on big data, machine learning and artificial intelligence for data communication networks* (pp. 42–48).

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on management of data* (pp. 93–104).

Brundage, M. P., Bernstein, W. Z., Morris, K. C., & Horst, J. A. (2017). Using graph-based visualizations to explore key performance indicator relationships for manufacturing production systems. *Procedia Cirp*, *61*, 451–456.

Caggiano, A., Mattera, G., & Nele, L. (2023). Smart tool wear monitoring of CFRP/CFRP stack drilling using autoencoders and memory-based neural networks. *Applied Sciences*, *13*(5), 3307.

Caggiano, A., Napolitano, F., Teti, R., Bonini, S., & Maradia, U. (2020). Advanced die sinking EDM process monitoring based on anomaly detection for online identification of improper process conditions. *Procedia CIRP*, *88*, 381–386.

Camacho, J., Pérez-Villegas, A., García-Teodoro, P., & Maciá-Fernández, G. (2016). PCA-based multivariate statistical network monitoring for anomaly detection. *Computers & Security*, *59*, 118–137.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, *41*(3), 1–58.

Dominguez-Monferrer, C., Guerra-Sancho, A., Caggiano, A., Nele, L., Miguélez, M. H., & Cantero, J. L. (2024). Multiresolution analysis for tool failure detection in CFRP/Ti6Al4V hybrid stacks drilling in aircraft assembly lines. *Mechanical Systems and Signal Processing*, *206*, Article 110925.

Fan, J., Wang, W., & Zhang, H. (2017). AutoEncoder based high-dimensional data fault detection system. In *2017 ieee 15th international conference on industrial informatics (indin)* (pp. 1001–1006). IEEE.

Fokianos, K., Rahbek, A., & Tjøstheim, D. (2009). Poisson autoregression. *Journal of the American Statistical Association*, *104*(488), 1430–1439.

Grassi, A., Guizzi, G., Santillo, L. C., & Vespoli, S. (2020). A semi-heterarchical production control architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters*, *24*, 43–46.

Harrou, F., Kadri, F., Chaabane, S., Tahon, C., & Sun, Y. (2015). Improved principal component analysis for anomaly detection: Application to an emergency department. *Computers & Industrial Engineering*, *88*, 63–77.

Herzog, T., Brandt, M., Trinchi, A., Sola, A., & Molotnikov, A. (2024). Process monitoring and machine learning for defect detection in laser-based metal additive manufacturing. *Journal of Intelligent Manufacturing*, *35*(4), 1407–1437.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504–507.

Hong, S., Kang, M., Kim, J., & Baek, J. (2023). Sequential application of denoising autoencoder and long-short recurrent convolutional network for noise-robust remaining-useful-life prediction framework of lithium-ion batteries. *Computers & Industrial Engineering*, *179*, Article 109231.

Hummel, V., & Schuhmacher, J. (2023). Method for semi-automated improvement of smart factories using synthetic data and cause-effect-relationships. *ESSN: 2701-6277*, 371–381.

Jaramillo-Alcazar, A., Govea, J., & Villegas-Ch, W. (2023). Anomaly detection in a smart industrial machinery plant using IoT and machine learning. *Sensors*, *23*(19), 8286.

Jin, M., Koh, H. Y., Wen, Q., Zambon, D., Alippi, C., Webb, G. I., et al. (2024). A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Kusiak, A. (2018). Smart manufacturing. *International Journal of Production Research*, *56*(1–2), 508–517.

Latsou, C., Farsi, M., & Erkoyuncu, J. A. (2023). Digital twin-enabled automated anomaly detection and bottleneck identification in complex manufacturing systems using a multi-agent approach. *Journal of Manufacturing Systems*, *67*, 242–264.

Li, G., & Jung, J. J. (2023). Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges. *Information Fusion*, *91*, 93–102.

Li, W., Shang, Z., Zhang, J., Gao, M., & Qian, S. (2023). A novel unsupervised anomaly detection method for rotating machinery based on memory augmented temporal convolutional autoencoder. *Engineering Applications of Artificial Intelligence*, *123*, Article 106312.

Li, Y., Tao, Z., Wang, L., Du, B., Guo, J., & Pang, S. (2023). Digital twin-based job shop anomaly detection and dynamic scheduling. *Robotics and Computer-Integrated Manufacturing*, *79*, Article 102443.

Liu, Z., Lang, Z.-Q., Gui, Y., Zhu, Y.-P., & Laalej, H. (2024). Digital twin-based anomaly detection for real-time tool condition monitoring in machining. *Journal of Manufacturing Systems*, *75*, 163–173.

Liu, M., Lv, J., Du, S., Deng, Y., Shen, X., & Zhou, Y. (2024). Multi-resource constrained flexible job shop scheduling problem with fixture-pallet combinatorial optimisation. *Computers & Industrial Engineering*, *188*, Article 109903.

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. In *2008 eighth ieee international conference on data mining* (pp. 413–422). IEEE.

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2012). Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *6*(1), 1–39.

Maciá-Fernández, G., Camacho, J., García-Teodoro, P., & Rodríguez-Gómez, R. A. (2016). Hierarchical PCA-based multivariate statistical network monitoring for anomaly detection. In *2016 IEEE international workshop on information forensics and security* (pp. 1–6). IEEE.

Malinovskaya, A., & Otto, P. (2021). Online network monitoring. *Statistical Methods & Applications*, *30*(5), 1337–1364.

Malinovskaya, A., Otto, P., & Peters, T. (2022). Statistical learning for change point and anomaly detection in graphs. In *Artificial intelligence, big data and data science in statistics: challenges and solutions in environmetrics, the natural sciences and technology* (pp. 85–109). Springer.

Manupati, V. K., Thakkar, J. J., Wong, K. Y., & Tiwari, M. K. (2013). Near optimal process plan selection for multiple jobs in networked based manufacturing using multi-objective evolutionary algorithms. *Computers & Industrial Engineering*, *66*(1), 63–76.

Marchesano, M. G., Guizzi, G., Vespoli, S., & Ferruzzi, G. (2023). Battery swapping station service in a smart microgrid: A multi-method simulation performance analysis. *Energies*, *16*(18), 6576.

Mattera, G., Polden, J., Caggiano, A., Nele, L., Pan, Z., & Norrish, J. (2024). Semi-supervised learning for real-time anomaly detection in pulsed transfer wire arc additive manufacturing. *Journal of Manufacturing Processes*, *128*, 84–97.

Mattera, G., Polden, J., & Norrish, J. (2024). Monitoring the gas metal arc additive manufacturing process using unsupervised machine learning. *Welding in the World*, 1–15.

Mattera, G., Vozza, M., Polden, J., Nele, L., & Pan, Z. (2024). Frequency informed convolutional autoencoder for in situ anomaly detection in wire arc additive manufacturing. *Journal of Intelligent Manufacturing*, 1–16.

Monek, G. D., & Fischer, S. (2023). IIoT-supported manufacturing-material-flow tracking in a DES-based digital-twin environment. *Infrastructures*, *8*(4), 75.

Mozharovskyi, P. (2022). Anomaly detection using data depth: multivariate case. arXiv preprint arXiv:2210.02851.

Nele, L., Mattera, G., Yap, E. W., Vozza, M., & Vespoli, S. (2024). Towards the application of machine learning in digital twin technology: a multi-scale review. *Discover Applied Sciences*, *6*(10), 1–23.

Olszewski, D., Iwanowski, M., & Graniszewski, W. (2024). Dimensionality reduction for detection of anomalies in the iot traffic data. *Future Generation Computer Systems*, *151*, 137–151.

Ranshous, S., Shen, S., Koutra, D., Harenberg, S., Faloutsos, C., & Samatova, N. F. (2015). Anomaly detection in dynamic networks: a survey. *Wiley Interdisciplinary Reviews: Computational Statistics*, *7*(3), 223–247.

Rashid, L., Rubab, S., Alhaisoni, M., Alqahtani, A., Alsubai, S., Binbusayyis, A., et al. (2022). Analysis of dimensionality reduction techniques on internet of things data using machine learning. *Sustainable Energy Technologies and Assessments*, *52*, Article 102304.

ur Rehman, M. H., Chang, V., Batool, A., & Wah, T. Y. (2016). Big data reduction framework for value creation in sustainable enterprises. *International Journal of Information Management*, *36*(6), 917–928.

Salatiello, E., Guizzi, G., Marchesano, M. G., & Santillo, L. C. (2022). Assessment of performance in industry 4.0 enabled job-shop with a due-date based dispatching rule. *IFAC-PapersOnLine*, *55*(10), 2635–2640.

Saran, N., & Kesswani, N. (2023). A comparative study of supervised machine learning classifiers for intrusion detection in internet of things. *Procedia Computer Science*, *218*, 2049–2057.

Shahraki, A., Abbasi, M., & Haugen, Ø. (2020). Boosting algorithms for network intrusion detection: A comparative evaluation of real AdaBoost, gentle AdaBoost and modest AdaBoost. *Engineering Applications of Artificial Intelligence*, *94*, Article 103770.

Shakya, V., Choudhary, J., & Singh, D. P. (2024). Principal component analysis and deep learning-based traffic anomaly detection in WSN. In *2024 IEEE international students' conference on electrical, electronics and computer science* (pp. 1–6). IEEE.

Shi, S., & Xiong, H. (2024). Solving the multi-objective job shop scheduling problems with overtime consideration by an enhanced NSGA-II. *Computers & Industrial Engineering*, *190*, Article 110001.

Tao, F., Qi, Q., Liu, A., & Kusiak, A. (2018). Data-driven smart manufacturing. *Journal of Manufacturing Systems*, *48*, 157–169.

Wang, S., Balarezo, J. F., Kandeepan, S., Al-Hourani, A., Chavez, K. G., & Rubinstein, B. (2021). Machine learning in network anomaly detection: A survey. *IEEE Access*, *9*, 152379–152396.

Wang, S., Li, J., Jiao, Q., & Ma, F. (2024). Design patterns of deep reinforcement learning models for job shop scheduling problems. *Journal of Intelligent Manufacturing*, 1–19.

Wang, H., Peng, T., Nassehi, A., & Tang, R. (2023). A data-driven simulation-optimization framework for generating priority dispatching rules in dynamic job shop scheduling with uncertainties. *Journal of Manufacturing Systems*, *70*, 288–308.

Wang, Y., Yao, H., & Zhao, S. (2016). Auto-encoder based dimensionality reduction. *Neurocomputing*, *184*, 232–242.

Weise, J., Benkhardt, S., & Mostaghim, S. (2018). A survey on graph-based systems in manufacturing processes. In *2018 IEEE symposium series on computational intelligence* (pp. 112–119). IEEE.

Yan, S., Shao, H., Xiao, Y., Liu, B., & Wan, J. (2023). Hybrid robust convolutional autoencoder for unsupervised anomaly detection of machine tools under noises. *Robotics and Computer-Integrated Manufacturing*, *79*, Article 102441.

Yeganeh, A., Chukhrova, N., Johannssen, A., & Fotuhi, H. (2023). A network surveillance approach using machine learning based control charts. *Expert Systems with Applications*, *219*, Article 119660.

Yu, B., Zhang, Y., Xie, W., Zuo, W., Zhao, Y., & Wei, Y. (2023). A network traffic anomaly detection method based on Gaussian mixture model. *Electronics*, *12*(6), 1397.

Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., & Saeed, J. (2020). A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. *Journal of Applied Science and Technology Trends*, *1*(1), 56–70.

Zoppi, T., Ceccarelli, A., Salani, L., & Bondavalli, A. (2020). On the educated selection of unsupervised algorithms via attacks and anomaly classes. *Journal of Information Security and Applications*, *52*, Article 102474.