THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING

Modeling Manufacturing Observations

Neural Time-Series Modeling under Irregular and Incomplete Observations

SILVAN MARTI

Department of Mechanical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden, 2026

Modeling Manufacturing Observations
Neural Time-Series Modeling under Irregular and Incomplete Observations

Silvan Marti

© Silvan Marti, 2026
except where otherwise stated.
All rights reserved.

Department of Mechanical Engineering
Division of Production Systems
Chalmers University of Technology
SE-412 96 Göteborg,
Sweden
Phone: +46(0)31 772 1000

Printed by Chalmers Digitaltryck,
Gothenburg, Sweden 2026.

"In the first fire-engines, a boy was constantly employed to open and shut alternately the communication between the boiler and the cylinder [...]. One of those boys [...] observed that, by tying a string from the handle of the valve which opened this communication to another part of the machine, the valve would open and shut without his assistance, and leave him at liberty to divert himself with his playfellows. One of the greatest improvements that has been made upon this machine [...] was in this manner the discovery of a boy who wanted to save his own labour."

— Adam Smith, *The Wealth of Nations*, Book I, Chapter 1 (1776)

Modeling Manufacturing Observations
Neural Time-Series Modeling under Irregular and Incomplete Observations

SILVAN MARTI
Department of Mechanical Engineering
Chalmers University of Technology

Abstract

Neural time-series methods hold increasing promise for improving how manufacturing systems operate, enabling more accurate monitoring, earlier fault detection, and better-informed operational decisions. Human-in-the-loop architectures extend this further, pairing machine learning with human expertise to refine and guide how analytical pipelines perform in practice.

Building such pipelines, however, surfaces a range of challenges. Industrial systems record machine states, buffer levels, and event logs for operational supervision, not to produce clean training sequences. The data entering neural time-series pipelines are often irregular in time and incomplete across variables, while most neural methods assume regularly sampled and densely observed inputs.

The aim is to relate how observation conditions arise in manufacturing systems to where neural methods encode assumptions about time and absence, and thereby ground method selection in the observation process for human-in-the-loop (HITL) decision-support settings.

A simulation study of a discrete manufacturing system shows how ordinary production-structure mechanisms generate temporally uneven, asynchronous, and dependent traces under controlled assumptions. A structured survey examines how neural methods for irregular and incomplete time series differ in their treatment of time and absence within the modeling pipeline. The main contribution is methodological: comparison should begin from the observation process, treating irregular sampling and missingness as distinct properties with different implications for representation, evaluation, and method selection in HITL decision-support settings.

Keywords

manufacturing cyber-physical systems, neural time-series modeling, irregular sampling, observational missingness, observation processes

List of Publications

Appended publications

This thesis is based on the following publications:

[**Paper I**] **S. Marti**, P. V. Lopes, S. Chen, M. Rajashekarappa, E. R. Bana, A. Göppert, M. Despeisse, J. Stahre, B. Johansson,
Synthetic Simulated Environment for Discrete Manufacturing Systems: A Demonstrator Through a Computational Modeling Approach
2024 Winter Simulation Conference (WSC), 1716–1727.

[**Paper II**] **S. Marti**, L. Kötz, E. T. Bekar, B. Johansson,
Learning from Time Series with Irregular Sampling and Missingness: A Survey
Under review.

Other publications

The following publications were published during my PhD studies. However, they are not appended to this thesis, due to contents overlapping that of appended publications or contents not related to the thesis.

[**a**] S. Chen, P. V. Lopes, **S. Marti**, M. Rajashekarappa, S. Bandaru, C. Windmark, J. Bokrantz, A. Skoogh,
Enhancing Digital Twins With Deep Reinforcement Learning: A Use Case in Maintenance Prioritization
2024 Winter Simulation Conference (WSC), 1611–1622.

[**b**] S. Chen, S. Bandaru, **S. Marti**, E. Turanoglu Bekar, A. Skoogh,
Comparison of Unsupervised Image Anomaly Detection Models for Sheet Metal Glue Lines
Engineering Applications of Artificial Intelligence
Vol. 153 (2025), 110740.

[**c**] A. Chari, **S. Marti**, B. Johansson, M. Despeisse, J. Stahre, P. V. Lopes,
Modeling Risk Prioritization of a Manufacturing Supply Chain Using Discrete Event Simulation
2023 Winter Simulation Conference (WSC), 2064–2075.

[**d**] S. Ghanbari, **S. Marti**, M. Granath, B. Johansson,
CTAT: Continuous-Time Anomaly Transformers for Irregular Time Series
Under review.

Acknowledgements

I am grateful to my main supervisor, Björn Johansson, for the trust and freedom to develop this work in my own direction, and for his steady and amicable support throughout. My co-supervisor, Mélanie Despeisse, brought methodological rigour and clarity of thinking on research design at the moments it was needed most. I would like to thank my examiner, Johan Stahre, for his engagement with the work and his constructive perspective during the preparation process.

I would like to thank Ebru Turanoglu Bekar for her thorough feedback on Paper II. Lasse Kötz, a fellow PhD student and co-author, brought a valuable control-systems perspective to the work on continuous-time models and was a good discussion partner throughout.

I am grateful to the Production Systems division for the collegial atmosphere and the many conversations that have fed into this work, often in ways that are difficult to trace. This research was supported by Vinnova, Produktion2030, the European Commission, and Chalmers Production Area of Advance.

Finally, my thanks go to my family and close friends for their support.

Contents

List of Terms and Abbreviations

Research Framework

AI artificial intelligence

DRM Design Research Methodology

DS-I Descriptive Study I

DS-II Descriptive Study II

HITL human-in-the-loop

PS Prescriptive Study

RC Research Clarification

RQ1 research question 1

RQ2 research question 2

SSE Synthetic Simulated Environment

Manufacturing and Production

Blocking condition in which a machine that has finished processing cannot release its output because the downstream buffer or machine is full

CPS cyber-physical system

DAG directed acyclic graph

DES discrete-event simulation

FSM finite-state machine

Historian industrial service that stores time-stamped process observations for retrieval and analysis

HMI human-machine interface

ISA-95 enterprise-control system integration standard

MES manufacturing execution system

PLC programmable logic controller

PN Petri net

SCADA supervisory control and data acquisition

Starvation condition in which a machine cannot begin processing because no input material is available from the upstream buffer

Timed Petri net a Petri net variant in which transitions carry firing delays or duration distributions, enabling the model to represent when events occur as well as how they are connected

Observation and Data

Aligned computational representation a model-ready structure in which observations are indexed to a reference time axis, producing a value array and an observation mask that records which values are present

Asynchrony non-coincident updates across channels or sources; channels are individually observed but record values at different times without coordination

EHR electronic health record

Event-triggered sampling a data-collection policy that records measurements when state changes or events occur rather than at fixed time intervals; contrasts with periodic polling

Irregular sampling non-uniform observation timing where time gaps between samples are not constant

MAR missing at random

MCAR missing completely at random

Missingness value absence conditioned on a chosen time axis or aligned representation

MNAR missing not at random

Observation process the sensing, recording, timing, and storage decisions through which a system becomes an observed dataset; recorded data are a selective trace, not a direct copy of system state

Reference time axis the ordered set of distinct times formed by merging all timestamped events across channels, used to align observations into matrix form; need not be regular or equally spaced

Semantic heterogeneity coexistence of record types with different meanings, such as measurements, events, and traceability records

Neural Temporal Modeling

ARIMA autoregressive integrated moving average

CDE controlled differential equation

CNN convolutional neural network

CT continuous-time

DT discrete-time

ELBO evidence lower bound

GAN generative adversarial network

GRU gated recurrent unit

LLM large language model

LSTM long short-term memory

ODE ordinary differential equation

RNN recurrent neural network

SDE stochastic differential equation

SSM state-space model

TPP temporal point process

Benchmark Observation Regimes

CON/MUL continuous or multi-source telemetry observation regime; intended acquisition is periodic but irregularity enters through equipment outages, communication failures, or asynchronous multi-source acquisition

EPS episodic decision-driven observation regime; measurement timing is coupled to system state, so observation density and pattern may be informative under MAR or MNAR mechanisms

EVT event-driven observation regime; data arrive as discrete occurrences in continuous time and inter-event time carries intrinsic signal rather than reflecting a fixed sampling design

TRJ/SYN trajectory-based simulation observation regime; data are generated from known continuous-time dynamics and subsampled to produce irregular observation patterns for controlled evaluation

List of Figures

List of Tables

Part I

Summary

Chapter 1

Introduction

This chapter introduces the background, vision, problem framing, aim, research questions, and scope of the thesis, and outlines its structure.

1.1 Background

Manufacturing performance is shaped, in part, by the quality of operational decisions [1]. Operators decide when to let production proceed, intervene, or inspect, but rarely with direct visibility into the underlying physical process. Those choices are made through mediated records (machine states, alarms, event logs) and their usefulness depends on what the records make visible, when, and how reliably their temporal structure can be read [2]. A first-pass yield falling steadily while the machine reports normal state, throughput holds at target, and no individual measurement leaves its tolerance band is not yet a fault; it is a pattern visible only when production counts, inspection outcomes, and inline measurement traces are read together against their usual rhythm. Reading such a pattern means reading the record against time: what appeared when, what came before, and what went unrecorded.

The infrastructure that produces these records is already in place, because automation requires it. Sensors feed programmable logic controller (PLC) controllers at millisecond timescales, while supervisory control and data acquisition (SCADA) systems and historians aggregate that stream for human oversight through deadbanding and time-averaging [3]–[5]. The scale is far beyond the span of routine operator attention; sensor counts in large industrial plants can reach into the millions [6], [7]. Established analytics, such as condition monitoring, work well within their design assumptions [8], [9]: stable operating conditions and predetermined fault signatures. They are correspondingly weaker on patterns that were not anticipated in advance, and on signals that only become legible when records across the stack are read together.

What follows from this is a different posture toward the existing infrastructure. The sensors, controllers, and historians already produce a continuous trace of the production system; yet the use of that record for decision support beyond routine control remains limited, a missed opportunity.

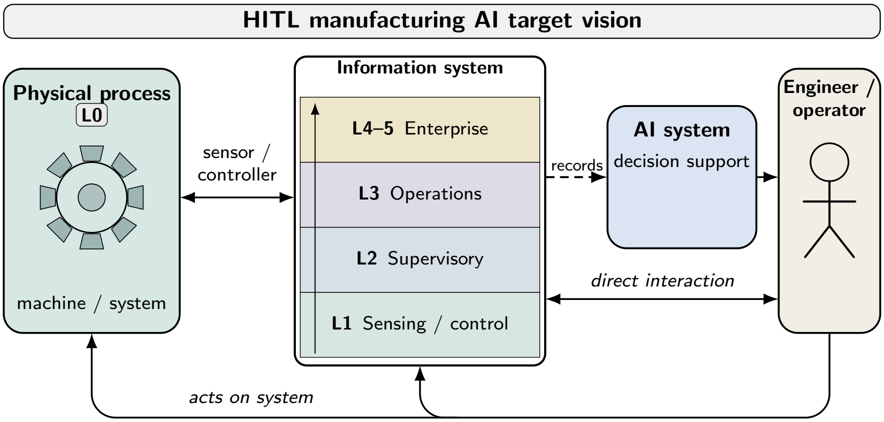

Figure 1.1: Human-in-the-loop (HITL) manufacturing AI: the target system. An AI layer identifies complex patterns in records, extending what the information infrastructure alone can surface to engineers and operators.

1.2 Vision

The vision this thesis works toward is a production environment where engineers and operators read complex patterns in the records their plants already generate, with neural analytics surfacing what unaided observation cannot. In the human-in-the-loop (HITL) manufacturing artificial intelligence (AI) system this thesis works toward, a skilled engineer or operator works with neural analytics rather than without them. In practice this means surfaced signals: an anomaly score that has crossed a threshold appearing in a monitoring dashboard, a flagged event in a maintenance queue, or a predicted bottleneck forming upstream before it becomes visible to unaided observation. The system surfaces the pattern; the engineer reads it against operational context, decides whether to intervene, and remains the locus of accountability for that decision. The judgment, the contextual reasoning, and the final action remain with the human. The analytical scope is the combined-record setting from Section 1.1: pipelines drawing from across the observation stack rather than a single stream. The operational insight HITL is meant to support depends on patterns visible across that combined view.

HITL is not a concession to current AI maturity; it is a design choice. Safety, accountability, and contextual judgment are structural parts of how production systems remain reliable [10]–[12]. The same posture has been studied in clinical medicine [12], [13], where high-stakes decisions and incomplete information make human oversight of model outputs a similar design requirement, and applied to industrial fault diagnosis [14]. The broader target system is sketched in Figure 1.1.

1.3 Problem Framing

The production systems that generate those records were not designed with AI in mind [15], [16]. They were built for control and supervisory visibility, and the system stack is heterogeneous across factories, production lines, and individual machines. What is recorded, at what resolution, and in what format reflects control requirements rather than modeling assumptions. Testing neural methods in running operations adds a further constraint: production time carries direct opportunity cost and safety requirements limit the controlled evaluation needed to characterize system behavior under fault conditions. A deeper constraint is the absence of labels: the conditions most consequential for operations (a developing bottleneck, or a process shift that precedes a fault) are precisely those not anticipated in advance. Neural methods address this in principle through unsupervised and self-supervised approaches [17]–[19], but only if the records can be brought to those methods in a form that preserves what they encode.

Manufacturing records are not neutral raw data. They are mediated traces produced by observation infrastructure operating on control logic. A machine may record only state changes while a related signal is sampled on a different schedule; both describe the same production system, but under incompatible timing logics. The resulting records are often irregular, asynchronous, and incomplete relative to the regular-grid assumptions of many neural time-series methods [17]–[20]. Many records arise from event-triggered rather than clock-driven observation. While standard data-quality frameworks treat sparsity as a defect, event-driven sparsity is better read as an infrastructure property of the observation regime [5], [21]. Log-format records require an explicit parsing step before temporal analysis can begin, and that choice is not neutral [22], [23]; in practitioner surveys, timestamp integrity consistently ranks as the primary quality concern in operational data [24].

The most common responses to irregular or incomplete records (gap-filling and temporal smoothing) do not allow a model to distinguish a true observation from a reconstructed one, discarding the information the absence itself carries [25]. When missing ratios are high, incorrect imputation introduces systematic bias into the processed record [26], [27]. Absence in manufacturing records frequently reflects the observation infrastructure itself (event-driven acquisition and logging design) rather than random data-quality failure [28], [29]: what is not recorded can be as informative about system state as what is. Operational records are also structurally imbalanced, and data quality assurance is not well incorporated into production systems despite the volume of data they generate [30]. A model trained on records whose temporal structure has been misrepresented generates signals whose apparent precision conceals an unreliable observational basis. The problem is not that the records are bad; it is that they are not in the form a model expects, and the way they are brought to that form can misrepresent what they encode.

1.4 Aim and Research Questions

Within the trajectory sketched in Section 1.2, this licentiate commits to one step on the path to that vision: characterizing the records-to-representation interface where manufacturing observation conditions meet neural modeling assumptions. Upstream lie the infrastructure and sensing layers that generate records: sensors, PLC controllers, SCADA systems, and historians; downstream lie the operator workflows and decision-support contexts in which a trained model surfaces signals to a human. This thesis addresses neither; it addresses what the records look like when they arrive at a model and what assumptions a model places on those records. The aim is stated at that interface.

This licentiate thesis aims to develop an understanding of how manufacturing-side observation conditions relate to neural time-series modeling assumptions. The enquiry has two parts: the first concerns the manufacturing side, the second the model side.

research question 1 (RQ1) *What time-series characteristics can arise from a simulated discrete manufacturing system?*

RQ1 addresses the manufacturing side of the interface [4], [31]: before comparing how neural methods handle manufacturing records, it is necessary to examine the observation conditions from which those records arise. Two representation characteristics are of particular relevance on the model side: irregular sampling, in which records arrive at uneven or state-driven intervals rather than on a regular grid, and observational missingness, in which values that would be expected are absent [17]–[20].

research question 2 (RQ2) *How do neural time-series methods differ in their treatment of irregular sampling and observational missingness?*

How a pipeline handles these characteristics, whether through interpolation, masking, or another form of treatment, carries assumptions about timing and availability that shape what a model can learn [18], [20]. Together, the two questions span the formation of time-series records on the manufacturing side and their handling on the model side [15], [20], addressing the records-to-representation interface as the prerequisite step for HITL manufacturing AI. The licentiate takes that step: it characterizes which observation conditions arise from discrete manufacturing and how representation choices encode assumptions about them, so later benchmarking and deployment can proceed with the interface understood rather than assumed.

1.5 Scope and Delimitations

The thesis covers the relation between observed manufacturing time-series conditions and neural modeling assumptions. Its empirical focus is on supervisory time-series signals from simulated discrete manufacturing, which provides a generative setting where observation conditions are inspectable under known structural assumptions; discrete manufacturing is chosen because its event-driven, buffer-mediated temporal structure is the primary source of the observation conditions studied here. On the model side, the scope is limited to neural time-series methods that address irregular sampling and/or observational missingness [18], [20], the class of methods increasingly applied to manufacturing records in practice, and where the interaction between representation choices and modeling assumptions is most consequential.

Several topics are outside the present scope. The thesis does not propose a new architecture, rank model families universally, or include high-dimensional modalities such as images or unstructured text. It also does not address downstream decision processes, control-loop integration, or fully autonomous operation. The target system is human-in-the-loop, and the present contribution sits upstream of the decision interface itself. Non-neural methods, including Markov approaches and process mining, remain established baselines for event-native logs and should feature in later benchmark studies, but are outside the present scope. Spatial and graph-relational methods, such as spatial-temporal graph neural networks over sensor or machine topologies, are likewise outside scope. They address a related but distinct observation condition, namely known relational structure between channels, which warrants separate characterization.

1.6 Thesis Structure

- Chapter 2 builds the frame of reference from manufacturing observation infrastructure through representation and neural temporal modeling to controlled evaluation.
- Chapter 3 operationalizes the frame of reference through the two nested studies and their procedures.
- Chapter 4 summarizes the two appended papers and their individual findings.
- Chapter 5 synthesizes the findings across the research questions and reflects on the aim, limitations, future directions, and contributions.
- Chapter 6 states what the evidence shows and where its scope ends.

Chapter 2

Frame of Reference

This chapter builds the conceptual route from manufacturing as it is currently operated and observed toward manufacturing with augmented decisions. Figure 2.1 guides that route, and the chapter moves through operational manufacturing, the infrastructure that observes it, how observations are represented, how representations are modeled in time, and how all of that is evaluated.

2.1 Manufacturing as an Operational System

2.1.1 Production as coordinated flow

Manufacturing is the coordinated production of goods across linked operations, often through long chains of heterogeneous steps [1], [4]. Figure 2.2 illustrates this for the automotive case, where body, powertrain, chassis, and interior streams proceed in parallel before converging at final assembly, where they must meet in the correct order and configuration. This coordination challenge is especially visible in built-to-order settings, where parallel streams must remain synchronized despite product variation.

Even for simpler products, production requires coordinated tooling, transport equipment, workers, control systems, and maintenance [1], [4], [34]. In practice, these elements are arranged in workstations, cells, lines, factories, and broader production networks. Manufacturing is not a single isolated transformation step, but a coordinated system of interdependent operations in which local changes (in machine condition, scheduling, transport, or operator intervention) can propagate outward across the production system.

2.1.2 Operational interpretation through mediated views

Operational personnel such as operators, engineers, supervisors, or maintenance staff see manufacturing through information provided by systems such as manufacturing execution system (MES), process-monitoring tools, and event logs [4], [15], [35]–[37]. What becomes available for interpretation is already selective and mediated: it reflects those aspects made visible through the

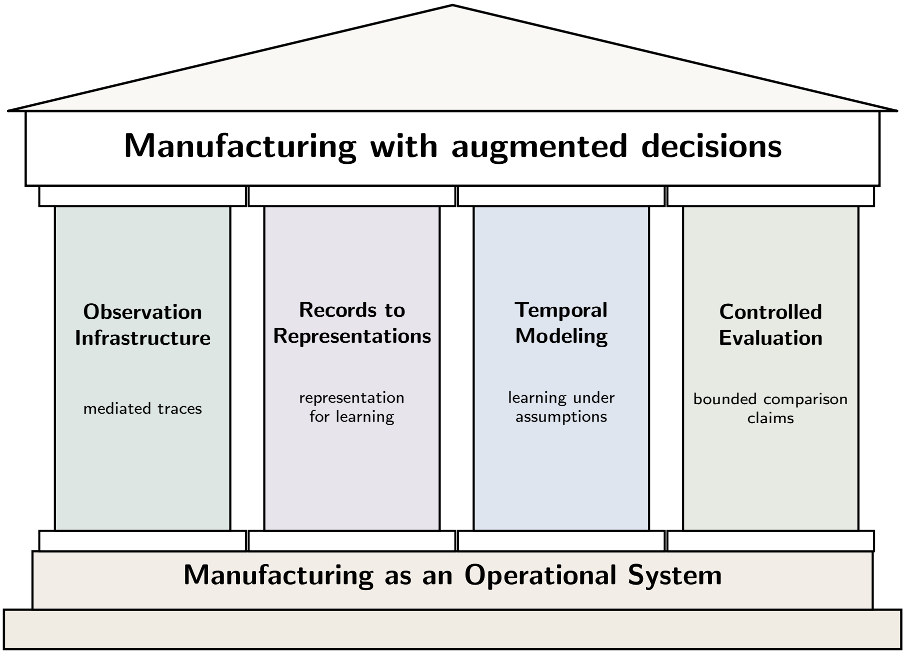

Figure 2.1: Conceptual guide to Chapter 2: base = current manufacturing practice, pillars = the four enabling concepts, roof = the target vision of augmented decision-making.

technical and organizational means by which manufacturing is supervised, not the underlying process in any direct sense.

A machine-state transition may be detected very quickly, while maintenance and scheduling decisions may follow on longer timescales, from minutes to hours. Each layer operates on a different clock, and manufacturing behavior is observable at multiple temporal scales [31]. At process level, relevant changes may occur very quickly, for example in sensing, control, actuation, and machine-state transitions. At supervisory and operational levels, information is often gathered, updated, and acted upon more slowly, shaped by machine cycles, operator responses, maintenance actions, scheduling decisions, and broader production coordination. Between process behavior and later analysis sits an observation infrastructure that determines what becomes visible, when it becomes available, and in what recorded form.

The next section focuses on that observation infrastructure and on how it turns process evolution into records available for analysis.

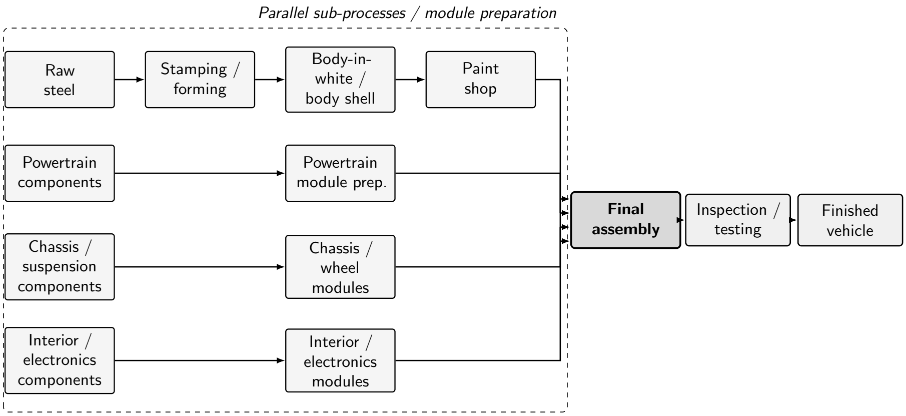

Figure 2.2: Concurrent and converging sub-processes in automotive manufacturing [32], [33].

2.2 Manufacturing Observation Infrastructure

2.2.1 Process, observation, and record

A vibration sensor mounted on machine components can continuously record vibration signals, which may change as operating conditions or faults develop. A controller log notes when a machine enters a failure state. These records, not the motor or the machine directly, are what later enters analysis. Characterizing the observation infrastructure through which selected aspects of manufacturing process behavior become available, and the relation those records bear to the underlying system, is a prerequisite for modeling.

The current signal from a motor can provide information relevant to its mechanical condition, but it does not by itself fully specify that condition. A controller alarm records that certain logic conditions were met, not the full trajectory that led there. The gap between what happens and what is stored makes it useful to distinguish the *process*, the *observation*, and the *record* [38]. The process is the underlying manufacturing cyber-physical system (CPS) (a system coupling physical operation with sensing, embedded control, and supervisory software) with its physical and logical dynamics. Observation is the infrastructural chain through which selected aspects of that process become detectable. The record is the stored temporal trace produced through that chain.

Modern manufacturing systems combine physical production resources with mechanical and electrical equipment, embedded electronics, and control logic. Material flows and operational coordination link these elements. In contemporary settings, many are usefully understood as manufacturing CPS, because physical operation is coupled to sensing, embedded control, communication, supervisory software, and higher-level information systems [10], [38], [39]. The framing implies that the interaction between the physical process and the digital infrastructure, not the physical process alone, generates the temporal traces used later for analysis. A given controller alarm records that certain logic conditions were met, not the full trajectory that led there; a historian trace records the values the infrastructure retained, not the full state evolution the system underwent. Manufacturing systems are only partially supervised, and many states of interest become visible only through proxies.

2.2.2 Layered infrastructure and recorded time

2.2.2.1 Layered architectures and their record types

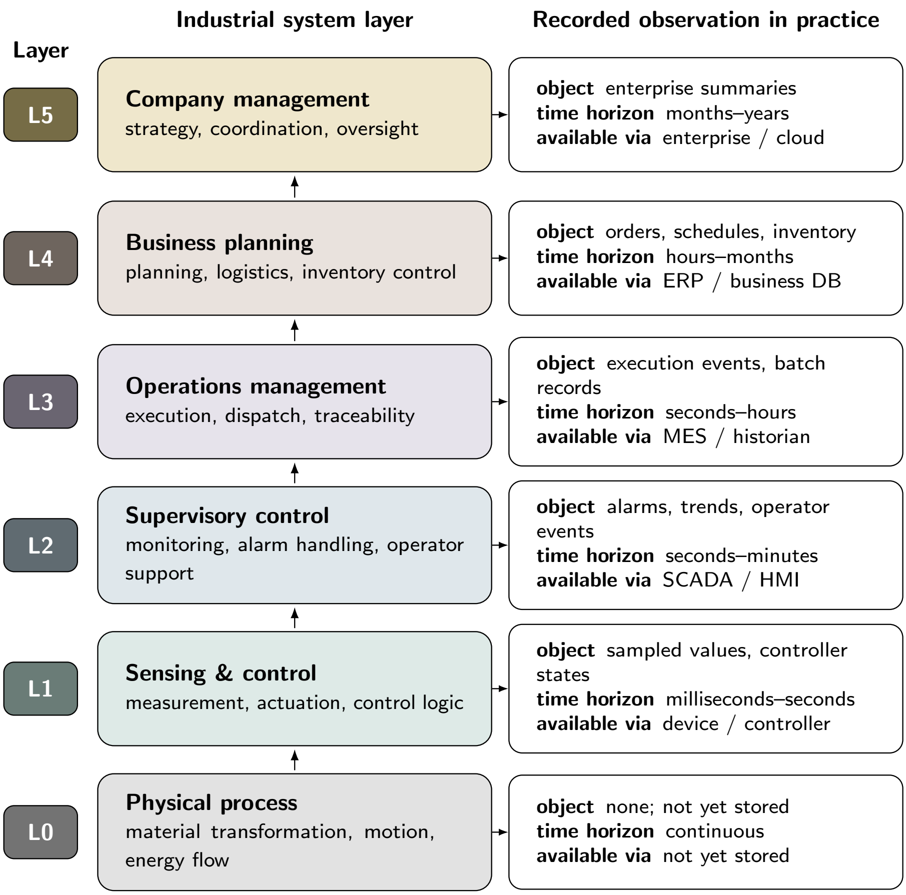

Figure 2.3: Layered path from system to record, adapted from [4]. The figure reads from the process level at the bottom to enterprise systems at the top.

One common way to describe such infrastructure is through layered automation and information architectures such as Purdue and enterprise-control system integration standard (ISA-95)/IEC 62264. As summarized in Figure 2.3, the infrastructure spans from the physical process to enterprise systems, with each layer serving a distinct operational purpose and producing a distinct class of record [4], [38], [40]. Rates across the stack span orders of magnitude. PMU standards mandate at least 30 frames per second in power infrastructure [41], while wind-turbine supervisory control and data acquisition (SCADA) systems typically average one sample per 10 minutes across 20–30 sensors per turbine [42]. These are cited as bracketing references rather than manufacturing instances: both rely on the same Purdue/ISA-95-style layered acquisition stack, and the cadence span they reveal is representative of the dynamic range manufacturing acquisition shares. Cadences across the stack span at least four orders of magnitude in common industrial practice.

ISA-95 serves here as an organizing framework, not as an exclusive boundary of data collection: deployment varies substantially by industry, plant scale, and vendor history, and a uniform implementation is not feasible across sites because interpretation of the standard strongly affects the outcome [3], [43]. In practice, records may also come from local controller logs, retrofitted sensing systems, laboratory instrumentation, maintenance systems, or manual reporting pipelines. The broader point is that manufacturing data usually becomes available through pre-existing operational platforms built for control, supervision, and planning, whose original purposes predate any downstream artificial intelligence (AI) use [26], [39], [44]. The four ISA-95 categories of enterprise-control information exchange (production definition, capability, schedule, and performance) are structured operational records, not raw sensor streams, and any pipeline that ingests them needs to handle their structured semantics accordingly [43].

The deployment of IIoT and 5G network infrastructure accelerates data accumulation but reinforces rather than eliminates temporal irregularity, since sensors continue to log on change rather than on a fixed schedule, and managing heterogeneous multi-source data at scale is itself an open research challenge in Industry 4.0 contexts [45]–[47].

2.2.2.2 Recorded time and timestamp semantics

Historians do not simply record every value at every instant. Industrial historians typically operate in exception-reporting mode, writing a new value only when the signal changes beyond a configurable threshold. In the Rockwell FactoryTalk historian, three parameters govern this behavior: *ExcMax* defines the maximum time allowed between successive writes regardless of value change; *ExcMin* defines the minimum interval between writes to suppress noise; and *ExcDev* defines the minimum change in value required to trigger a write at all [48]. Event-driven controllers obey the same logic. Event-driven controllers can require substantially fewer samples for the same control task: in one reported case, an event-driven implementation used roughly a quarter of the samples a 1 kHz time-driven controller would have produced over the same window [49]. Sparse records are therefore the intended output of the observation infrastructure, not a data-quality failure. The same compression mechanism that keeps historian storage and network traffic manageable during data collection also operates during live deployment: a model that learns from exception-reported traces learns from gaps that obey the same policy as the gaps it will encounter in production [50].

The observation path shapes the record further through time adjustments, interface conventions, and other system-specific transformations such as buffering, aggregation, synchronization, retention, and storage policies [4], [34], [51]. Time is not merely an ordering coordinate; it is part of record meaning. The OPC Unified Architecture (OPC UA) historical access specification distinguishes source timestamps, recorded when the originating device observed the value, from server timestamps, recorded when the historian received it: the two may differ whenever clock drift is present or when outage-recovery inserts reconstructed values after the fact [52], [53]. Beyond this source-versus-server distinction, timestamps may be coarse relative to the events they register, and human-entered records can inadvertently carry incorrect timestamps when a recorder's proximity to a prior event creates confusion about which moment is being logged [24].

Clock skew between networked devices adds a further layer: even small drift rates distort the temporal proximity between records from different sources, blurring the signal patterns that models are expected to learn [7], [54]. The same synchronization challenge extends to IoT deployments more broadly, where traditional imputation techniques are not designed for the restricted connectivity and processing constraints of sensor systems [26]. A controller-time occurrence, a historian write time, and a recovered value inserted after an outage each carry a different interpretive status [53]. Observability is partly an infrastructural achievement: what becomes visible depends on how process behavior, sensing, timing, storage, and integration are coupled. For temporal industrial records, data quality can be understood along three named axes: accuracy (measurement precision and validity), availability (completeness, consistency, and absence of loss), and timeliness (delivery within acceptable latency relative to the event) [55]. All three are constrained by the observation infrastructure described above and can fail independently: a record may be timely yet inaccurate, or accurate yet too delayed to inform the decision it was meant to support.

The result of this chain is the manufacturing record that later enters analysis. It may contain sampled measurements, controller states, alarms, logged transfers, execution events, and related operational traces. Alarm and event logs are not structurally equivalent to numeric sensor streams: they range from structured comma-separated values through semi-structured key-value pairs to unstructured human-readable messages with heterogeneous event types [56], and this structural difference implies different representation and processing strategies from those applied to continuous measurements. Such records are not raw process reality but infrastructurally produced traces whose timing, availability, and meaning have already been shaped by the observation path. In the emerging digital twin paradigm, where virtual replicas of physical systems are updated continuously from sensor streams, this challenge becomes a system-level requirement: the twin is only as current as the data that feeds it, and that data arrives asynchronously and sparsely [47].

Table 2.1: Core notation used in §2.3 and throughout the remainder of the thesis.

Symbol	Meaning
$\mathcal{D} = \{(t_i, y_i, c_i)\}_{i=1}^{N}$	Observed event record: the full set of timestamped triples from all channels
t_i	Timestamp of event i
y_i	Recorded value or state of event i
$c_i \in \{1, \ldots, C\}$	Channel identifier of event i
N	Total number of recorded events in $\mathcal{D}$
C	Number of channels
$\mathbf{t} = (t_{(1)}, \ldots, t_{(K)})$	Reference time axis: ordered union of distinct observation times across channels
K	Number of distinct instants on $\mathbf{t}$
$X \in \mathbb{R}^{K \times C}$	Value matrix of the aligned representation
$M \in \{0, 1\}^{K \times C}$	Observation mask; $M_{k,c}$=1 if channel c was observed at $t_{(k)}$, 0 otherwise
$\Delta t_k = t_{(k)} - t_{(k-1)}$	Inter-event gap on the reference time axis
f_θ	Learned model mapping with parameters θ

2.3 From Records to Representations

This section examines how a manufacturing record becomes a model-ready object and why the choices made in that step matter for what a temporal model can learn. Table 2.1 collects the notation used throughout this section and the chapters that follow.

2.3.1 Representation as problem formulation

Representation choices are not preprocessing decisions tucked in before the model; they fix what the model is allowed to treat as observation, absence, and continuity. Consider pre-aligning machine-state and buffer-level traces on a common time axis, for example using time-series alignment or forward-filling to handle gaps. The model now sees smooth, gap-free inputs: the gaps themselves may have been informative. More precisely, no matter how cleverly a gap is filled through interpolation, the fill cannot certify whether an anomalous event occurred during that interval [50]. The gap is resolved representationally but remains a blind spot epistemically, and a model trained on the same infrastructure-induced gaps it will later encounter in deployment is learning from absences that follow the same observation policy, not from errors.

Representation choices determine which temporal relations a model can learn from, and do more than the label "preprocessing" suggests. A recorded manufacturing trace is not automatically a model-ready time series. Choices about alignment, interpolation, masking, and encoding determine what counts as temporal structure for the model. Temporal discretization, for example, requires choosing both a bin width and an aggregation rule, and both choices determine which within-bin dynamics the model can ever observe [20].

The point is not simply that data must be formatted. It is that the model never receives the manufacturing process directly, and it does not receive the record in any untouched sense either. It receives a represented temporal object whose structure already traces back to decisions about time axes, channel integration, missingness handling, and task construction. Those decisions affect not only convenience, but also which temporal relations remain available for learning and what later comparison claims can reasonably mean.

Manufacturing records are typically stored as discrete snapshots, state changes, alarms, transactions, or other logged events rather than as a continuous record of process evolution. Different channels operate at different rates, and industrial acquisition commonly mixes periodic polling with event-triggered logging [11], [57], [58]. Since manufacturing systems evolve through concurrent and condition-dependent processes, relevant changes occur when the necessary conditions are reached. These include state transitions and material transfers, and the disruption states surrounding them: failures, repairs, blocking, and starvation [31], [38]. Irregular timing and cross-channel asynchrony arise in the process and observation chain itself, not only later as preprocessing problems.

Manufacturing records are also heterogeneous in meaning, in data type, and in timing. A sampled sensor value, a machine-state transition, a completed transfer, and a genealogy relation can all appear in the same corpus, yet they refer to different aspects of manufacturing activity [16], [31], [38]. That heterogeneity extends to data type: binary alarm states and categorical machine-state transitions coexist with continuous sensor readings in the same pipeline, and type-naive processing treats them as interchangeable when they are not [59], [60]. These are best treated as distinct observation types with different interpretive roles. The record already contains interpretive structure before it enters representation or learning [15].

2.3.2 Observation conditions in aligned records

Figure 2.4 illustrates the situation. A machine-state log records transitions at t=180 s, t=310 s, and t=515 s; a buffer-level log records values at t=180 s and t=310 s but has no event at t=515 s. These are three distinct situations: the channels update asynchronously (neither observed on the other's schedule), the inter-event gaps are non-uniform, and the buffer channel is absent at t=515 s on the aligned axis even though that instant belongs to the shared time axis. Collapsing them into one "missing data" problem loses the distinction between them. *Irregular sampling*, *asynchrony*, and *observational missingness* are therefore the three observation conditions central to this thesis; they arise from different sources and require different handling strategies [4], [61].

To reason about that step more precisely, it is useful to begin from a minimally committed event-style description of the recorded trace, given in Equation 2.1.

$$\mathcal{D} = \{(t_i, y_i, c_i)\}_{i=1}^{N}, \tag{2.1}$$

where t_i is a timestamp, y_i is a recorded value or state, and $c_i \in \{1, \ldots, C\}$ identifies the channel or record type that produced it. This form clarifies that one channel being observed at a certain time does not imply simultaneous

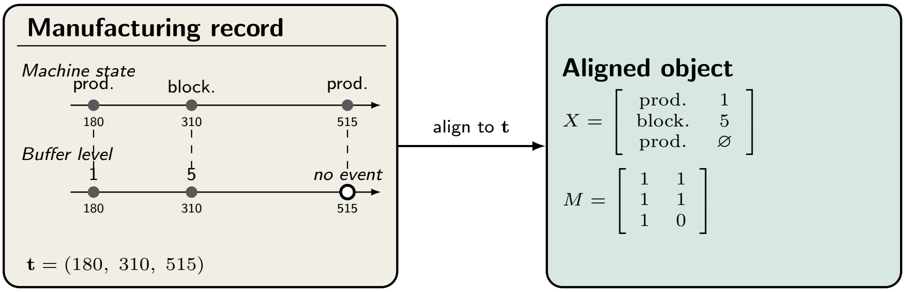

Figure 2.4: Two asynchronous channels aligned to their union of observation instants $\mathbf{t}$ (right). The buffer has no event at t=515, so $M_{3,2}$=0: mask M keeps value absence distinct from time irregularity. Here $\varnothing$ marks no recorded value, distinct from a recorded zero.

observation of all other channels. The triple structure makes no assumption about how many channels observe simultaneously or whether events cluster in time [4], [11]. A reference axis can then be formed from the distinct recorded times, as shown in Equation 2.2.

$$\mathbf{t} = (t_{(1)}, \dots, t_{(K)}), \qquad \{t_{(k)}\}_{k=1}^{K} = \mathrm{sort}\big(\mathrm{unique}(\{t_i\}_{i=1}^{N})\big), \tag{2.2}$$

The axis $\mathbf{t}$ orders the observation timestamps of records across channels chronologically. It makes no assumption that all channels were observed at each instant; it establishes a common timeline against which channels can be placed.

The record can then be projected onto that axis. For each instant $t_{(k)}$, a value matrix X records what was observed and a binary mask M records which channels contributed, as in Equation 2.3.

$$(X, M) = \mathrm{Align}(\mathcal{D}; \mathbf{t}), \qquad X \in \mathbb{R}^{K\times C}, \quad M \in \{0,1\}^{K\times C}, \tag{2.3}$$

where $M_{k,c} = 1$ indicates that channel c was observed at time $t_{(k)}$, while $M_{k,c} = 0$ indicates that no value is available there on the chosen axis [62].

This notation is useful because it helps distinguish three observation conditions that are often collapsed too early. Irregular sampling concerns the geometry of the time axis itself; the gaps $\Delta t_k = t_{(k)} - t_{(k-1)}$ between recorded instants are uneven rather than fixed. Asynchrony concerns non-coincident updates across channels or sources. Observational missingness, by contrast, concerns value absence conditional on a chosen axis or representation [18], [61]. These conditions are related, but they are not interchangeable. A record can therefore be irregular even if every observation event is fully specified, several channels can be individually well observed yet remain asynchronous with one another, and a representation can be regular in time while still containing substantial value-wise absence.

Where value-wise absence does occur, the standard taxonomy in the missing-data literature distinguishes three mechanisms by what the probability of absence depends on [63], [64]. Under *missing completely at random* (MCAR), absence is independent of both the observed values and the absent value. Under *missing at random* (MAR), absence may depend on observed values but not on the absent value given what is observed. Under *missing not at random* (MNAR), absence depends on the absent value or on quantities not among the recorded variables. In manufacturing records produced by event-triggered acquisition, the absence of an entry $M_{k,c} = 0$ at a non-event instant is by construction tied to whether the generating event occurred (the very thing the channel would have recorded), and so cannot satisfy the independence required by MCAR or the conditional independence required by MAR. This structural dependency will recur in §3.2.1.2 as the basis of Study 1's redirection finding.

2.3.3 Interpretive consequences of representation choices

Absence in a record can carry information about how the system behaves. A silent channel in a machine log may indicate a sensor or acquisition problem, or that the machine never entered a state that would trigger logging. In a designed event-triggered system, silence may carry information about system behavior. A missing sensor sample differs in interpretive status from an unlogged event [15], [31]. Absence is not always best interpreted as failure or corruption. A silent channel may reflect source logic rather than malfunction. An omitted event may mean that the system never entered a certain state. A sparse trace may follow a designed event-triggered policy rather than broken sensing. This does not mean that every absence is meaningful, nor that imputation is always inappropriate. It means that incompleteness can carry information about process behavior, source policy, or infrastructure behavior, and that the analyst has to decide whether that information is preserved, modeled, or suppressed [15], [28].

Once records have been aligned, synchronized, or otherwise adjusted, timing distortions and data-quality consequences often become more explicit. Alignment decides what counts as simultaneous. Windowing decides where one analytical unit begins and ends. Forward filling assumes persistence. Interpolation introduces between-observation behavior that was not directly observed. Event abstraction changes what qualifies as an event in the first place. Synchronization across SCADA, machine, historian, and enterprise records decides which timestamps, identifiers, and semantics are treated as commensurate. These operations do not merely tidy data; they shape which temporal relations remain available to the model and which have already been smoothed, collapsed, or introduced upstream [65]–[67]. The practical magnitude of this effect is visible in log-based anomaly detection: swapping the upstream log parser while keeping model and task fixed reduces the F1 score (harmonic mean of precision and recall) from 0.755 to 0.609 [22], [23]. That gap is larger than most architectural gains reported in that literature: a concrete illustration that representation choices matter more than the label “preprocessing” implies.

In manufacturing, records from different acquisition channels rarely arrive as a unified sequence. A controller event log, a historian trace, and an order record may cover the same production window yet differ in sampling rate, timestamp semantics, and event granularity. Combining them forces alignment decisions before modeling begins. In one setting, it may be appropriate to treat the combined record as one joint temporal object. In another, it may be more defensible to preserve the partial independence of sources and expose their couplings only through the chosen representation. What counts as the same sequence across machines, runs, or conditions is not given for free.

Representation is an integral part of problem formulation, not merely a peripheral preprocessing step [15], [20]. It determines what the model is allowed to treat as observation, absence, simultaneity, and continuity, and thus helps define what the learning problem is. Different temporal relations are also observed with different degrees of directness: temporal order may be directly recorded, while continuity between observations is usually not; state changes may be explicit, while cross-channel alignment may have to be reconstructed. When records from several sources are synchronized, the resulting representation may appear cleaner than the observational situation ever was. In that sense, preprocessing can reshape the phenomenon under study rather than merely tidy it. The multimodal machine-learning literature frames this more directly: for heterogeneous industrial data, the category boundary (what counts as a channel, an event, or a missing value) is itself an analytical decision, and no commonly applied dataset or standardized construction procedure exists for comparing results across settings [68]. Data-centric AI makes the same point from a different angle, treating data construction as part of the AI method rather than its antecedent [69]. For this thesis, that is the main reason the transition from record to representation matters: it is one of the primary places where assumptions enter before temporal modeling and evaluation [15].

2.4 Neural Temporal Modeling

2.4.1 Temporal modeling as a learning problem

2.4.1.1 Objectives and assumption placement

Temporal modeling involves learning from ordered sequences of observations to infer outputs or future states. Depending on the objective, the model produces a forecast, a detection score, a reconstruction, an imputed value, a latent embedding, or a class label for an observed episode [70]–[72]. Each objective relies on a different aspect of temporal structure: forecasting on autocorrelation and trend, detection on internalized expected behavior, reconstruction and imputation on cross-channel couplings, and representation learning on what the encoding retains. Objective definition and method choice are interdependent. The useful orienting distinction is where modeling assumptions about time and observation are placed. Traditional methods (spectral analysis, filtering, autoregressive integrated moving average (ARIMA), and state-space models) place strong assumptions in the representation or in an explicit dynamical model, and are most effective when those assumptions fit the data [70]–[72]. Neural sequence models relax some of those structural assumptions and can represent more complex dependencies, but require that the represented temporal object already carries enough signal for the learning task. Both families are parametric, but traditional methods embed structural assumptions in a small number of interpretable parameters, whereas neural methods distribute them across high-dimensional weight spaces learned from data. Whether architectural complexity translates into performance gains depends on what the represented data require. In standard forecasting benchmarks covering traffic, energy, and weather data, simple linear models have outperformed much more complex Transformer-based architectures by 20–50%. The reason is not that the more complex models are poorly designed; it is that the temporal structure of those datasets favors linear extrapolation over learned attention [73]. The illustrative point transfers because the lesson is about assumption fit rather than domain content: whenever the represented temporal object does not call for the inductive bias an architecture supplies, headline complexity does not translate into performance. The useful comparison between these families is therefore not architectural novelty but assumption placement: where the commitment to regularity, continuity, or observation structure is made, and whether that commitment fits the record.

2.4.1.2 Manufacturing-specific constraints

Manufacturing modeling objectives sit inside that same assumption-fit picture, with their own constraints on what the data require. In manufacturing, temporal modeling objectives arise in familiar and recognizable forms. A model may be asked to anticipate future process development, detect departures from expected behavior, reconstruct partially observed system state, estimate unobserved values, learn a representation that makes runs or operating regimes comparable, or assign labels or scores to episodes. Applying these objectives directly to manufacturing records is not straightforward, however. Data volume and label availability do not usually scale together. Industrial systems generate substantial temporal records through routine supervision, control, and traceability. Yet analytically useful labels are often incomplete, delayed, ambiguous, or costly to establish. Fault onset may be uncertain, interventions may weakly document what happened, and maintenance histories do not always align with the time-series episode later used for learning [44], [54], [74]. This gives particular importance to unsupervised or semi-supervised objectives such as anomaly detection, reconstruction, imputation, and representation learning, which can exploit large volumes of weakly annotated or unannotated records while still supporting monitoring and interpretation [15], [16]. These objectives are not interchangeable, however: imputation-based anomaly detection aims to enlarge reconstruction errors at anomalous points, whereas imputation for missingness recovery aims to minimize reconstruction error uniformly. A method optimized for one objective may systematically underperform on the other even when both are described as imputation [75].

A forecast can help an operator detect when a process may be heading toward failure or an anomalous regime, enabling timely intervention. A detection score surfacing an early deviation can trigger an inspection. These outputs are not operational decisions in themselves, but they can support one. A reconstructed state or learned representation can help compare episodes or interpret process behavior when direct observation is partial. Downstream manufacturing use is

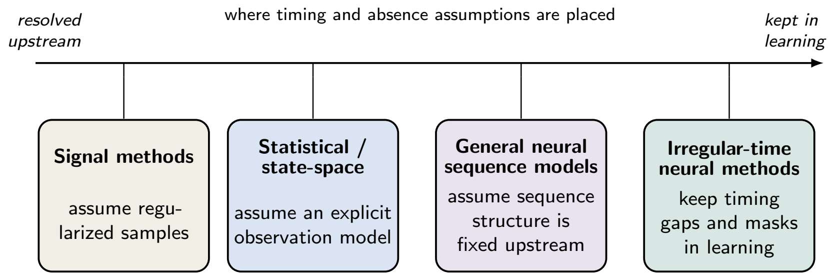

Figure 2.5: Four families of neural temporal-modeling methods, ordered by where each places its assumptions about time and observation.

assembled from discrete analytical components rather than delivered by one complete model.

The downstream usefulness of a model depends not only on architectural flexibility but on whether the represented temporal object preserves the distinctions the task depends on. If timing is collapsed too early, absences are given the wrong meaning, or asynchronous channels are forced into artificial simultaneity, later outputs may appear precise while resting on a weakened observational basis. For the present thesis, that point sets the scope clearly: the contribution remains upstream, studying a precondition for later monitoring, detection, and interpretation-oriented use rather than a complete decision-support pipeline.

At the most basic level, a neural unit computes

$$a = \phi(w^\top x + b), \tag{2.4}$$

where x is the input, w and b are learned parameters, and ϕ is a nonlinear activation. In this thesis, the relevant point is not the internal mechanics of neural networks as such, but that the model never receives the manufacturing process directly. It receives an input x whose order, spacing, alignment, masking, and observation structure already trace back to upstream choices [4], [15], [61]. Temporal models are therefore treated less as a catalogue of architectures and more as different ways of learning from a represented temporal object shaped by those upstream choices.

2.4.2 Families of temporal-modeling commitments

Neural sequence models differ less in their headline architecture than in *where* they ask irregularity to be handled: at the input, inside the recurrence, or in continuous latent dynamics. Figure 2.5 arranges four families along that axis. At the left, signal-processing methods resolve timing upstream in the representation; at the right, irregular-time neural methods keep it live inside the learning problem. The arrangement is descriptive, not evaluative: each family rests on different prior commitments: how timing is handled, how observations are aligned, whether continuity is assumed, how dependence is modeled, and what counts as an observation [18], [76].

Classical signal-processing methods (Signal methods, leftmost card) often place their key assumptions in the signal model or representation by exploiting known properties or structure of the signal. Fourier and wavelet transforms, filtering, and related tools are powerful when the represented record already behaves like a reasonably regular measured signal [77], [78]. They typically presuppose approximate stationarity over the analysis window. In manufacturing, that fit is strongest for continuous diagnostics such as vibration, current, temperature, or flow, where the data object is already close to a sampled signal. In such cases, upstream regularization has already determined which timing variation will be treated as meaningful structure and which will be treated as noise. Conclusions drawn from these methods depend not only on the method itself, but also on whether the record was first reshaped into something signal-like [57], [58], [79].

Statistical time-series models and state-space model (SSM) (second card) place more of the commitment in an explicit temporal and observation model. ARIMA-type models describe autocorrelation in sampled sequences, while state-space formulations separate hidden system evolution from noisy observation [70]–[72]. State-space formulations typically also place a Markov assumption on the hidden state. These models are closer than pure spectral tools to the problem of imperfect observation. However, layered logging, channel asynchrony, and semantically mixed records still force the analyst to decide what should count as hidden state, what should count as observation noise, and how timing enters the observation model. Their usefulness depends on whether those simplifying commitments still match the represented record [19], [72].

General neural sequence models (third card) such as recurrent neural network (RNN), sequence convolutional neural network (CNN), and Transformer-style models can learn complex temporal dependencies without specifying an explicit dynamical system, with much of the inductive bias encoded in the chosen input representation and architecture [72], [80], [81]. Yet the input they receive has already fixed choices about ordering, spacing, alignment, masking, and encoding [25], [62]. Their flexibility should therefore not be mistaken for an absence of assumptions. These models also presuppose that the represented data form a sequence, that is, an ordered and discretely indexed series. They inherit the regularity and contiguity this representation imposes, even when the underlying record is event-driven. They learn over a constructed temporal object, and their behavior remains conditioned by how that object was formed.

Some irregular-time neural methods (rightmost card) keep more of the observation structure inside the learning problem itself, rather than first converting irregular observations into regularly spaced reference points [18], [19], [82]. Some encode elapsed time and masks directly in the input. Others adapt updates to time since the last observation. Continuous-time or event-native variants place more of the main commitment into hidden dynamics between observations. These variants typically presuppose well-behaved, for example locally Lipschitz, evolution so that ODE solvers and continuous latent flows are well-defined. That assumption fits sampled physical signals but is poorly aligned with categorical or step-changing discrete channels. Related work also treats imputation as part of learning rather than as a fixed preprocessing step

[19], [83]. Even here, upstream decisions still matter, because one must still decide what counts as an observation event, what counts as a gap, and whether absence is modeled as defect, signal, or both. These methods preserve more of the observational structure inside the model, but they do not eliminate the need for representational judgment.

Large language models have been proposed as a general-purpose temporal modeling approach. Recent evaluations indicate they perform at or below specialized alternatives on numerical time series: in one systematic ablation, removing the language-model component improved or matched results across the majority of tested few-shot settings [84], [85]. The gap stems from tokenization and architecture choices that favor text over numerical precision. Their clearest current manufacturing role is as an orchestration and explanation layer, not as the core model operating on raw sensor records: decomposing diagnostic tasks, retrieving domain knowledge, and generating operator-readable explanations of model outputs [14], [86]. Purpose-built time-series foundation models trained on numerical temporal data are a stronger candidate for that core modeling role [87], [88].

For this reason, the present section treats temporal modeling as learning over represented traces rather than as an architecture survey. Once those observation and representation commitments are treated as part of the learning problem, model comparison can no longer be interpreted as architecture alone. The next section turns to evaluation as the place where those earlier commitments become claim boundaries.

2.5 Controlled Evaluation

2.5.1 Controlled evaluation and its rationale

Controlled evaluation means assessing systems under conditions deliberately specified by the evaluation design, rather than under whatever conditions the available data happen to provide. The analyst defines what varies between compared runs, what is held constant, and what property the comparison is meant to reveal. The value of that specification is that it reduces the number of explanations available for any observed difference: a performance gap is more likely to reflect method capability and less likely to reflect data provenance, target construction, or protocol choices [19], [89], [90].

In the development of machine learning systems for autonomous driving, simulation environments are widely used to test perception and planning systems: a synthetic environment provides ground-truth labels at scale, enables systematic variation of rare events, and allows faster iteration cycles than real-world collection. The parallel to manufacturing is structural rather than domain-specific: in both settings, fault or edge conditions are rare in operation, safety-critical to evaluate, and expensive to provoke in the live system, so simulation serves the same epistemic role of bounding what can be claimed under controlled conditions before the system leaves the controlled setting. Methodologically, this is the same logic as controlled experimentation in any engineering discipline: specify the test conditions, vary what you want to test, and hold everything else constant so that the result means something [91], [92].

One anomaly detector may have been evaluated on a regular sampled time-series representation, while another was evaluated on event-aligned observations with explicit handling of missingness and time gaps. Both may be described as anomaly detectors for machine data, yet they are not being evaluated under the same informational conditions. When manufacturing time-series methods are compared without such control, the result depends on more than model family alone: it also depends on how the temporal record was represented, how targets were defined, and under what protocol the comparison was carried out. Evaluation is not a neutral final step; it is part of how the problem itself is constructed. What is ultimately evaluated is not manufacturing reality in an untouched sense, but a represented task under specified observational and protocol conditions [15], [93].

In manufacturing, apparently similar learning problems may be built from quite different temporal objects. Reported performance can reflect not only method capability, but also how the benchmark made temporal structure available in the first place.

Industrial environments can generate substantial amounts of data, but labels are often incomplete, maintenance histories are uneven, infrastructures differ across sites, and the same signal can take on different meaning under different operating regimes [34], [38], [44]. These conditions make rigorous comparison difficult for further reasons beyond method capability alone. Benchmark outcomes can be shaped as much by target construction, split logic, and metric choice as by the model being tested. In SWaT (Secure Water Treatment), one of the most-cited industrial CPS benchmarks, 51 recorded attributes divide into 25 sensors and 26 actuators [94], [95]. Actuator states and sensor readings carry different observation and imputation logics, so a pipeline that does not distinguish them risks treating discrete control states under assumptions appropriate only to continuous measurements. In SECOM (semiconductor manufacturing), 1,567 examples include only 104 failures (6.6%) [96], with more than 40% of features carrying missing values [97], a configuration that the broader literature treats as the canonical example of records not yet ready for direct ML application.

2.5.2 Controlled artifacts and claim boundaries

In controlled evaluation settings, observation choices, label semantics, and model/protocol parameters are stated, and the state-transition structure under study is also made visible. In synthetic, simulated, or experimental settings, these elements can be specified and inspected directly [2], [98], [99]. Controlled evaluation does not replace real industrial data or establish deployment readiness. Its role is narrower: it helps clarify what a method is learning from, what kinds of variation are being held fixed or varied systematically, and what kind of claim the resulting score can reasonably support [4], [62], [83]. A further constraint applies at the detection level: reconstruction-error methods cannot detect anomalies whose deviation from expected behavior falls within the normal reconstruction envelope. When an anomaly is small enough that its reconstruction error is indistinguishable from normal-data error, the method will not surface it regardless of architectural sophistication [100]. This constraint connects directly to the exception-reporting mechanism discussed above: historian deadband compression is designed to suppress small-amplitude value changes, which are precisely the changes that small or gradual anomalies produce.

A benchmark for industrial machine-state forecasting need not replicate every plant detail, but it should preserve the temporal structure and missingness patterns relevant to the forecasting task, and ideally retain key inter-variable dependencies. Benchmark validity should therefore be judged against intended use [18], [72], [101]. The intended use in this thesis is methodological rather than deploymental: to support clearer comparison of representations and temporal models under manufacturing-relevant observation conditions. From that perspective, a useful benchmark does not need to reproduce every detail of a plant. It needs to preserve the system and observation properties that matter for the comparison being made, including system coupling and asynchronous or multirate observation structure. It also needs to state label semantics and allow operating variability to be changed systematically [98], [99], [102]. Labels are imposed, not discovered. In XJTU-SY (a bearing degradation dataset), the failure time is defined as the moment vibration amplitude exceeds 20 g, an operational safety threshold rather than a measurement of physical failure onset [103]. In MetroPT, one of three labeled catastrophic failures produced no observable sensor signal, because the fault type (oil leak) was invisible to available instrumentation [104]. Benchmark scarcity compounds the challenge: in a formal review of six candidate industrial datasets for failure prediction, only two met stated qualification criteria [96]. A benchmark that appears realistic while suppressing these mechanisms may be less informative than a simpler one that keeps them visible.

Simulation is therefore especially useful for methodological comparison and not for literal plant replication [2], [101]. In manufacturing, discrete-event simulation (DES) and Petri net (PN) are natural tools because many relevant phenomena are event-driven, concurrent, buffer-mediated, and resource-coupled [2], [5]. They make arrivals, completions, failures, repairs, blocking, and starvation visible in the model structure, while also making synchronization relations traceable. That makes them suitable not only for generating traces, but for preserving the kinds of mechanisms through which manufacturing observation conditions arise under controlled assumptions. Paper I (Appendix 1) follows this approach directly, building a parametrizable Synthetic Simulated Environment (SSE) for discrete manufacturing in which the observation conditions studied in this chapter (buffering, concurrency, event-driven state changes, and multirate observation) arise from known structural assumptions, and in which failure events and buffer levels are available as labeled ground truth. The simulation does not replicate a specific plant; it preserves the mechanisms that matter for the comparison.

Figure 2.6 summarizes this logic. The figure is not meant to suggest that controlled evaluation is sufficient for deployment. Its point is that relative claims

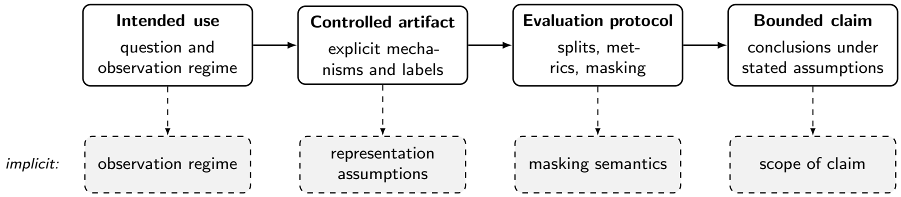

Figure 2.6: The controlled-evaluation chain (top) and the implicit assumptions each stage carries (bottom).

first become defensible when intended use, controlled artifact, and evaluation protocol have been linked. At that stage, the comparison is no longer between architectures in the abstract, but between methods evaluated under stated assumptions about observation, representation, and task construction.

Protocol design therefore remains part of the methodological problem rather than an administrative detail. Train–test splits, masking schemes, subsampling rules, discretization choices, label windows, and metric selection all affect what a reported score means [67], [82], [90], [102]. In C-MAPSS (turbofan degradation), the collection design produces one measurement snapshot per flight, a windowing decision made upstream that determines what temporal resolution the model can ever access [105]. A model tested on a densely regularized representation is not being evaluated under the same informational conditions as one tested on event-aligned masked observations. A detector scored on broad fault windows is not solving exactly the same problem as one judged on earlier or more precise targets. Controlled testing can therefore support bounded and credible claims, but not unrestricted claims of universal superiority [47], [106].

Chapter 3

Methodology

This chapter operationalizes the frame of reference from Chapter 2 through two complementary studies that move from manufacturing-side observation conditions to model-side assumptions. The thesis applies Design Research Methodology (DRM) [107] as its design logic. Detailed experiment variants, parameter settings, and paper-format protocol records remain in the appended papers.

3.1 Methodological framing

3.1.1 Philosophical stance and reasoning mode

The philosophical stance is pragmatist in Creswell's sense: method selection is governed by fitness for purpose and what works for the research problem, not by loyalty to a single epistemological tradition [108]. The reasoning mode throughout is abductive: neither theory-first nor data-first, but a recursive process of inference to the best explanation in which the theoretical framing of the problem was progressively sharpened through empirical confrontation [109], [110]. This is consistent with the DRM stage structure: Research Clarification is, by design, the phase that produces the problem formulation as its output, not a phase that tests a pre-specified hypothesis [107].

3.1.2 Choice of DRM and alternatives considered

The thesis uses one design logic and two nested studies with study-specific procedures. Study 1 constructs a bounded case generating manufacturing-side traces under controlled assumptions. A bounded generative case specifies its structural and temporal assumptions explicitly, making the conditions under which observations arise inspectable. Study 2 maps how neural methods represent and handle irregular timing and observational missingness once such records enter the learning pipeline.

The broader problem of applying neural methods to manufacturing involves many challenges this thesis does not address (Section 1.5); the focus here on the records-to-representation interface follows from the RC finding described in Section 3.2.1.2. The research began from a working hypothesis that simulation would serve as a direct testbed for neural method comparison; running the simulation produced the finding that event-drivenness is a structural constraint on method applicability, which confirmed irregular timing and observational missingness as the right interface to study and redirected the licentiate scope accordingly.

The stance is engineering-oriented and pragmatic in its method choice [107], [111]. Knowledge claims are framed to support later design work rather than to establish a universal theory. Study 1 supports mechanism-level claims about how irregular and incomplete observations arise in a controlled simulated production setting. Study 2 supports analytical claims about how existing neural methods place assumptions about time and missingness. Together, the studies provide the basis for relating manufacturing-side observation conditions to model-side assumption placement.

DRM was selected for two reasons: its epistemological commitments fit the engineering-pragmatic stance, and its staged structure matches the shape of the problem [107]. DRM was developed specifically for engineering design research whose goal is to produce prescriptively useful knowledge; it is not a generic empirical or theoretical framework. This is appropriate here because design problems are inherently ill-defined [112]: the problem formulation is an output of Research Clarification, not a precondition for it, which distinguishes DRM from design science research [113], where a pre-specified artifact and problem are assumed at the outset. Its staged progression formalises the move from understanding a problem to improving practice, making the transition from descriptive to design knowledge explicit and traceable rather than implicit. The research questions reflect this staging directly: research question 1 (RQ1) is a Research Clarification question about what observation conditions arise in discrete manufacturing, and research question 2 (RQ2) is a Descriptive Study I question about how each neural method commits to time and absence handling; together they occupy exactly the first two DRM stages. The alternative of proceeding directly to a controlled method comparison was foreclosed by an early finding: without first characterizing the observation conditions, any comparison design would rest on unexamined assumptions about what the data are. A methodology without an explicit problem-clarification stage struggles to accommodate this; DRM's Research Clarification (RC) stage is the vehicle through which Study 1's finding redirected the licentiate's scope. Three alternatives were considered and set aside:

- **Pure controlled experimentation** assumes a well-formed evaluation design already exists; the RC finding showed that this assumption was premature.
- **Design science research** centers on artifact construction, which falls outside the scope of a licentiate positioned at RC and Descriptive Study I (DS-I) within a longer programme.
- **A standalone systematic literature review** could not have produced the mechanism-level manufacturing-side finding of Study 1.

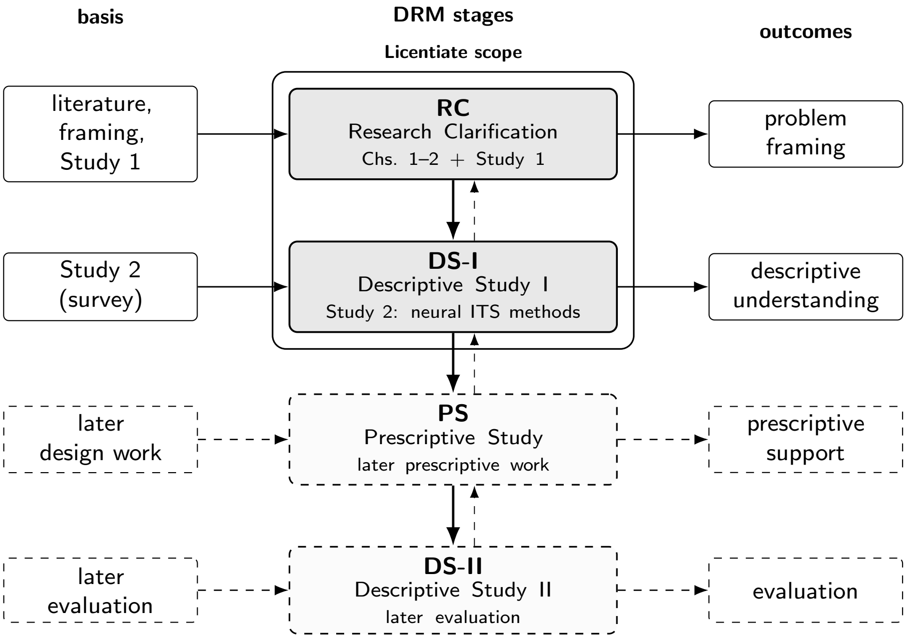

Figure 3.1: DRM operationalization for the licentiate. Solid boxes (RC, DS-I) fall within the present scope; dashed boxes (PS, DS-II) are future work.

No alternative framework offers an equivalent staging concept; DRM was selected on that basis.

3.1.3 DRM stages and licentiate scope

DRM distinguishes Research Clarification (RC), Descriptive Study I (DS-I), Prescriptive Study (PS), and Descriptive Study II (DS-II) [107], [111]. Figure 3.1 shows the four stages, their inputs and outputs, and the licentiate scope (RC and DS-I).

DRM's staged progression from problem clarification through descriptive and prescriptive studies provides the organizing logic for the two nested studies and makes the transition from descriptive to design knowledge traceable.

The engineering-oriented and pragmatic stance bounds knowledge claims to what is instrumentally useful for later design decisions. It licenses mechanism-level claims, such as that event-drivenness is a structural constraint or that a given method family places its core assumption at a particular pipeline stage. It also accepts that a simulated system is not a real plant and that a structured survey is not exhaustive, provided the evidence is sufficient for the intended design work [107], [111].

3.2 Study design and procedures

The two studies operationalize the RC and DS-I stages of DRM outlined above. Table 3.1 maps the aim and research questions to the two studies, the procedures each uses, and the evidence each produces.

Table 3.1: Research-question and study mapping.

	Study 1	**Study 2**
Research question	RQ1	RQ2
DRM stage	RC	DS-I
Paper	P1	P2
Procedure	Simulation	Structured survey
Evidence	Controlled traces	Method map

Study 1 addresses RQ1 by producing controlled traces under stated structural assumptions, making observation conditions inspectable and later model-comparison claims interpretable. It clarifies what kinds of irregular, asynchronous, and dependent traces can arise in the simulated setting. Study 2 addresses RQ2 by synthesizing how neural methods handle irregular sampling and missingness, and where in the pipeline they do so. Together, the two studies provide the basis for relating data-generation conditions to assumption placement [107], [110]. Study 1's RC finding, that event-drivenness is a structural constraint entering at the representation stage, determines which niche of neural methods Study 2 must cover; without that finding, Study 2's scope (irregular and incomplete time series) would be an unsupported assertion.

3.2.1 Study procedures

3.2.1.1 Study 1: procedure and model specification

Study 1 uses a simulated discrete manufacturing system to generate inspectable temporal traces under stated structural assumptions. The study represents the system as a directed acyclic graph (DAG) of production assets and material-flow dependencies [106], [114]. At each node, a Timed Petri net models event-driven flow and resource constraints, while a finite-state machine (FSM) governs operational states. A Petri net represents a discrete-event system as a bipartite graph of *places* (holding tokens that represent resources or buffer contents) and *transitions* (events that fire when their input places hold sufficient tokens, redistributing tokens to output places). A *Timed* Petri net augments transitions with firing durations, making the timing of state changes part of the model rather than an external annotation. A finite-state machine (FSM) represents per-node operational state (idle, producing, blocked, or failed) as one of a finite set of labelled states, together with the event-triggered rules governing transitions between them. The combination places production-network flow and resource logic in the Timed Petri net layer and per-node operational status in the FSM layer. Timed Petri nets are used rather than agent-based or other discrete-event simulation (DES) formalisms because their established formal semantics for concurrency, blocking, and resource coupling in manufacturing flow make the generated observation conditions directly traceable to production-system structure [5], [115]. The simulation follows DES logic: events such as production completion, blocking, failure, repair, and maintenance change node states, propagate through the flow structure, and generate timestamped records [2], [5].

The environment is instantiated in the appended paper as a controlled demonstrator case. In this licentiate, the demonstrator serves as a controlled generative setting in which uneven timing, asynchronous channel updates, and cross-node dependence become visible in traces such as machine states and buffer levels. Detailed parameterization, event scheduling logic, and run-specific settings are reported in the appended paper.

The primary data products are timestamped multivariate traces modeled under stated structural and temporal assumptions. These traces are used descriptively to inspect non-uniform inter-event spacing, blocking-related burst–gap patterns, and dependency propagation across nodes [62]. Traceability is supported by documented model structure, defined event types, reported baseline configuration, and retained timestamped outputs.

3.2.1.2 Study 1: RC finding: event-drivenness as a structural constraint

Study 1 began from a working hypothesis that simulation would provide a testbed for direct neural method comparison: the intended design was to construct the simulation, generate traces, and then run method comparisons on it directly. Running the simulation, however, produced a finding that redirected the scope of the licentiate: the event-driven character of discrete manufacturing observation is a hard structural constraint on neural method applicability. In Dubois and Gadde's terms, this constitutes a direction and redirection: the matching of theoretical framing against empirical material revealed that the original comparison design rested on unexamined assumptions about the data, redirecting the research toward the constraint itself as the primary finding of the RC phase [110]. This constraint enters at the representation stage, before model selection or training, and is not resolved by preprocessing. It is a property of the data-generation process, not an incidental data-quality issue.

The mechanism is as follows. The same system components produce two kinds of observable channels: event-driven transitions (machine state transitions, failures, blockings, and repairs) and value-type measurements (buffer levels and throughput rates). Each channel type is captured through a distinct observation scheme. When their records are merged onto a shared reference time axis, the result is a heterogeneous stream that is neither a pure event sequence nor a regular multivariate time series. Three properties of this stream matter:

- the reference time axis is event-triggered and non-uniform;
- channels update asynchronously at their own event times;

- apparent value absence in the aligned representation is an observation-regime artifact, not sensor failure.

The pattern of apparent missingness is driven by the event process and is therefore structurally missing not at random (MNAR) (in the sense defined in §2.3.2) [20], [25]. Event-drivenness and observational missingness frame the same structural constraint two ways. Weerakody et al. note that missing data and irregular data are "often used interchangeably" in the time-series literature [25]. Shukla and Marlin show formally that discretising an irregularly sampled series onto a regular grid produces missing values wherever no event fell in a time bin [20]. The review in Study 2 therefore covers both model families (those that operate on irregular event streams and those that address observational missingness) because they address the same underlying structural problem from different representational starting points. Standard formalisms from the event-sequence tradition, including temporal point processes, Hawkes processes [116], and their neural extensions, treat the merged observation as a pure marked event stream: they handle event-driven timing correctly but have no mechanism for value-type channels that carry defined states between events. Standard time-series formalisms treat value absence as missing data under conventional assumptions and apply interpolation, imputation, or masking to a regularized grid: they can represent irregular gaps but lose the event-driven structure of the reference axis and misattribute the structurally MNAR missingness to data-quality issues rather than observation regime. Neither standard handling accommodates the inter-modality structure without modification; the problem sits between the two families.

The contribution of Study 1 within the thesis is discussed in Chapter 5.

3.2.1.3 Study 2: search and selection protocol

Figure 3.2 shows the five-stage survey procedure applied in Study 2: seed assembly, iterative snowballing, corpus screening, grouping by dominant method feature, and structured characterization.

Study 2 followed a structured survey procedure centered on iterative snowballing in the sense of Wohlin [117]. A seed set was assembled from targeted keyword searches in major scholarly indexes and from prior surveys on irregular and incomplete time series [20], [25], [26], [118]. These prior surveys were used as seeds rather than as substitutes for the review. Each is bounded in a way that does not align with the problem framing of this thesis: some have older coverage that predates a substantial body of recent neural methods, and others delimit their scope by architecture family rather than by the structural observation problem. Existing reviews do not center on the records-to-representation interface as their organizing concept, which is why an own review centered on that framing was required, and why the seeds were used as entry points into the citation network rather than as a finished corpus.

Snowballing was preferred over a conventional PRISMA-style keyword search because the field uses inconsistent terminology across communities (*irregular time series*, *missing data*, *event-driven observations*, *sparse asynchronous sequences*) and pre-specified search strings would have missed papers that address

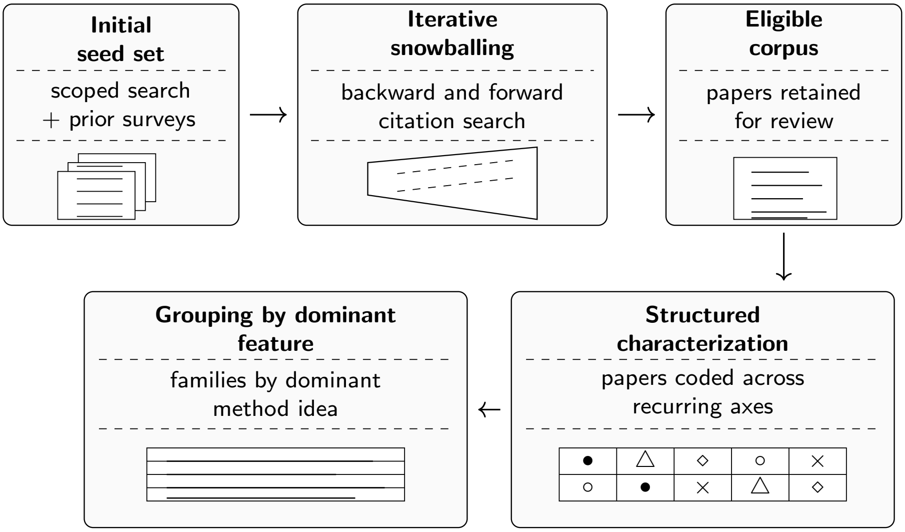

Figure 3.2: Study 2 survey procedure across five stages: seed assembly, iterative snowballing via forward and backward citation search, screening to an eligible set, grouping into method families, and structured characterization.

the same structural problem under different labels [117], [119]. In machine learning model papers, citing a prior work as a quantitative baseline implies shared datasets, evaluation metrics, and problem scope. This is a stronger semantic connection than generic academic citation. Following backward and forward citation chains from seed papers known to benchmark against each other recovers the papers that address the same problem formulation through that comparison network [117]. This produces a concept-centric review in Webster and Watson's sense: the shared benchmark problem is the concept, and papers connected by direct experimental comparison address that concept in practice [120]. The review continued until each additional pass yielded only papers covering model families already fully represented in the corpus, applying the criterion of theoretical saturation [121] to the citation graph; the stopping rule is empirical adequacy, not an arbitrary iteration count.

Candidate papers were screened first by title and abstract and then by full text when needed.

Inclusion required a neural method for sequential observations or event streams, explicit treatment of irregular timing and/or variable-wise missingness in the modeling pipeline, and empirical evaluation [83]. Exclusion covered non-neural approaches, preprocessing-only handling of irregularity, papers without empirical evaluation, and papers outside the review scope. Snowballing is a recursive procedure rather than a sequence of linear passes: each seed paper is the root of a search tree whose branches are formed by backward references (the works it cites) and forward citations (the works that cite it). Each newly included paper becomes a node from which the same backward-and-forward expansion is repeated, so the corpus grows along a tree-structured traversal of the citation graph rather than through a fixed number of database queries. A branch was terminated when continued expansion produced only papers covering method families already fully represented in the corpus; the corpus as a whole was considered saturated when all open branches met this termination criterion. Because the procedure is recursive and branch-local rather than batched, it does not yield a meaningful "number of passes" in the PRISMA sense; the appropriate replication artifact is the seed set together with the inclusion and exclusion criteria, which together determine the tree the procedure traverses. Seed searches combined terms for observation structure (irregular sampling, observational missingness, asynchrony) with terms for method family (neural time series, temporal point process, sequence model, continuous-time); the seed set itself is reported in the appended paper.

3.2.1.4 Study 2: evidence and safeguards

Methodological safeguards included explicit inclusion and exclusion criteria, documented seed-set sources, iterative snowballing records, and a fixed characterization template for cross-paper comparison [117], [120]. Coding consistency was maintained through sequential application of the fixed template with systematic cross-reference to previously coded entries; characterization was carried out by a single coder without inter-rater reliability checking, which is acknowledged as a limitation. These safeguards ensure that method characterization is faithful to reported designs and that synthesis groupings reflect actual method mechanisms rather than ad hoc categorization.

3.2.1.5 Study 2: characterization and synthesis

Included papers were coded using a common characterization template, and the appended survey paper includes representative excerpts illustrating that template. For each method, the coding captured six dimensions [18], [20], defined in Section 4.2.3.1. The guiding question was where assumptions about irregularity and missingness enter the learning pipeline.

The synthesis proceeded in two stages. First, we decompose the modeling pipeline into the layers where a method places its assumptions about irregular timing and observational missingness. The input encoding turns raw observations into the representation the model consumes. The discrete-time recurrent update propagates a state across gaps between observations. The continuous-time latent dynamics evolve the state on a continuous time axis irrespective of when observations arrive. The training objective encodes assumptions about timing or absence directly in the loss. These layers are not alternatives but successive opportunities for an assumption to enter the model; a single method typically commits at one or two of them and leaves the others implicit.

Second, for each layer the included methods were grouped into a small number of categorical approaches that share a mechanism for handling irregularity or missingness at that layer. At the input encoding layer, for example, methods divide into approaches that discretize onto a regular grid with masking, approaches that augment each observation with explicit time-gap features, and approaches that treat the input as a marked event stream. Analogous categorizations were produced at the discrete-time (DT) recurrent-update layer, the continuous-time (CT) latent-dynamics layer, and the training-objective layer. Each category was then analyzed for the kind of observation regime it implicitly assumes and the structural information it preserves or discards.

This two-level structure (layers as loci of assumption placement, categories as the recurring mechanistic choices within each layer) is what enables the comparison RQ2 requires. It locates each method not just *where* in the pipeline it acts, but *which type of commitment* it makes at that location relative to alternatives at the same layer.

Alongside the layered analysis of methods, the synthesis also recorded recurring observation regimes used in evaluation, including event-driven streams, decision-driven clinical sampling, telemetry with disruptions, and synthetic trajectories.

The primary data product is a structured map of neural method choices and their observation-process assumptions.

3.3 Research quality

Research quality across the chapter rests on traceability between five elements: the problem framed in Chapters 1 and 2, the DRM design logic, the nested study procedures, the evidence each study produces, and the claims made from that evidence [90], [107]. Table 3.1 and Figure 3.1 provide that trace, while Section 3.2 states what each study can support and how procedures are structured. Chapter 4 reports the study outputs descriptively, and Chapter 5 performs the joint interpretation.

Construct validity. For Study 1, the construct of interest is the observation conditions that arise in discrete manufacturing: irregular timing, channel asynchrony, and structurally dependent missingness. The controlled generative case captures these directly, because the constructs are properties of the observation process and the traces are the direct output of that process under stated structural assumptions. For Study 2, the construct of interest is assumption placement within the learning pipeline. The six coding dimensions (Section 3.2.1.5) correspond to distinct pipeline stages and rest on established taxonomies of sequential modeling [70], [122].

Internal validity. For Study 1, the primary internal validity threat is that observed trace characteristics may reflect parameterisation choices rather than structural properties of the model. This threat is partially mitigated by the controlled generative case design: the structural properties of interest, namely event-triggered timing, asynchronous channel updates, and MNAR missingness, follow from the formal semantics of the Timed Petri net and FSM combination, not from specific parameter values. Whether the constraint's severity varies with system scale, buffer capacity, or failure rates is a genuine open question; systematic parameter variation is left to later prescriptive and evaluative work.

External validity. Study 1's controlled demonstrator represents discrete job-shop-type manufacturing with buffered workstations, event-driven state transitions, and finite-capacity queues. The structural conclusion, that the co-existence of event-driven and value-type channels generates the inter-modality problem, generalises to any discrete manufacturing system where these two channel types operate on different observation logics, regardless of scale or topology. Continuous-process and hybrid manufacturing systems are outside scope, as noted in Chapter 1, because their observation regimes differ fundamentally.

Reliability. Characterization for Study 2 was carried out by a single coder without inter-rater reliability checking; this is a genuine limitation, and the method groupings and synthesis conclusions reflect one analyst's reading. The primary consistency mechanisms were the fixed six-dimension characterization template, applied in identical form to every included paper, and sequential cross-referencing of each new entry against previously coded papers to maintain stable interpretation of shared dimensions. The iterative Wohlin snowballing structure provided a further consistency check at the inclusion stage: the same inclusion and exclusion criteria were applied at three decision points in each level of tree expansion (reference-list screening, title and abstract screening, and full-text screening), and this sequence was repeated consistently across all levels of expansion until saturation. Borderline cases (papers that partially met inclusion criteria) were resolved at the full-text screening stage; the decisions are documented in the appended survey paper. These mechanisms constrain but do not eliminate the single-coder limitation: they ensure that the template was applied consistently and that inclusion decisions followed stated criteria, but they do not substitute for independent coding. Replication would require the same seed set, independent application of the inclusion and exclusion criteria, the same characterization template applied to the resulting corpus, and comparison of grouping outcomes across coders.

Limitations. Study 1 does not compare neural methods (out of scope at the RC stage); it characterizes the observation conditions that any comparison would need to address. The controlled demonstrator's external validity is limited to the class of discrete manufacturing systems described above. Study 2 coverage may not be exhaustive: the snowballing protocol begins from a seed set and is subject to the coverage of the indexed sources; non-English papers were not systematically included. The synthesis groupings by dominant modeling commitment are analyst-constructed categories and should be read as an organizing frame, not an ontology.

Chapter 4

Summary of Appended Papers

This chapter briefly summarizes what the two appended papers establish within the scope of this licentiate. Paper I reports the manufacturing-side generative case, and Paper II reports the model-side landscape. The chapter is descriptive; synthesis is reserved for Chapter 5. Table 4.1 provides an overview of the two papers, their relation to the research questions, and their role in the thesis.

Table 4.1: Overview of the appended papers and their thesis role.

Paper	Study object	Main finding	Why it matters for the thesis
Paper I	Discrete-manufacturing SSE	Production structure generates uneven timing, burst–gap behavior, and cross-node dependence in simulated traces	Provides the controlled manufacturing-side case used to answer RQ1
Paper II	Survey of neural irregular-time methods	Method families differ in where they place assumptions about time gaps, missingness, and latent evolution	Provides the model-side map used to answer RQ2 and connect observation conditions to representation choices

Together, the two papers cover both sides of this interface. Paper I addresses RQ1 by showing how production structure shapes the manufacturing record before any model sees it. Paper II addresses RQ2 by showing that comparing neural methods by where each places its observation-handling assumption is more informative for manufacturing contexts than comparing by architecture label alone.

4.1 Paper I: Simulated Environment

Title: *Synthetic Simulated Environment for Discrete Manufacturing Systems: A Demonstrator through a Computational Modeling Approach*

Paper I provides the controlled manufacturing-side case used to address RQ1.

4.1.1 Problem

Paper I addresses how relevant time-series characteristics can arise from production structure in a controlled discrete-manufacturing setting. The paper starts from the observation that evaluation of artificial intelligence (AI) methods in manufacturing may be more informative when the data preserve operational structure such as production flow, buffering, and event order [2], [123]. In discrete manufacturing, recorded traces are determined by production flow, buffering, concurrency, and event-driven state changes [2], [5]. Within the thesis, this corresponds directly to RQ1: *What time-series characteristics can arise from a simulated discrete manufacturing system?*

4.1.2 Approach

Paper I (Appendix 1) develops a parameterized SSE for discrete manufacturing in which material flow is modeled as a directed acyclic graph (DAG) with node-level Timed Petri net and finite-state machine (FSM) logic. Timed Petri nets can represent manufacturing flow, concurrency, and resource constraints with explicit state-transition and resource-dependency semantics [115]. The environment is implemented in PyTorch and generates labeled multivariate traces of machine states and buffer levels under controlled structural assumptions.

4.1.3 Results

The demonstrator produces two labeled multivariate outputs: machine operational states and buffer levels over time. In the reported configuration, these outputs are generated from a five-node production network with unit local production capacity, finite output buffers, and stochastic cycle, failure, and repair behavior. The configuration makes it possible to inspect how local production dynamics appear as coupled temporal records.

The traces suggest a coupling mechanism in which a block at one node can influence adjacent nodes by propagating interaction patterns and state changes across the network [5], [34].

4.1.3.1 Observed trace properties

The simulated traces exhibit four properties relevant to downstream modeling. They are temporally dependent and mutually dependent across variables, and may be discrete-valued and irregularly sampled depending on the data-generating process. Table 4.2 summarizes the observable time-series properties and the manufacturing mechanisms that produce them.

Table 4.2: Observed trace properties in the SSE demonstrator.

Property	**Generating mechanism**	**Output type**	**Sample record pattern**
Integer-valued buffer levels	Finite buffer capacity	Bounded integers	$0 \to 1 \to 5 \to 4$
Four-state machine traces	FSM state transitions	Four-state categorical	idle $\to$ producing $\to$ blocked
Non-uniform inter-event times	Event-driven scheduling	Irregular Δt	$\Delta t = 180$ s, 130 s, 205 s
Burst–gap state patterns	Downstream blocking	Variable-duration states	short producing bursts separated by blocked intervals
Cross-node correlations	DAG connectivity	Correlated multivariate traces	downstream congestion appears in upstream state and buffer updates

The main reported result is that a specified inter-event process can generate bursty, non-uniform event spacing and that cross-variable dependencies arise, demonstrating how these dynamics can be tied to production mechanisms in the modeled network [5], [36]. For the present licentiate, these are treated as manufacturing-side observation conditions made visible under controlled assumptions. The simulation produces labeled, parameterized traces of machine states and buffer levels with the observation properties listed in Table 4.2.

4.2 Paper II: Survey of Neural Methods

Title: *Learning from Time Series with Irregular Sampling and Missingness: A Survey*

Paper II provides the structured model-side survey used to address RQ2.

4.2.1 Problem

Paper II addresses RQ2. Many neural methods presuppose regular sampling and dense observation, whereas manufacturing cyber-physical system (CPS) often produce records with non-uniform observation times and variable-wise absence patterns. The survey therefore treats *irregular sampling* and *observational missingness* as distinct properties of the observation process, with different implications for representation, learning, and evaluation [20].

Prior methods for incomplete time series include end-to-end models trained directly on missing data and classical imputation-based approaches such as expectation-maximization or last-observation-carried-forward. Continuous-time latent-variable models offer an alternative by explicitly representing irregular temporal dynamics. Without a common vocabulary, comparing across them is comparing unlike things. The specific survey problem is fragmentation across families, terminology, datasets, and evaluation protocols [75], [78], [83], which makes comparison difficult. Prior surveys [20], [25], [26], [118] organize methods primarily by application domain or architecture label. Paper II organizes methods by where in the pipeline each handles time and absence, rather than by application domain or architecture label as in [20], [25], [26], [118]. This is RQ2.

4.2.2 Approach

Paper II uses a structured snowballing survey [117] restricted to neural methods that treat irregular timing and/or missingness within the modeling pipeline. The reviewed methods are compared using the six-dimension characterization framework introduced in Section 3.2.1.5, applied uniformly across the surveyed corpus.

4.2.3 Results

The survey produces four main results: an observation-process framing with a canonical representation, a characterization framework for neural learning pipelines, a mapping of method families and representation strategies, and an overview of recurring observation regimes and benchmark practices.

4.2.3.1 Characterization framework

The characterization framework identifies six recurring dimensions that record where methods place assumptions about time, variable absence, supervision, and between-observation dynamics [26], [62], [118]. The first is *domain*, the application context in which the method was developed. The second is *objective*, which fixes what is optimized and what counts as supervision. The third is *input encoding*, the representation strategy that determines what computational object the learner receives and which observation-process signals, including masks, time gaps, and event tuples, remain explicit after encoding. The fourth is *backbone*, the primary mechanism for aggregating and propagating dependencies among encoded observations. The fifth is *latent evolution*, which specifies whether and how temporal dynamics are modeled between observations. The sixth is *learning paradigm*, the statistical training formulation: discriminative, likelihood-based, variational, adversarial, or contrastive.

The six dimensions are approximately orthogonal but frequently coupled in practice: many methods pair interpolation with a recurrent backbone, or combine mask-aware encoding with a reconstruction objective. For the present work, this framework provides the structured basis for mapping each method's position in the model-side answer to RQ2.

4.2.3.2 Method families and representation strategies

On this basis, the surveyed literature is organized into five method families, grouped by their treatment of temporal structure, missing data, and latent dynamics [70], [83].

The first family comprises *input-side encoder or encoder-decoder models* that address missing data and irregularity during representation learning, before downstream sequence modeling. Strategies within this family include event-set formulations that retain original event tuples and timestamps; canonical alignment with explicit mask and time-gap augmentation; imputation prior to backbone processing; discretization that declares a new regular time axis; time reparameterization that synthesizes values at reference points; and continuous-path interpolation. What unifies the family is that the backbone receives a representation in which timing and availability information has already been absorbed, retained, or discarded.

The second family is *gap-aware sequence models*, which augment step-wise architectures to handle variable inter-observation intervals and irregular sampling. Two sub-patterns recur: transition-semantics adaptation, in which gap-dependent decay or time-modulated gating determines how latent state persists across variable-length intervals; and observation-interface adaptation, in which mask-conditioned or inline-imputed updates control how partial inputs enter the recurrent cell. Attention-based discrete-time (DT) variants handle irregularity through time-aware or interval-aware similarity in the attention kernel rather than through state persistence.

The third family is *continuous-time (CT) latent models*, which treat observations as samples from a latent path defined over real time. Sub-variants include:

- **Latent ordinary differential equation (ODE) formulations** infer initial conditions and integrate dynamics between observation times.
- **Recurrent–continuous hybrids** apply a continuous integrator between events and a discrete update at each event time.
- **Controlled differential equation (CDE)-based formulations** convert the interpolated observation path into a continuous driving signal for the latent dynamics.
- **Stochastic differential equation (SDE) formulations** add diffusion-driven stochastic uncertainty.
- **Structured state-space model (SSM) parameterizations** derive inter-observation transitions in closed form.

The *event-time modeling family* treats event timing as part of the predictive target, aiming to forecast when future events will occur rather than treating timestamps only as preprocessing input. These methods model the conditional intensity of a temporal point process (TPP), trained by a likelihood that jointly rewards high intensity at observed event times and penalizes excess intensity in intervals without arrivals. The family is naturally suited to logs, alarms, and other data where inter-arrival time carries signal that coarser temporal aggregation would suppress.

The fifth family comprises *imputation and generative models*, which are defined by learning to reconstruct or generate missing and latent values through generative objectives. Reconstruction targets may be deterministic, evidence lower bound (ELBO)-based variational, generative adversarial network (GAN)-based adversarial, or diffusion-based; in each case, missing values or their distributions become the supervision signal rather than a preprocessing problem.

Many methods in the surveyed literature combine innovations across families, for example interpolation-to-ODE pipelines that couple input encoding and CT dynamics in the same model, so the five-family organization is best read as a pragmatic grouping by dominant modeling emphasis. Table 4.3 summarizes the key distinctions across these families. At the thesis level, this organization shows that method choice is not architectural alone: it determines where assumptions about irregular timing and missingness are concentrated in the learning pipeline.

Table 4.3: Conceptual comparison of major method families in Paper II.

Method family	Time	Missingness	Assumption locus	Why it matters
Input-oriented encoders	Time exposed in inputs	Masks and gaps retained	Input representation	Observation patterns visible
Gap-aware sequence models	Gaps modify state updates	Masks and recency explicit	State transition	Fits asynchronous traces
Continuous-time latent models	State evolves between observations	Partial observations update trajectory	Latent dynamics	Between-sample evolution
Event-time models	Time modeled as events	Absence read as no event	Problem framing	Fits logs and alarms
Imputation and generative models	Time handled by backbone	Missing values become supervision	Learning objective	Imputation or uncertainty focus

The survey also records recurring representation strategies across these families, including event-tuple formulations, aligned-grid encodings with masks and gaps, and imputation- or interpolation-based variants [75]. These choices matter descriptively because they determine which parts of the observation process remain visible downstream [38], [82].

4.2.3.3 Observation regimes and datasets

Across the surveyed application literature, a pipeline-oriented view picks out four recurring observation regimes, each with its own modeling requirements [83]:

- **Event-driven processes (EVT).** Data arrive as discrete occurrences in continuous time.
- **Episodic decision-driven settings (EPS).** Clinical or operational decisions order the measurements.

- **Continuous or multi-source telemetry (CON/MUL).** Data are nominally periodic but subject to outages and asynchrony.
- **Trajectory-based simulation settings (TRJ/SYN).** Data are generated from known continuous-time dynamics and subsampled to produce irregular observation patterns.

These categories are analytical groupings rather than a partition; real datasets often combine properties of more than one, and each is introduced in detail below.

In *event-driven processes* (EVT), data arrive as discrete occurrences in continuous time and inter-event time carries intrinsic signal; irregularity is not a degraded version of a regular record. The primary modeling stress falls on the learning objective and time semantics: event-time objectives that jointly model occurrence time and mark type align naturally with this regime, whereas coarse temporal aggregation can suppress the fine-scale structure that carries information. Financial limit-order-book streams, represented by the LOBSTER (Limit Order Book System: The Efficient Reconstruction) dataset [124], are a canonical example; their relevance here is structural, since like manufacturing event logs they consist of timestamped marked events whose inter-arrival times themselves carry the signal of interest.

In *episodic decision-driven settings* (EPS), measurements are ordered by clinical or operational decisions, and the resulting observation pattern often reflects the underlying system or patient context. Absence may indicate improvement or stability rather than missing data, making the observation mask potentially informative under missing at random (MAR) or missing not at random (MNAR) mechanisms. What matters most here is input encoding: representations that retain the observation mask and timing metadata as first-class inputs are better positioned to preserve the signal carried by the observation pattern. PhysioNet Challenge 2012 [125] and MIMIC-III (Medical Information Mart for Intensive Care) [126] are the canonical EPS benchmarks in the survey. Their relevance to manufacturing is conditional: like operator-initiated inspections or condition-triggered measurements, clinical observations are decision-driven, so the timing of an observation can itself encode information about the underlying state. Where manufacturing measurements are clock-driven rather than decision-driven, the analogy breaks down and other regimes apply.

In *continuous or multi-source telemetry* (CON/MUL), data are often intended to be collected regularly, but acquisition can become irregular because sensors are sampled asynchronously or at different times. The dominant constraint here is interpolation quality and gap-aware backbone robustness. A recurring concern noted in the survey is that protocols which induce irregularity by randomly removing entries from an otherwise dense record may underrepresent the structured contiguous outages that appear in real deployments. PEMS-BAY (Performance Measurement System, Bay Area) [127] is the representative CON/MUL benchmark; this matches manufacturing SCADA and historian streams in being nominally periodic but subject to network outages, sensor dropouts, and asynchronous updates.

In *trajectory-based simulation settings* (TRJ/SYN), data are generated from known continuous-time dynamics and subsampled to produce irregular observation patterns, enabling controlled tests of interpolation, extrapolation, and density sensitivity. For this regime, the modeling load shifts to CT dynamics assumptions and solver behavior, since the ground truth is a latent trajectory and observation density is a sampling design parameter. A recognized limitation of this regime is external validity: observation times are typically independent of latent state in TRJ/SYN benchmarks, whereas in EPS regimes they are coupled to it. MuJoCo (Multi-Joint dynamics with Contact)-style physics simulations [128] are the canonical TRJ/SYN example.

Because the same nominal learning task can stress different pipeline components depending on which regime the benchmark instantiates, Paper II argues that regime details should be reported as part of benchmark specification rather than treated as a neutral background condition. The survey records representative datasets for these regimes, including PhysioNet Challenge 2012 [125], MIMIC-III [126], PEMS-BAY [127], LOBSTER [124], and MuJoCo-style synthetic trajectories [128].

4.2.3.4 Benchmark protocols and recurring issues

The survey records several recurring protocol patterns that affect comparison, including sampling regularity, benchmark dynamics, and whether irregularity is intrinsic or evaluation-induced [82].

The first is *informative observation intensity*: in decision-driven regimes, how often a variable is measured may correlate with the state of the underlying system, so models may benefit from using observation density or time-since-last-observation in addition to observed values. Methods that encode timing and masks explicitly can potentially exploit this signal; methods that homogenize observation density through discretization, masking, or upstream imputation may suppress it.

A second limitation is protocol-induced comparability: benchmark choice, inclusion criteria, and evaluation measures can shape what is assessed and how results are compared. Masking-as-ground-truth imputation, which creates evaluation targets by artificially removing observed entries, couples the result to the specific masking scheme rather than the natural observation mechanism. Timestamp discretization collapses within-bin timing and shifts the modeled object from the original event record to a new aligned grid. Induced irregularity, formed by subsampling from a dense regular record, tests robustness to a known corruption model rather than to intrinsic event-triggered or decision-driven sampling. Paper II records these as benchmark parameters that vary across studies and shape what is compared.

The third is *sensitivity and transfer*: methods may perform well under one sampling regime or domain yet may not generalize across rates, heterogeneity, or settings. Paper II also records sensitivity to encoding choices, limited robustness, and restricted transfer as recurring challenges. For the present work, these results provide the model-side description of where assumption choices enter once observations are represented for learning.

Chapter 5

Discussion

This chapter brings together the results of the two appended studies and develops a synthesis. It begins with answers to the two research questions, then a synthesis, the relation to the aim and vision, the limitations, future directions, and contributions.

5.1 Reflection on research questions, synthesis, and aims

The thesis converges on a structural point: the moment a manufacturing record is aligned, interpolated, or masked, the model's access to the original observation conditions changes, and that change has consequences for what any reported performance comparison can mean.

5.1.1 Answers to the research questions

This licentiate addresses one methodological interface from two directions. research question 1 (RQ1) examines what kinds of observation conditions can arise on the manufacturing side before records enter a learning pipeline, while research question 2 (RQ2) examines how neural methods handle two central conditions once those records are represented for learning. The discussion begins with the most specific level of result: what the thesis answers to these two questions are, and what was learned from each answer within the scope defined in Section 1.4.
RQ1 (Section 1.4): ***What time-series characteristics can arise from a simulated discrete manufacturing system?*** In Paper I's setting, production-system dynamics generate temporally dependent and non-stationary traces, with irregular temporal behavior arising through event-driven timing and sampling effects. More specifically, buffering, inter-machine coupling, and event-driven scheduling generate non-uniform inter-event intervals, blocking-related burst–gap dynamics, asynchronous updates across channels, and cross-node dependence in the resulting traces. Within the studied setting, the answer

to RQ1 is therefore that learning-relevant observation conditions can arise from production logic itself, rather than only from later data cleaning, storage defects, or ad hoc preprocessing [15].

RQ2 (Section 1.4): *How do neural time-series methods differ in their treatment of irregular sampling and observational missingness?* Paper II shows that neural methods differ primarily in where they place assumptions about time and absence within the learning pipeline. Some methods place the main commitment in the input representation, for example through encoding of time gaps or masks. Others place it in discrete-time state updates, continuous-time latent dynamics, or learning objectives such as imputation, reconstruction, generation, or event modeling. Within the same setting, the survey supports a parallel answer for RQ2: relevant differences between neural methods are better understood through assumption placement than through architectural labels alone [20], [82].

5.1.2 Synthesis

The two answers in Section 5.1.1 form one connected chain: production systems generate behavior; observation infrastructure turns selected parts of that behavior into records; representation choices then turn those records into model-processable temporal objects; and neural methods place assumptions about time, absence, and between-observation dynamics at different points within that pipeline. Meaningful comparison therefore requires the full chain, from generation conditions to representation and evaluation, to be kept visible [20], [80].

Absence is not always recording failure; two concrete cases make this distinction operational for method selection. A machine that never enters a fault state produces no alarm log entry: that silence is informative, because it reflects operational behavior rather than recording failure. A vibration sensor that drops out during high-amplitude events, by contrast, produces gaps that are data-quality problems, because the absence reflects infrastructure limitation rather than process state. After alignment onto a common time axis, both appear as absent values in the same column. The choice of whether to impute, mask, or model each as an explicit absence is difficult to make on principled grounds without knowing which kind of absence it is.

5.1.3 The records-to-representation interface as a structural site

A further structural observation follows from reading the two studies together against the broader catalogue of observation conditions developed in Chapter 2. Each method family characterized in Paper II is tractable under the observation conditions its assumption set admits, and each locates its assumptions at a different pipeline locus. Continuous-time latent models commit to continuity assumptions that categorical and event-coded channels do not satisfy; event-time models commit to point-process semantics that continuous sensor traces do not satisfy; input-side encoders commit to a chosen alignment policy that absorbs timing information before the backbone receives it. Each family is well-formed within its own admissible regime and less so outside it. The observation conditions catalogued in Sections 2.3.2 and 2.4 (including multiple notions of recorded time, semantic heterogeneity across record types, cross-node asynchrony without a shared reference axis, and the irregularity and missingness studied here) are not jointly covered by any single surveyed family. Records carrying several of these conditions simultaneously require any single-family pipeline to resolve the residual conditions upstream of the model. Recent multi-modal encoders address semantic heterogeneity at the input layer, but do so under representational choices that still locate timing and absence assumptions at a single pipeline locus; the joint-coverage question is therefore not resolved by representational scope alone.

This mismatch sharpens with the heterogeneity of the records the pipeline combines: where records share a uniform timestamp regime and a single record type, the conditions catalogued in Chapter 2 do not coexist within the data, and the assumption-set mismatch within any single method family is correspondingly less acute.

Chapter 2 catalogues further characteristics of real-world manufacturing records, for instance parser sensitivity across heterogeneous source formats [22], [23], that this thesis did not analyze in the same depth as irregular timing and observational missingness. Even the two characteristics that were analyzed already constrain what a model can assume about its inputs, so covering a broader portion of the assumption space within a single pipeline is expected to be challenging at this transition rather than straightforward.

5.1.4 Relation to aim and vision

Read against the aim stated in Section 1.4, the synthesis in Sections 5.1.1–5.1.3 is where the licentiate's central claim becomes visible, and within the stated scope, the aim was met. The two studies together characterize one named interface (between the observation conditions discrete manufacturing systems generate and the assumptions neural time-series methods place on the records they receive) and show that the families surveyed do not jointly cover the catalogued conditions at that interface.

Read against the human-in-the-loop (HITL) vision in Section 1.2, the contribution is upstream and conditional. The vision the licentiate works toward is a production environment in which engineers and operators read patterns across combined records with analytics that respect how those records were generated; reaching it requires monitoring methods, decision-support interfaces, deployment infrastructure, and operator workflows the present licentiate does not address [12]. What it does deliver is one precondition for that path: a structural reading of the records-to-representation interface that lets later monitoring, anomaly-detection, and interpretation work be set up with the interface understood rather than assumed [31]. Successor work in the directions of Section 5.4 takes the next steps from here.

5.2 Limitations

The main limitations follow from the synthesis's treatment of the problem and from the incompleteness of the information it can establish.

First, the manufacturing-side contribution rests on a discrete manufacturing simulation context, rather than in a broader real-world manufacturing setting. Paper I makes production structure, buffering, coupling, and event-driven behavior inspectable under stated assumptions, which is precisely why it is useful for the present licentiate. However, that same strength also limits the transfer of its findings. The study does not cover the full diversity of manufacturing topologies, process physics, sensing strategies, or plant-specific logging policies. Nor does it instantiate the full observation stack through which industrial records are produced. Its claims should be read as explanatory and mechanism-oriented, not as predictive of all manufacturing settings. Beyond simulation scope, the thesis addresses two observation conditions (irregular timing and missingness). Operational manufacturing data carries additional structural challenges outside the present scope:

- class imbalance, in which normal operation heavily outweighs fault states;
- machine-to-machine variability across nominally identical equipment; and
- label scarcity from rare fault events and retrospective maintenance records.

Further challenges include:

- measurement noise from sensing and transmission;
- sensor drift and recalibration over equipment lifetimes; and
- non-stationary behavior or concept drift under tool wear, recipe changes, and operating-regime shifts [30], [129]–[132].

These are named structural properties of industrial monitoring data, not exceptional deviations, and they represent a further gap between the present scope and deployment-ready modeling.

Second, the model-side contribution reflects the scope and limitations of the reviewed literature. Although Paper II follows a structured and transparent survey procedure, it cannot be fully exhaustive in a rapidly moving field with overlapping terminology and fragmented application areas. In addition, much of the broader empirical literature on irregular and incomplete time series has been developed in a limited set of benchmark domains, especially healthcare [72], [93]. In the PhysioNet 2012 dataset, for example, the mean missing rate across variables exceeds 80%, with some variables absent more than 95% of the time at any given timestamp [133]. The quantitative sparsity is comparable to what manufacturing alarm logs exhibit, but the semantics differ: clinical absence reflects decision-driven measurement, while manufacturing absence frequently reflects exception-reported logging or outage. The meaning of timing gaps, masks, and absences cannot be assumed to transfer unchanged even when the represented tensors look similar.

A connected limitation concerns the structural observation made in Section 5.1.3. Chapter 2 catalogues observation conditions beyond irregularity and missingness: multiple notions of recorded time, semantic heterogeneity across record types, and cross-node asynchrony without a shared reference axis. The present empirical work examines only the first two. The structural claim in Section 5.1.3 therefore rests on the surveyed assumption space applied to the catalogued conditions, not on direct empirical evidence that single-pipeline neural modeling fails under all coexisting conditions. The claim is a structural reading of the assumption space, not a feasibility verdict on specific manufacturing learning tasks.

Third, the thesis remains descriptive and integrative rather than comparative in the strong empirical sense. It clarifies this interface, but it does not yet test multiple method families under one fully controlled manufacturing observation regime. As a result, no claim is made here about which method is superior under stated manufacturing conditions, nor is it determined how strongly particular representation choices affect downstream performance in a controlled benchmark [62], [93], [107]. That gap is not a weakness relative to the stated aim, but it is a real boundary on what can be concluded from the present licentiate.

Fourth, the thesis clarifies this interface and offers decision-making insights, but stops short of full industrial decision-support validation. The present licentiate addresses one upstream requirement for that vision but does not yet address plant-level deployment constraints, human-machine interface design, maintenance decision processes, or operational validation. The strongest supported claims remain methodological and conditional: they concern how observation conditions can arise, how methods place assumptions, and why the interface between these two is a precondition for credible comparison [12], [36], [93], [134], [135].

5.3 Practical Implications

The recommendation that follows from this licentiate is to treat the observation regime as a deliberate design input to a manufacturing learning pipeline, on the same footing as the model architecture or training objective: it concerns what the data are, not only what they contain. The four regimes named in Section 4.2.3.3 may give an engineer a starting point. The mappings below should be read as structural alignments between regime and assumption set, not as benchmarked recommendations.

Alarms, state transitions, and work orders are often timestamped events whose inter-arrival times may carry signal, placing them in the EVT regime where event-time modeling provides an appropriate family of assumptions. Temperatures, vibration traces, and torque profiles can often be read as samples from underlying continuous dynamics, closer to TRJ/SYN where continuous-time latent models are worth considering. supervisory control and data acquisition (SCADA) streams and historian traces are nominally periodic but may be interrupted by outages, asynchronous updates, or logging irregularities, resembling CON/MUL; gap-aware sequence models with mask-retaining encoders provide a relevant starting point. Condition-triggered inspections and planned-maintenance reports have decision-coupled timing, aligning them with EPS, so methods developed on clinical electronic health record (EHR) benchmarks may be informative. However, quantitative sparsity does not imply semantic transfer, and decision-coupled timing may also bring labels or intervention effects with it.

Analytics that comprehensively support engineers, as the vision in Section 1.2 outlines, may draw on event-coded alarms, continuous telemetry, and inspection records at once by surfacing unknown patterns, and the operational signal is typically visible only across that combined view. A one-model-fits-all architecture for this setting is not evident among the surveyed families, and case-by-case analysis of which assumption set fits which record remains necessary. This licentiate contributes a prior step toward that analysis: a vocabulary for naming the regime, a structural mapping of which assumption sets may fit which regime, and the recognition that mixed-regime records can turn the records-to-representation stage into a modeling problem of its own.

5.4 Future Directions

The future work that follows from this licentiate addresses the unresolved issues identified in Sections 5.2 and 5.3. It can be grouped into three directions, ordered from extending the descriptive basis, through making comparison itself a design problem, to connecting both to downstream manufacturing tasks.

Broadening the manufacturing-side condition space. The first direction extends the controlled generative case used here to a wider range of production structures, observation policies, and timestamp semantics, including source-dependent availability patterns [53], [136]. Richer simulated settings together with experimental and industrial traces would allow controlled assumptions to be compared against observed practice, testing how far the present interface view travels across different manufacturing observation regimes.

Comparison and benchmark design as a methodological problem. The second direction uses the present descriptive understanding to inform how comparison itself is set up. Useful manufacturing benchmarks should preserve, or at minimum report, the observation assumptions that bear on later comparison [90], [99], [102]: timing structure, masking logic, record construction, timestamp meaning, and the relation between generated conditions and represented objects. On that basis, controlled comparative studies should treat representation choices and assumption placement as explicit study factors [36], [93], [102], examining how encoding, masking, interpolation, discretization, and objective design interact with specific observation regimes. A specific candidate response is to treat the records-to-representation transition as an explicit modeling stage with separately statable assumption boundaries, composed with time-dependent learning over the resulting representation rather than folded into a single end-to-end model. Paper II's input-side encoder family partially embodies this factoring; whether it is more defensible than alternatives such as continuous-time latent models absorbing the same function differently is an open empirical question, and a defensible benchmark setup is a prerequisite for answering it.

Connection to downstream manufacturing tasks. The third direction connects the present framework to data-driven manufacturing tasks such as monitoring and anomaly detection, especially under heterogeneous signals and limited labeling conditions [18], [74]. In that setting, the relation between representation, target construction, and operational usefulness becomes empirically testable, and provides a more defensible path toward later claims about artificial intelligence (AI)-supported interpretation than transferring task formulations unchanged from benchmark domains with different observation semantics. A related open question concerns where large language models and time-series foundation models fit within manufacturing AI pipelines, with current evidence suggesting purpose-built time-series foundation models trained on numerical temporal data are stronger candidates for the core modeling role than generic large language model (LLM) [84], [87], [88], while LLM may serve as an orchestration layer [14], [86].

Moving closer to stronger practical claims will require evidence about how records are generated and interpreted in real operational settings, how model outputs relate to engineering judgment, and how representation choices affect practical use [44], [47], [99]. The present licentiate is best used as a foundation for that later benchmark, experimental, and application work, and not as a substitute for it.

Chapter 6

Conclusions

This licentiate thesis clarifies a methodological interface between manufacturing cyber-physical systems and neural time-series analysis. A controlled simulation study and a structured survey of neural methods address it from the data-generation and modeling sides, respectively.

The thesis makes three specific contributions:

- *On the manufacturing side*, it shows that ordinary production-structure mechanisms in the simulated discrete-manufacturing setting studied here can generate temporally uneven, asynchronous, and dependent traces such as machine states and buffer levels.
- *On the model side*, it distinguishes irregular sampling from observational missingness as related but distinct properties of recorded data, and shows that this distinction matters for how neural methods should be represented, compared, and evaluated.
- *In synthesis*, it connects the manufacturing-side case to neural approaches through the records-to-representation interface, letting methods be compared by how they represent time, value availability, and between-observation dynamics rather than by architecture labels, and shows that at this transition, the surveyed method families' assumptions do not jointly cover the catalogued conditions.

Taken together, the thesis recasts the records-to-representation interface from a preprocessing antecedent into a modeling site. Production structure generates temporally uneven traces under stated assumptions; neural methods carry their timing and absence commitments in different places, each admitting only part of the conditions the combined records carry.

Bibliography

[1] M. P. Groover, *Automation, Production Systems, and Computer-integrated Manufacturing 2nd ed.* Emerald Group Publishing Limited, 2002.

[2] A. M. Law, W. D. Kelton and W. D. Kelton, *Simulation modeling and analysis.* Mcgraw-hill New York, 2007, vol. 3.

[3] E. M. Martínez, P. Ponce, I. Macias and A. Molina, "Automation pyramid as constructor for a complete digital twin, case study: A didactic manufacturing system," *Sensors*, vol. 21, no. 14, p. 4656, 2021. DOI: 10.3390/s21144656.

[4] International Society of Automation, *Ansi/isa-95.00.01-2025 (iec 62264-1 mod): Enterprise-control system integration – part 1: Models and terminology*, International Society of Automation, Research Triangle Park, NC, 2025. [Online]. Available: https://www.isa.org/products/ansi-isa-95-00-01-2025-iec-62264-1-mod-enterprise.

[5] J. Bokrantz, A. Skoogh, D. Lämkull, A. Hanna and T. Perera, "Data quality problems in discrete event simulation of manufacturing operations," *Simulation*, vol. 94, no. 11, pp. 1009–1025, 2018. DOI: 10.1177/0037549717742954.

[6] M. Gerdes, K. Gross and G. C. Wang, "Electronic prognostics innovations for applications to aerospace systems," in *2023 IEEE Aerospace Conference*, IEEE, 2023, pp. 1–9.

[7] G. C. Wang and K. Gross, "Real time empirical synchronization of iot signals for improved ai prognostics," in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, IEEE, 2018, pp. 954–959.

[8] ISO, *Condition monitoring and diagnostics of machines — General guidelines*, Geneva: International Organization for Standardization, 2018.

[9] IAEA, "Implementation strategies and tools for condition based maintenance at Nuclear Power Plants," International Atomic Energy Agency, Vienna, Tech. Rep. IAEA-TECDOC-1551, 2007. [Online]. Available: https://www-pub.iaea.org/MTCD/Publications/PDF/te_1551_web.pdf.

[10] Y. Lu, K. C. Morris and S. Frechette, "Current standards landscape for smart manufacturing systems," National Institute of Standards and Technology, NIST Interagency/Internal Report (NISTIR) 8107, 2016. DOI: 10.6028/NIST.IR.8107. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8107.pdf.

[11] H. Reinhardt, M. Mahdaviasl, B. Prell *et al.*, "Efficient data acquisition for traceability and analytics," *Procedia CIRP*, vol. 122, pp. 73–78, 2024, 31st CIRP Conference on Life Cycle Engineering (LCE 2024). DOI: 10.1016/j.procir.2024.01.011.

[12] E. Mosqueira-Rey, E. Hernández-Pereira, D. Alonso-Ríos, J. Bobes-Bascarán and Á. Fernández-Leal, "Human-in-the-loop machine learning: A state of the art," *Artificial Intelligence Review*, vol. 56, pp. 3005–3054, 2023. DOI: 10.1007/s10462-022-10246-w.

[13] K. Lång, V. Josefsson, A.-M. Larsson *et al.*, "Artificial intelligence-supported screen reading versus standard double reading in the mammography screening with artificial intelligence trial (masai)," *The Lancet Oncology*, vol. 24, no. 8, pp. 936–944, 2023. DOI: 10.1016/S1470-2045(23)00298-X.

[14] Y. Zhang, M. Wang, X. Zhang *et al.*, "LLM-TSFD: An industrial time series human-in-the-loop fault diagnosis method based on a large language model," *Expert Systems with Applications*, vol. 273, p. 126 093, 2025. DOI: 10.1016/j.eswa.2024.126093.

[15] M. Mayr, S. Luftensteiner and G. C. Chasparis, "Abstracting process mining event logs from process-state data to monitor control-flow of industrial manufacturing processes," *Procedia Computer Science*, vol. 200, pp. 1442–1450, 2022. DOI: 10.1016/j.procs.2022.01.345.

[16] A. Farooqui, K. Bengtsson, P. Falkman and M. Fabian, "Towards data-driven approaches in manufacturing: An architecture to collect sequences of operations," *International Journal of Production Research*, vol. 58, no. 16, pp. 4947–4963, 2020. DOI: 10.1080/00207543.2020.1735660.

[17] J. Huang, B. Yang, K. Yin and J. Xu, "DNA-T: Deformable neighborhood attention transformer for irregular medical time series," *IEEE Journal of Biomedical and Health Informatics*, vol. 28, no. 7, pp. 4224–4237, 2024. DOI: 10.1109/JBHI.2024.3395446.

[18] S. Tipirneni and C. K. Reddy, "Self-supervised transformer for sparse and irregularly sampled multivariate clinical time-series," *ACM Trans. Knowl. Discov. Data*, vol. 16, no. 6, 2022. DOI: 10.1145/3516367.

[19] M. Schirmer, M. Eltayeb, S. Lessmann and M. Rudolph, "Modeling irregular time series with continuous recurrent units," in *International conference on machine learning*, PMLR, 2022, pp. 19 388–19 405.

[20] S. N. Shukla and B. M. Marlin, "A survey on principles, models and methods for learning from irregularly sampled time series," *arXiv preprint arXiv:2012.00168*, 2020.

[21] E. T. Bekar, P. Nyqvist and A. Skoogh, "An intelligent approach for data pre-processing and analysis in predictive maintenance with an industrial case study," *Advances in Mechanical Engineering*, vol. 12, no. 5, p. 1 687 814 020 919 207, 2020.

[22] Z. A. Khan, D. Shin, D. Bianculli and L. Briand, "Impact of log parsing on deep learning-based log anomaly detection," *Empirical Software Engineering*, 2024.

[23] J. Zhu, S. He, J. Liu *et al.*, "Tools and benchmarks for automated log parsing," *IEEE/ACM International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, 2019.

[24] D. Dakić, D. Stefanović, T. Vučković, M. Zizakov and B. Stevanov, "Event log data quality issues and solutions," *Mathematics*, vol. 11, no. 13, p. 2858, 2023. DOI: `10.3390/math11132858`.

[25] P. B. Weerakody, K. W. Wong, G. Wang and W. Ela, "A review of irregular time series data handling with gated recurrent neural networks," *Neurocomputing*, vol. 441, pp. 161–178, 2021.

[26] D. Adhikari, W. Jiang, J. Zhan *et al.*, "A comprehensive survey on imputation of missing data in internet of things," *ACM Comput. Surv.*, vol. 55, no. 7, 2022. DOI: `10.1145/3533381`.

[27] J. L. Alcaraz and N. Strodthoff, "Diffusion-based time series imputation and forecasting with structured state space models," *Transactions on Machine Learning Research*, 2023.

[28] Z. Che, S. Purushotham, K. Cho, D. Sontag and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific Reports*, vol. 8, no. 1, p. 6085, 2018. DOI: `10.1038/s41598-018-24271-9`.

[29] V. Gupta, S. Bedathur, S. Bhattacharya and A. De, "Learning temporal point processes with intermittent observations," in *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 130, 2021, pp. 3790–3798.

[30] J. Xie, L. Sun and Y. F. Zhao, "On the data quality and imbalance in machine learning-based design and manufacturing—a systematic review," *Engineering*, vol. 45, pp. 105–131, 2025. DOI: `10.1016/j.eng.2024.03.017`.

[31] I. Koren, M. Jarke, J. Michael *et al.*, "Navigating the data model divide in smart manufacturing: An empirical investigation for enhanced ai integration," in *Enterprise, Business-Process and Information Systems Modeling*, ser. Lecture Notes in Business Information Processing, H. van der Aa, D. Bork, R. Schmidt and A. Sturm, Eds., vol. 511, Cham: Springer, 2024, pp. 275–290. DOI: `10.1007/978-3-031-61007-3_21`.

[32] G. Michalos, S. Makris, N. Papakostas, D. Mourtzis and G. Chryssolouris, "Automotive assembly technologies review: Challenges and outlook for a flexible and adaptive approach," *CIRP Journal of Manufacturing Science and Technology*, vol. 2, no. 2, pp. 81–91, 2010. DOI: 10.1016/j.cirpj.2009.12.001.

[33] S. J. Hu, J. Ko, L. Weyand *et al.*, "Assembly system design and operations for product variety," *CIRP Annals*, vol. 60, no. 2, pp. 715–733, 2011. DOI: 10.1016/j.cirp.2011.05.004.

[34] H. Haskamp, F. Orth, J. Wermann and A. W. Colombo, "Implementing an OPC UA interface for legacy PLC-based automation systems using the Azure cloud: An ICPS-architecture with a retrofitted RFID system," in *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, 2018, pp. 115–121. DOI: 10.1109/ICPHYS.2018.8387646.

[35] A. Birk, Y. Wilhelm, S. Dreher, C. Flack, P. Reimann and C. Gröger, "A real-world application of process mining for data-driven analysis of multi-level interlinked manufacturing processes," *Procedia Cirp*, vol. 104, pp. 417–422, 2021.

[36] S. Li, S.-h. Yang, Y. Cao, X. Jiang and C. Zhou, "A benchmark of industrial polymerization process for thermal runaway process monitoring," *Process Safety and Environmental Protection*, vol. 193, pp. 353–363, 2025.

[37] B. Hollifield, "Understanding and applying the ansi/isa-18.2 alarm management standard," PAS, Houston, TX, Tech. Rep., 2010, White paper. [Online]. Available: https://www.isa.org/getmedia/55b4210e-6cb2-4de4-89f8-2b5b6b46d954/PAS-Understanding-ISA-18-2.pdf.

[38] S. Jaskó, A. Skrop, T. Holczinger, T. Chován and J. Abonyi, "Development of manufacturing execution systems in accordance with industry 4.0 requirements: A review of standard- and ontology-based methodologies and tools," *Computers in Industry*, vol. 123, p. 103 300, 2020. DOI: 10.1016/j.compind.2020.103300. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361520305340.

[39] J. Lee, B. Bagheri and H.-A. Kao, "A cyber-physical systems architecture for industry 4.0-based manufacturing systems," *Manufacturing Letters*, vol. 3, pp. 18–23, 2015, ISSN: 2213-8463. DOI: https://doi.org/10.1016/j.mfglet.2014.12.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S221384631400025X.

[40] T. Czvetkó and J. Abonyi, "Data sharing in Industry 4.0 - AutomationML, B2MML and International Data Spaces-based solutions," *Journal of Industrial Information Integration*, vol. 33, p. 100 438, 2023. DOI: 10.1016/j.jii.2023.100438. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2452414X23000110.

[41] NASPI, "Real-time application of synchrophasors for improving reliability," North American SynchroPhasor Initiative, Tech. Rep., Oct. 2010. [Online]. Available: `https://www.naspi.org/sites/default/files/reference_documents/rapir_final_20101017.pdf`.

[42] R. K. Pandit, D. Infield and J. Carroll, "SCADA data for wind turbine data-driven condition/performance monitoring: A review on state-of-art, challenges and future directions," *Wind Engineering*, vol. 47, no. 2, pp. 422–441, 2023. DOI: `10.1177/0309524X221124031`.

[43] A. Lundius and P. Johansson, "Initial assessment of manufacturing execution systems: Development of a methodology to define business needs and functional requirements," Master's thesis, KTH Royal Institute of Technology, 2019. [Online]. Available: `https://kth.diva-portal.org/smash/get/diva2:1374272/FULLTEXT01.pdf`.

[44] A. Goknil, P. Nguyen, S. Sen *et al.*, "A systematic review of data quality in cps and iot for industry 4.0," *ACM Computing Surveys*, vol. 55, no. 14s, pp. 1–38, 2023.

[45] M. Åkerman, J. Stahre, U. Engström *et al.*, "Technical interoperability for machine connectivity on the shop floor," *Technologies*, vol. 6, no. 3, p. 57, 2018. DOI: `10.3390/technologies6030057`.

[46] J. Yan, Y. Meng, L. Lu and L. Li, "Industrial big data in an industry 4.0 environment: Challenges, schemes, and applications for predictive maintenance," *Ieee Access*, vol. 5, pp. 23 484–23 491, 2017.

[47] T. P. Raptis, A. Passarella and M. Conti, "Data management in industry 4.0: State of the art and open challenges," *IEEE Access*, vol. 7, pp. 97 052–97 093, 2019.

[48] Rockwell Automation, *FactoryTalk Historian to Historian Interface User Guide*, Document no. H2H-UM001E-EN-E. Accessed 2026-03-11, Rockwell Automation, 2021. [Online]. Available: `https://literature.rockwellautomation.com/idc/groups/literature/documents/um/h2h-um001_-en-e.pdf`.

[49] W. P. M. H. Heemels, J. H. Sandee and P. P. J. V. D. Bosch, "Analysis of event-driven controllers for linear systems," *International Journal of Control*, vol. 81, no. 4, pp. 571–590, 2008.

[50] K. Gross and G. C. Wang, "Ai decision support prognostics for iot asset health monitoring, failure prediction, time to failure," in *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, IEEE, 2019, pp. 244–248.

[51] V. Jirkovský, M. Obitko and V. Mařík, "Understanding data heterogeneity in the context of cyber-physical systems integration," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 660–667, 2017. DOI: `10.1109/TII.2016.2596101`.

[52] International Electrotechnical Commission, “IEC 62541-11:2020 OPC Unified Architecture – Part 11: Historical Access,” International Electrotechnical Commission, Tech. Rep., 2020, Second edition. [Online]. Available: https://webstore.iec.ch/en/publication/61129.

[53] V. Schiffer, *The common industrial protocol (cip™) and the family of cip networks*, PUB00123R1, On behalf of Rockwell Automation, ODVA, Inc., Feb. 2016. [Online]. Available: https://www.odva.org/wp-content/uploads/2020/06/PUB00123R1_Common-Industrial_Protocol_and_Family_of_CIP_Networks.pdf.

[54] A. J. Goodwin, D. Eytan, W. Dixon *et al.*, “Timing errors and temporal uncertainty in clinical databases—a narrative review,” *Frontiers in Digital Health*, vol. Volume 4 - 2022, 2022, ISSN: 2673-253X. DOI: 10.3389/fdgth.2022.932599. [Online]. Available: https://www.frontiersin.org/journals/digital-health/articles/10.3389/fdgth.2022.932599.

[55] Z. Huang, P. Du, D. Kosterev and B. Yang, “Data quality issues for synchrophasor applications, Part I: A review,” *Journal of Modern Power Systems and Clean Energy*, vol. 4, no. 3, pp. 342–352, 2016. DOI: 10.1007/s40565-016-0222-5.

[56] M. Landauer, S. Onder, F. Skopik and M. Wurzenberger, “Deep learning for anomaly detection in log data: A survey,” *Machine Learning with Applications*, vol. 12, p. 100 470, 2023. DOI: 10.1016/j.mlwa.2023.100470.

[57] J. Wang, T. Chen and B. Huang, “Multirate sampled-data systems: Computing fast-rate models,” *Journal of Process Control*, vol. 14, no. 1, pp. 79–88, Feb. 2004. DOI: 10.1016/S0959-1524(03)00033-7.

[58] R. Telatko, G. Maurer and D. Reichelt, “Event-based data acquisition for production process analyses: A systematic mapping study,” in *2021 IEEE 13th International Conference on Computer and Automation Engineering (ICCAE)*, Melbourne, Australia: IEEE, Mar. 2021, pp. 105–110. DOI: 10.1109/ICCAE51876.2021.9426134.

[59] K.-J. Jeong, J.-D. Park, K. Hwang, S.-L. Kim and W.-Y. Shin, “Two-stage deep anomaly detection with heterogeneous time series data,” *IEEE Access*, vol. 10, pp. 13 704–13 714, 2022.

[60] J. Zhan, C. Wu, C. Yang, Q. Miao and X. Ma, “Hfn: Heterogeneous feature network for multivariate time series anomaly detection,” *Information Sciences*, vol. 670, p. 120 626, 2024.

[61] A. Bezerra, I. Silva, L. A. Guedes, D. Silva, G. Leitão and K. Saito, “Extracting value from industrial alarms and events: A data-driven approach based on exploratory data analysis,” *Sensors*, vol. 19, no. 12, p. 2772, 2019. DOI: 10.3390/s19122772.

[62] D. J. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

[63] D. B. Rubin, “Inference and missing data,” *Biometrika*, vol. 63, no. 3, pp. 581–592, 1976.

[64] R. J. A. Little and D. B. Rubin, *Statistical Analysis with Missing Data.* John Wiley & Sons, 2019.

[65] X. Wang, H. Zhang, P. Wang *et al.*, “An observed value consistent diffusion model for imputing missing values in multivariate time series,” in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, 2023, 2409–2418. DOI: 10.1145/3580305.3599257.

[66] E. Oh, T. Kim, Y. Ji and S. Khyalia, “STING: Self-attention based time-series imputation networks using GAN,” in *2021 IEEE International Conference on Data Mining (ICDM)*, 2021, pp. 1264–1269. DOI: 10.1109/ICDM51629.2021.00155.

[67] Y. Tashiro, J. Song, Y. Song and S. Ermon, “CSDI: Conditional score-based diffusion models for probabilistic time series imputation,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 24 804–24 816, 2021.

[68] S. Kamm *et al.*, “A survey on machine learning based analysis of heterogeneous data in the industrial automation domain,” *Computers in Industry*, vol. 149, p. 103 928, 2023. DOI: 10.1016/j.compind.2023.103928.

[69] D. Zha *et al.*, “Data-centric artificial intelligence: A survey,” *arXiv preprint arXiv:2303.10158*, 2023.

[70] P. J. Brockwell and R. A. Davis, *Time series: theory and methods.* Springer Science & Business Media, 2009.

[71] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control.* San Francisco, CA: Holden-Day, 1976.

[72] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, 2009.

[73] A. Zeng, M. Chen, L. Zhang and Q. Xu, “Are transformers effective for time series forecasting?” In *Proceedings of the AAAI conference on artificial intelligence*, 2023, pp. 11 121–11 128.

[74] A. Garcia, X. Oregui, J. Franco, U. Arrieta, J. Ferreres and J. A. Valencia, “Time series manufacturing data edge monitoring and visualization to support industrial maintenance teams,” *SN Computer Science*, vol. 5, no. 1, p. 131, 2024.

[75] C. Xiao, Z. Gou, W. Tai, K. Zhang and F. Zhou, “Imputation-based time-series anomaly detection with conditional weight-incremental diffusion models,” in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, 2023, 2742–2751. DOI: 10.1145/3580305.3599391.

[76] C. Herrera, F. Krach and J. Teichmann, “Neural jump ordinary differential equations: Consistent continuous-time prediction and filtering,” in *International Conference on Learning Representations*, 2021.

[77] R. N. Bracewell, *The Fourier Transform and Its Applications*. McGraw-Hill, 1986.

[78] Y. Lee, E. Jun, J. Choi and H.-I. Suk, "Multi-view integrative attention-based deep representation learning for irregular clinical time-series data," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 8, pp. 4270–4280, 2022. DOI: 10.1109/JBHI.2022.3172549.

[79] T. Blu and M. Unser, "Quantitative fourier analysis of approximation techniques: Part i—interpolators and projectors," *IEEE Transactions on Signal Processing*, vol. 47, no. 10, pp. 2783–2795, Oct. 1999. DOI: 10.1109/78.790659.

[80] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[81] Z. C. Lipton, D. Kale and R. Wetzel, "Directly modeling missing data in sequences with rnns: Improved classification of clinical time series," in *Machine learning for healthcare conference*, PMLR, 2016, pp. 253–270.

[82] R. Deng, M. A. Brubaker, G. Mori and A. Lehrmann, "Continuous latent process flows," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2021, pp. 5162–5173.

[83] Y. Chen, K. Ren, Y. Wang, Y. Fang, W. Sun and D. Li, "Contiformer: Continuous-time transformer for irregular time series modeling," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2023, pp. 47 143–47 175.

[84] M. Tan *et al.*, "Are language models actually useful for time series forecasting?" In *Advances in Neural Information Processing Systems*, vol. 37, 2024.

[85] M. A. Merrill *et al.*, "Language models still struggle to zero-shot reason about time series," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024.

[86] S. Freire *et al.*, "Harnessing large language models for cognitive assistants in factories," in *Proceedings of the 5th International Conference on Conversational User Interfaces*, 2023. DOI: 10.1145/3571884.3604313.

[87] M. Goswami *et al.*, "MOMENT: A family of open time-series foundation models," in *International Conference on Machine Learning*, 2024.

[88] A. F. Ansari *et al.*, "Chronos: Learning the language of time series," *arXiv preprint arXiv:2403.07815*, 2024.

[89] J. Xu, H. Wu, J. Wang and M. Long, "Anomaly transformer: Time series anomaly detection with association discrepancy," in *International Conference on Learning Representations*, 2022.

[90] R. Wu and E. J. Keogh, "Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 3, pp. 2421–2429, 2023. DOI: 10.1109/TKDE.2021.3112126.

[91] R. G. Sargent, “Verification and validation of simulation models,” in *Proceedings of the 2010 winter simulation conference*, IEEE, 2010, pp. 166–183.

[92] P. V. Lopes, L. Silveira, R. D. Guimaraes Aquino, C. H. Ribeiro, A. Skoogh and F. A. Verri, “Synthetic data generation for digital twins: Enabling production systems analysis in the absence of data,” 2024. DOI: 10.1080/0951192x.2024.2322981.

[93] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[94] J. Goh, S. Adepu, K. N. Junejo and A. Mathur, “A dataset to support research in the design of secure water treatment systems,” in *International Conference on Critical Information Infrastructures Security (CRITIS)*, 2016. DOI: 10.1007/978-3-319-71368-7_8.

[95] M. Aslam, D. Kim, H. Lee *et al.*, “An optimized anomaly detection framework in industrial control systems,” *Scientific Reports*, vol. 15, 2025. DOI: 10.1038/s41598-025-12775-0.

[96] O. Diallo, N. Hammerla, K. Bhavsar and S. Mouton, “Identifying benchmarks for failure prediction in Industry 4.0,” *Applied Sciences*, vol. 11, no. 6, p. 2555, 2021. DOI: 10.3390/app11062555.

[97] M. McCann and A. Johnston, *SECOM dataset*, UCI Machine Learning Repository, 2008.

[98] D. Kus, E. Wagner, J. Pennekamp *et al.*, “A false sense of security? revisiting the state of machine learning-based industrial intrusion detection,” in *Proceedings of the 8th ACM on Cyber-Physical System Security Workshop*, 2022, pp. 73–84.

[99] A. Melo, M. M. Câmara, N. Clavijo and J. C. Pinto, “Open benchmarks for assessment of process monitoring and fault diagnosis techniques: A review and critical analysis,” *Computers & Chemical Engineering*, vol. 165, p. 107 964, 2022.

[100] J. Audibert, P. Michiardi, F. Guyard, S. Marti and M. A. Zuluaga, “Usad: Unsupervised anomaly detection on multivariate time series,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Association for Computing Machinery, 2020, 3395–3404. DOI: 10.1145/3394486.3403392.

[101] A. P. Mathur and N. O. Tippenhauer, “SWaT: A water treatment testbed for research and training on ICS security,” in *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*, IEEE, 2016, pp. 31–36.

[102] Q. Liu and J. Paparrizos, "The elephant in the room: Towards a reliable time-series anomaly detection benchmark," in *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave *et al.*, Eds., vol. 37, Curran Associates, Inc., 2024, pp. 108 231–108 261. DOI: 10.52202/079017-3437. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2024/file/c3f3c690b7a99fba16d0efd35cb83b2c-Paper-Datasets_and_Benchmarks_Track.pdf.

[103] B. Wang, Y. Lei, N. Li and N. Li, "A hybrid prognostics approach for estimating remaining useful life of rolling element bearings," *IEEE Transactions on Reliability*, vol. 69, no. 1, pp. 401–412, 2020. DOI: 10.1109/TR.2018.2882682.

[104] B. Veloso, J. Gama, B. Malheiro and J. Vinagre, "The MetroPT dataset for predictive maintenance," *Scientific Data*, vol. 9, p. 764, 2022. DOI: 10.1038/s41597-022-01877-3.

[105] A. Saxena, K. Goebel, D. Simon and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *International Conference on Prognostics and Health Management (PHM 2008)*, 2008. DOI: 10.1109/PHM.2008.4711414.

[106] B. Kim, M. A. Alawami, E. Kim, S. Oh, J. Park and H. Kim, "A comparative study of time series anomaly detection models for industrial control systems," *Sensors*, vol. 23, no. 3, p. 1310, 2023.

[107] L. T. Blessing and A. Chakrabarti, *DRM, a design research methodology*. Springer, 2009.

[108] J. W. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*, 4th. SAGE Publications, 2014.

[109] M. Ketokivi and S. Mantere, "Two strategies for inductive reasoning in organizational research," *Academy of Management Review*, vol. 35, no. 2, pp. 315–333, 2010.

[110] A. Dubois and L.-E. Gadde, "Systematic combining: An abductive approach to case research," *Journal of Business Research*, vol. 55, no. 7, pp. 553–560, 2002.

[111] R. J. Wieringa, *Design Science Methodology for Information Systems and Software Engineering*. Berlin, Heidelberg: Springer, 2014, ISBN: 978-3-662-43838-1. DOI: 10.1007/978-3-662-43839-8.

[112] N. Cross, "Designerly ways of knowing: Design discipline versus design science," *Design Issues*, vol. 17, no. 3, pp. 49–55, 2001.

[113] A. R. Hevner, S. T. March, J. Park and S. Ram, "Design science in information systems research," *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004.

[114] P. Mallioris, E. Aivazidou and D. Bechtsis, "Predictive maintenance in industry 4.0: A systematic multi-sector mapping," *CIRP journal of manufacturing science and technology*, vol. 50, pp. 80–103, 2024.

[115] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989. DOI: 10.1109/5.24143.

[116] A. G. Hawkes, "Spectra of some self-exciting and mutually exciting point processes," *Biometrika*, vol. 58, no. 1, pp. 83–90, 1971.

[117] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, 2014, pp. 1–10.

[118] M. Jin, H. Y. Koh, Q. Wen *et al.*, "A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 10 466–10 485, 2024. DOI: 10.1109/TPAMI.2024.3443141.

[119] M. Templier and G. Paré, "A framework for guiding and evaluating literature reviews," *Communications of the Association for Information Systems*, vol. 37, pp. 112–137, 2015.

[120] J. Webster and R. T. Watson, "Analyzing the past to prepare for the future: Writing a literature review," *MIS Quarterly*, vol. 26, no. 2, pp. xiii–xxiii, 2002.

[121] B. G. Glaser and A. L. Strauss, *The Discovery of Grounded Theory: Strategies for Qualitative Research.* Chicago: Aldine de Gruyter, 1967.

[122] M. P. Deisenroth, A. A. Faisal and C. S. Ong, *Mathematics for Machine Learning.* Cambridge University Press, 2020.

[123] A. Schlemitz and V. Mezhuyev, "Approaches for data collection and process standardization in smart manufacturing: Systematic literature review," *Journal of Industrial Information Integration*, vol. 38, p. 100 578, 2024.

[124] R. Huang and T. Polak, "LOBSTER: Limit order book reconstruction system," Humboldt-Universität zu Berlin, Tech. Rep., 2011, SSRN working paper. DOI: 10.2139/ssrn.1977207. [Online]. Available: https://ssrn.com/abstract=1977207.

[125] I. Silva, G. Moody, D. J. Scott, L. A. Celi and R. G. Mark, "Predicting in-hospital mortality of patients in ICU: The PhysioNet/Computing in Cardiology challenge 2012," in *Computing in Cardiology*, vol. 39, 2012, pp. 245–248.

[126] A. E. W. Johnson, T. J. Pollard, L. Shen *et al.*, "MIMIC-III, a freely accessible critical care database," *Scientific Data*, vol. 3, p. 160 035, 2016. DOI: 10.1038/sdata.2016.35.

[127] Y. Li, R. Yu, C. Shahabi and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *International Conference on Learning Representations (ICLR)*, 2018. [Online]. Available: https://openreview.net/forum?id=SJiHXGWAZ.

[128] E. Todorov, T. Erez and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5026–5033. DOI: `10.1109/IROS.2012.6386109`.

[129] A. T. Oliveira-Filho *et al.*, "Wind turbine SCADA data imbalance: A review of its challenges and mitigation strategies," *Energies*, vol. 18, no. 1, p. 59, 2024. DOI: `10.3390/en18010059`.

[130] K. Kammerer, B. Hoppenstedt, R. Pryss, S. Stökler, J. Gripp and M. Reichert, "Anomaly detections for manufacturing systems based on sensor data—insights into two challenging real-world production settings," *Sensors*, vol. 19, no. 24, p. 5370, 2019. DOI: `10.3390/s19245370`.

[131] F. Pittino, M. Puggl, T. Moldaschl and C. Hirschl, "Automatic anomaly detection on in-production manufacturing machines using statistical learning methods," *Sensors*, vol. 20, no. 8, p. 2344, 2020. DOI: `10.3390/s20082344`.

[132] F. J. Maseda, I. Martija, I. Aramendia, A. J. Garrido and I. Garrido, "Sensors data analysis in supervisory control and data acquisition (SCADA) systems to foresee failures with an undetermined origin," *Sensors*, vol. 21, no. 8, p. 2762, 2021. DOI: `10.3390/s21082762`.

[133] Y. Luo, X. Cai, Y. Zhang, J. Xu *et al.*, "Multivariate time series imputation with generative adversarial networks," *Advances in neural information processing systems*, vol. 31, 2018.

[134] Z. Sun, Z. Zou, W. J. Staszewski and Y. Bao, "A critical review for trustworthy and explainable human–cyber reliability in structural health monitoring," *Sustainability*, vol. 15, no. 8, p. 6389, 2023. DOI: `10.3390/su15086389`.

[135] D. T. Pham, Y. Lim and V. Duong, "Human-ai hybrid paradigm for collaborative air traffic conflict resolution," in *SESAR Innovation Days 2024*, 2024.

[136] J. Pennekamp, A. Belova, T. Bergs *et al.*, "Evolving the digital industrial infrastructure for production: Steps taken and the road ahead," in *Internet of Production: Fundamentals, Applications and Proceedings*, Springer, 2023, pp. 1–26.