https://pubsonline.informs.org/journal/serv

SERVICE SCIENCE

Submitted to *Service Science*

Vol. 00, No. 0, Xxxxx 0000, pp. 000–000

ISSN 2164-3962, EISSN 2164-3970

Digital Twins in Service Systems: The Process Twin Loop in Courts

Abstract. Congested service systems pose challenges for efficiency, equity, and access, with court systems being a particularly salient example. Misalignment between the five faces of a process: designed, expert-perceived, data-reflected, real, and optimal, highlights why static approaches fall short and motivates the vision of a digital twin of the system as a unifying, adaptive model.

We introduce the Process Twin Loop (PTL) as an operational framework that moves from existing information-system data to event logs, process maps, and dashboards, and then conducts baseline simulations and counterfactual evaluations of interventions. The post-intervention state is subsequently fed back into the models for ongoing improvement. Using courts as the anchor setting, we detail each step and outline the remaining requirements for live integration. While real-time connections and optimization are future work, the PTL shows how management analytics can bridge the gap to prescriptive digital twins in congested services. A practical benefit of our approach is rapid model generation from event logs, which enables frequent cycles from data to simulation to decision.

Key words:

1. Introduction

Service systems that face high demand and limited resources often struggle with congestion. In such environments, the need for improvement is clear, yet the ability to test and implement reforms is limited. Courts are a salient example: they operate under persistent congestion, yet reforms are costly, slow to evaluate, and difficult to reverse once implemented (Church et al., 1978; CEPEJ, 2016; Voigt, 2016; Vitkauskas and Dikov, 2017).

Understanding and improving such systems is complicated by the fact that every process has multiple faces. A process can be seen as it is *designed*, as it is *perceived by experts* (each one of them may well see a different face), as it is *reflected in the data*, as it *actually performs in practice* (according to some performance indicators), and as it would look in its *optimal form*. These "five faces" often diverge, e.g., in the case of the courts, the gaps are particularly extensive: envisioned design differs from courtroom practice, administrative records can be incomplete or inconsistent, and optimal flows are rarely attainable under high congestion. This misalignment illustrates why descriptive accounts are not enough, and motivates the need for frameworks that can integrate across perspectives and support data-driven experimentation.

Existing approaches to system improvement have important limitations, particularly when applied to courts. Pilots are expensive and produce results that are often context-specific. Traditional simulation models require months of detailed system analysis, collection of customized data, and calibration, making them difficult to maintain and update. Descriptive dashboards improve visibility but cannot anticipate the consequences of interventions. These approaches fall short in highly congested and complex environments, where decision makers need tools that allow them to understand processes, evaluate interventions, and update models continuously as the system changes (Azaria et al., 2025).

We extend the management analytics framework described by Baron (2021) to introduce the Process Twin Loop (PTL) as a practical framework to automatically implement management analytics in queueing networks. The PTL starts at the descriptive analytics phase and builds, from data in the existing information system *event logs*, process maps, and dashboards. This phase allows to easily visualize past performance of the existing system and analyze it, e.g., by highlighting bottleneck activities. The PTL continues with the predictive analytics phase and automatically specify and train a data-driven simulation model of the current system. This phase allows to (i) easily visualize future performance of the existing system, (ii) identify interventions that look particularly promising to achieve desired process improvements, and (iii) capture the causal congestion context in the existing queueing network system. The PTL then uses this simulation model to perform comparative analytics, such as counterfactual analysis of interventions in the past and future. This phase supports what-if analysis. The next phase of the PTL is prescriptive analytics, where specific changes to improve the system are recommended, chosen, and implemented. This phase brings the PTL to practice, and the outcomes of these implemented changes are fed back into the loop to keep the simulation model current. The PTL thus represents a pathway toward digital twins of service systems that act as a mechanism for continuous learning and improvement.

This approach aligns with recent developments in operations and manufacturing that emphasize the concept of a digital twin: a continuously updated virtual representation of a physical or organizational system used to support monitoring, simulation, and decision-making (Grieves, 2014; Haag and Anderl, 2018; Tao et al., 2019). While digital twins are well established in engineering and production settings, their application to service systems remains limited, where the objects of study are processes and human interactions rather than machines (Zhang et al., 2019). The Process Twin Loop (PTL) contributes to this emerging direction by demonstrating how digital-twin principles can be adapted to complex service environments such as courts.

The courts' setting motivates this work for two reasons. First, courts exemplify the consequences of congestion: backlogs and delays undermine efficiency and access to justice. Second, courts embody multiple forms of complexity: operational, organizational, and incentive-based. This unique combination of complexities makes them particularly challenging for traditional process improvement methods, creating a strong need for an approach that can serve as a bridge from descriptive to prescriptive analytics, without requiring disruptive or irreversible reforms.

Process mining (Van Der Aalst, 2012, 2016; Reinkemeyer, 2020) seeks to bridge the gap between the process as designed or perceived by experts, and the process as it actually performs in practice, by extracting information from the event logs (digital traces of the process) for *process discovery* and process mapping. While originally envisioned as mostly a descriptive tool, several extensions to capture stochastic features and enable data-driven representation of the process as a network of queues have been proposed (Senderovich et al., 2014b, 2015)), with the ultimate goal of using the discovered process for enabling data-driven process simulation (van der Aalst, 2018).

In recent years, several platforms have emerged to operationalize this vision of extending process mining towards simulation and digital twin capabilities. Leading tools such as Celonis and Apromore exemplify this trend, integrating event log analytics with performance prediction and scenario testing modules (Celonis, 2024; Apromore, 2024). These systems demonstrate the growing convergence between process discovery and simulation, yet most remain tailored to business operations and lack domain-specific modeling for complex public service systems such as courts. SiMLQ is another platform that contributes to this emerging landscape as a data-driven simulator specifically designed for creating digital twins for queueing network processes. SiMLQ rapidly transforms case-level event data into process maps, dashboards, and calibrated simulation models, enabling continuous experimentation and learning in highly congested environments.

This paper makes two contributions. First, it introduces the Process Twin Loop (PTL), a management analytics framework that formalizes how organizations can evolve from descriptive process mining toward adaptive, continuously updated simulation models that support prescriptive analytics. Second, it illustrates the implementation of this framework in the context of U.S. Federal District Courts. This case study demonstrates the feasibility of the PTL approach using existing operational data. While the full feedback loop is not yet enacted, the illustration provides proof of concept for the first stages of a data-driven process digital twin.

Roadmap. In Section 2, we describe courts as complex service systems, highlighting the many dimensions of complexity, the dynamics of congestion, and the barriers to reform. In Section 3, we introduce the Process Twin Loop (PTL), explain the gap it addresses, and describe its stages. In Section 4, we illustrate the first five stages of the Process Twin Loop (PTL) on U.S. Court of the Northern District of Illinois data, leveraging software packages designed for digital twins of service systems. Section 5 discusses the remaining gaps and requirements for realizing a live digital twin of courts. Section 6 contains some concluding remarks.

2. Courts as Complex Service Systems

Courts are a central public service system that operates under persistent congestion. Studies across jurisdictions consistently document large backlogs, long queues, and lengthy delays (Church et al., 1978; CEPEJ, 2016; Voigt, 2016). These conditions undermine efficiency, raise costs for litigants, and threaten access to justice as a basic civil right Vitkauskas and Dikov (2017). Congestion is a global and chronic phenomenon, and despite repeated reform efforts, many courts continue to struggle with unresolved cases that last years (Azaria, 2023).

Moreover, the courts as a service system is not only characterized by congestion, but also by complexity (Agmon-Gonnen, 2004; Provine, 1988; Tacha, 1995). This complexity can be described in three areas: Operational complexity, Organizational complexity, and complexity-driving Incentives.

Operational complexity. Court systems are large, geographically dispersed, and employ a broad range of staff in varied roles. They serve diverse populations and process a wide range of case types. Processes are not standardized, and case trajectories are uncertain in both sequence and duration. Small differences in case content or party behavior can drive large variations in processing time. As cases age, they often require additional hearings, motions, and judicial attention, complicating forecasting and scheduling (Azaria and Shamir, 2025).

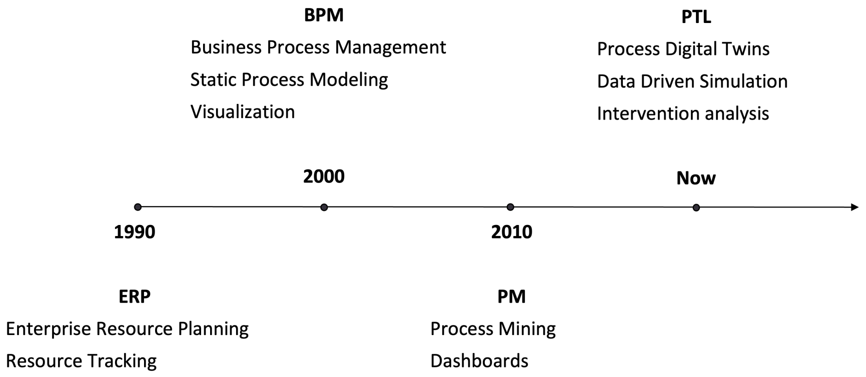
Organizational complexity. Courts operate under governance structures that limit centralized management. Judicial independence, while fundamental to democracy, constrains managerial discretion (Agmon-Gonnen, 2004; Tacha, 1995). In practice, judges have broad autonomy in case management, and administrators have limited authority to intervene. Systemic changes, such as new procedures or case management rules, often require formal legislative or regulatory action (Azaria et al., 2023, 2024).

Incentive complexity. The adversarial nature of litigation creates conflicting objectives. Plaintiffs, defendants, and attorneys may pursue strategies that prolong or accelerate cases. Strategic use of delay, motion practice, or requests for extensions interacts with congestion (Castro et al., 2015; Mitsopoulos and Pelagidis, 2010; Dalton and Singer, 2014). Judges balance fairness and due process with pressures for timeliness, leading to large variation in processing times.

This complexity is compounded by how court performance is measured. A typical case passes through several distinct stages—filing, discovery, trial, and post-judgment—that impose very different workloads on judges and staff. Speedy resolution is valuable in some stages but less so in others, and uniform expectations of efficiency can obscure these differences. Moreover, courts are evaluated by multiple, sometimes conflicting Performance Indicators. For example, while the average "length of stay" of cases is often viewed as an indicator to minimize, the share of cases resolved through mutual agreement is regarded as an important indicator to maximize. Because parties require time to negotiate settlements, and negotiated settlements reduce court workload, intentional slowdowns at certain stages are common and can even reflect effective judicial management rather than inefficiency (Pereira, 2025; Kapelko, 2025).

Congestion emerges from the interaction of these complexities. It is not only driven by rising demand and limited resources, but also by systemic phenomena that worsen when resources are overloaded. Judges and staff in congested systems divide attention between many open cases, and

Figure 1 **Data-driven process management eras.**



evidence shows that multitasking reduces productivity and slows progress (Coviello et al., 2015). Long lead times change the content of work: older cases often require more judicial effort, generate additional motions, and become more complex to resolve. This dynamic creates reinforcing cycles of delay, the congestion vortex (Azaria and Shamir, 2025).

Attempts to address congestion through reforms or pilot programs face substantial barriers. Pilots are costly to design, slow to evaluate, and often do not generalize across jurisdictions. Legal reforms require political and legislative action, making them uncertain and expensive to implement; once enacted, they are difficult to reverse. Organizational constraints and stakeholder resistance add further friction. This underscores the need for analytic approaches that allow reform options and pilot initiatives to be first evaluated virtually before being implemented.

3. The Process Twin Loop (PTL)

The digital twin concept originated in NASA's Apollo program in the 1960s, where engineers updated physical simulators with data to mirror spacecraft conditions in real time. The term digital twin was later introduced by (Grieves, 2002, 2014), who conceptualized the digital twin as a virtual counterpart connected to a physical system through bidirectional data flows, enabling synchronized monitoring, analysis, and control across the product life-cycle.

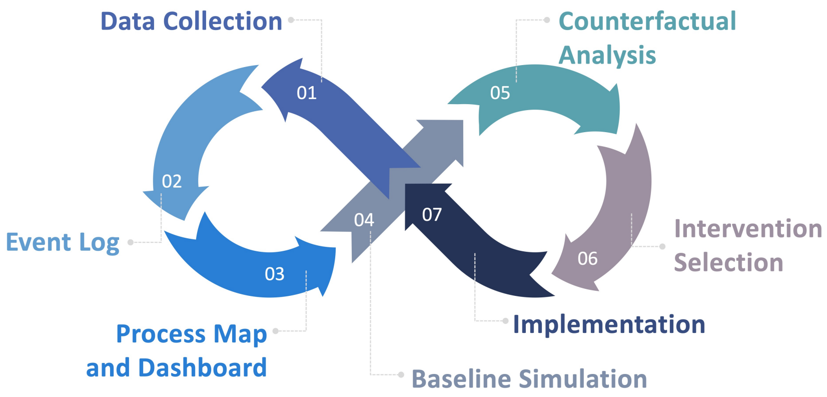
Building on this foundation, digital twins became a cornerstone of cyber–physical integration. They have been used to represent patients, production systems, equipment, supply chains, and assets that can be simulated, optimized, and continuously updated based on real-time operational data (Haag and Anderl, 2018; Negri et al., 2017).

The application of digital twins to service systems is more recent and remains an emerging field. In services, the focus shifts from physical assets to processes and human interactions, requiring the integration of heterogeneous data and behavioral dynamics. Service digital twins seek to capture how work flows through organizations, incorporating both operational and contextual information to support performance improvement (Zhang et al., 2019).

The Process Twin Loop (PTL) builds on this evolution by translating digital-twin principles into the service domain. It models the organization as a living process system whose data can be continuously transformed into models, simulations, and decision-support tools. The Process Twin Loop (PTL) extends process mining into a dynamic, simulation-based feedback loop that connects descriptive insights with prescriptive interventions, enabling continuous organizational learning and improvement.

As illustrated in Figure 1, the PTL builds on successive eras in the evolution of data-driven feedback and improvement in operations. The first era emerged with the spread of computerized record keeping and the integration of operational data across functions, giving rise to *Enterprise Resource Planning (ERP)* systems in the 1990s (Davenport, 1998). ERP systems centralized information flows and standardized reporting, enabling consistent measurement and traceability but providing only static visibility into activities. The second era was characterized by the rise of *Business Process Management (BPM)*, which extended ERP infrastructures to emphasize process design, execution, and monitoring (Dumas et al., 2013). BPM introduced a systematic way to model end-to-end workflows, measure compliance, and institutionalize continuous improvement. The third era began with the advent of *process mining*, which leveraged digital event logs from ERP and BPM systems to algorithmically reconstruct how processes unfold in data (Van Der Aalst, 2012,

Figure 2 **Process Twin Loop (PTL) schematic.**



2016), in hope that these data capture the real process. Process mining bridged information systems, operations management, and data science by transforming transactional traces into empirical insight about operational dynamics.

The PTL represents the next era in this trajectory: a closed-loop analytics framework that not only discovers and visualizes processes from data but also simulates, tests, and continuously recalibrates models as new data arrive, turning static process intelligence into adaptive, evidence-based learning.

As shown in Figure 2, the PTL begins with (1) existing data collected in the organization's information systems. These records are then (2) transformed into an event log that captures case-level sequences of activities, timestamps, and attributes. From the event log, analysts develop (3) process maps and dashboards that provide visibility into how work is actually performed and expose misalignments across the five faces of the process. The event log also serves as the basis to (4) train a simulation model that mimics observed dynamics. With this model, managers and experts can (5) conduct what-if analyses, (6) compare the effectiveness of interventions, and (7) select actions to implement. Once implemented, these actions change the system and generate (8) new data that restart the loop. Steps 1-4 can be automated and feed into steps 5-7 that can be similarly automated, either as a decision support tool with human oversight or completely. Such optimization can be done using either traditional optimization techniques or modern ones, e.g., reinforcement learning.

We note that the automation and adaptivity of the PTL are enabled by advances in machine learning (ML) and artificial intelligence (AI). ML methods not only extract structure and behavioral rules from event data, but can also be used to automatically estimate the different building blocks of a queueing network. AI techniques, including optimization and reinforcement learning, could guide what-if analysis and adaptive model recalibration as new data are observed. Together, these capabilities transform the static and time-consuming modeling task into a continuous and automated management analytics system.

4. Case Study: Process Twin Loop for Northern District of Illinois Court Operations

To demonstrate the PTL in practice, we use the case of the US district court for the Northern District of Illinois (NDIL), one of the busiest U.S. federal district courts, handling thousands of cases annually across a wide range of case types (Azaria et al., 2025). Its scale and workload make it a representative setting for studying court congestion. This demonstration is enabled through collaboration with the SCALES-OKN project, which provides structured access to federal court records and supports research on judicial processes (Pah et al., 2020).

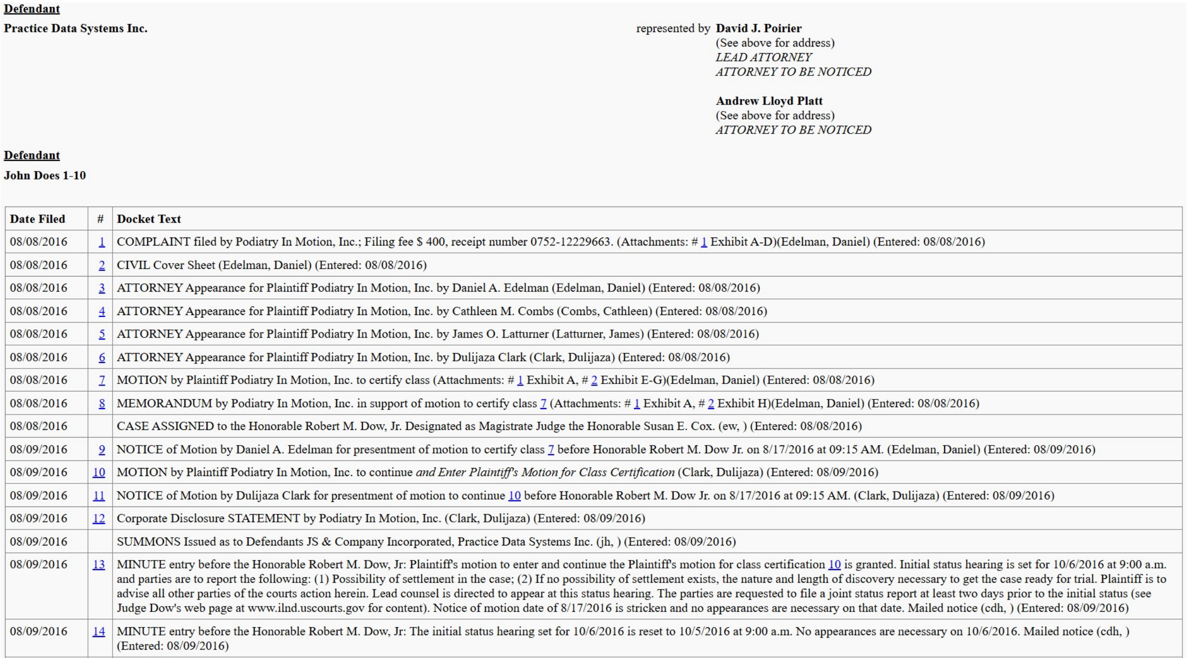
We now describe each stage of the PTL as applied to NDIL data, with illustrative figures and screenshots.

4.1. Data Extraction from Court Information Systems

The primary digital footrace for US federal courts is a collection of dockets. Case dockets are documents that hold information about the case, as well as capture and describe all key events of the case before the court, from the case origination to the final judgment or settlement.

Raw docket text, as presented in Figure 3, is not designed for analytics and is mostly in natural language form, so entries must be not only scraped but also standardized using Natural Language Processing (NLP) before further analysis. This stage uses the SCALES collaboration to access federal dockets (PACER) and applies the pipeline described in prior work (Azaria et al., 2025). The

Figure 3 Example of an unstructured docket entry from PACER (illustrative snippet).



Defendant

Practice Data Systems Inc.

represented by **David J. Poirier**
(See above for address)
LEAD ATTORNEY
ATTORNEY TO BE NOTICED

Andrew Lloyd Platt
(See above for address)
ATTORNEY TO BE NOTICED

Defendant

John Does 1-10

Date Filed	#	Docket Text
08/08/2016	1	COMPLAINT filed by Podiatry In Motion, Inc.; Filing fee $ 400, receipt number 0752-12229663. (Attachments: # 1 Exhibit A-D)(Edelman, Daniel) (Entered: 08/08/2016)
08/08/2016	2	CIVIL Cover Sheet (Edelman, Daniel) (Entered: 08/08/2016)
08/08/2016	3	ATTORNEY Appearance for Plaintiff Podiatry In Motion, Inc. by Daniel A. Edelman (Edelman, Daniel) (Entered: 08/08/2016)
08/08/2016	4	ATTORNEY Appearance for Plaintiff Podiatry In Motion, Inc. by Cathleen M. Combs (Combs, Cathleen) (Entered: 08/08/2016)
08/08/2016	5	ATTORNEY Appearance for Plaintiff Podiatry In Motion, Inc. by James O. Latturner (Latturner, James) (Entered: 08/08/2016)
08/08/2016	6	ATTORNEY Appearance for Plaintiff Podiatry In Motion, Inc. by Dulijaza Clark (Clark, Dulijaza) (Entered: 08/08/2016)
08/08/2016	7	MOTION by Plaintiff Podiatry In Motion, Inc. to certify class (Attachments: # 1 Exhibit A, # 2 Exhibit E-G)(Edelman, Daniel) (Entered: 08/08/2016)
08/08/2016	8	MEMORANDUM by Podiatry In Motion, Inc. in support of motion to certify class 7 (Attachments: # 1 Exhibit A, # 2 Exhibit H)(Edelman, Daniel) (Entered: 08/08/2016)
08/08/2016		CASE ASSIGNED to the Honorable Robert M. Dow, Jr. Designated as Magistrate Judge the Honorable Susan E. Cox. (ew,) (Entered: 08/08/2016)
08/09/2016	9	NOTICE of Motion by Daniel A. Edelman for presentment of motion to certify class 7 before Honorable Robert M. Dow Jr. on 8/17/2016 at 09:15 AM. (Edelman, Daniel) (Entered: 08/09/2016)
08/09/2016	10	MOTION by Plaintiff Podiatry In Motion, Inc. to continue *and Enter Plaintiff's Motion for Class Certification* (Clark, Dulijaza) (Entered: 08/09/2016)
08/09/2016	11	NOTICE of Motion by Dulijaza Clark for presentment of motion to continue 10 before Honorable Robert M. Dow Jr. on 8/17/2016 at 09:15 AM. (Clark, Dulijaza) (Entered: 08/09/2016)
08/09/2016	12	Corporate Disclosure STATEMENT by Podiatry In Motion, Inc. (Clark, Dulijaza) (Entered: 08/09/2016)
08/09/2016		SUMMONS Issued as to Defendants JS & Company Incorporated, Practice Data Systems Inc. (jh,) (Entered: 08/09/2016)
08/09/2016	13	MINUTE entry before the Honorable Robert M. Dow, Jr: Plaintiff's motion to enter and continue the Plaintiff's motion for class certification 10 is granted. Initial status hearing is set for 10/6/2016 at 9:00 a.m. and parties are to report the following: (1) Possibility of settlement in the case; (2) If no possibility of settlement exists, the nature and length of discovery necessary to get the case ready for trial. Plaintiff is to advise all other parties of the courts action herein. Lead counsel is directed to appear at this status hearing. The parties are requested to file a joint status report at least two days prior to the initial status (see Judge Dow's web page at www.ilnd.uscourts.gov for content). Notice of motion date of 8/17/2016 is stricken and no appearances are necessary on that date. Mailed notice (cdh,) (Entered: 08/09/2016)
08/09/2016	14	MINUTE entry before the Honorable Robert M. Dow, Jr: The initial status hearing set for 10/6/2016 is reset to 10/5/2016 at 9:00 a.m. No appearances are necessary on 10/6/2016. Mailed notice (cdh,) (Entered: 08/09/2016)

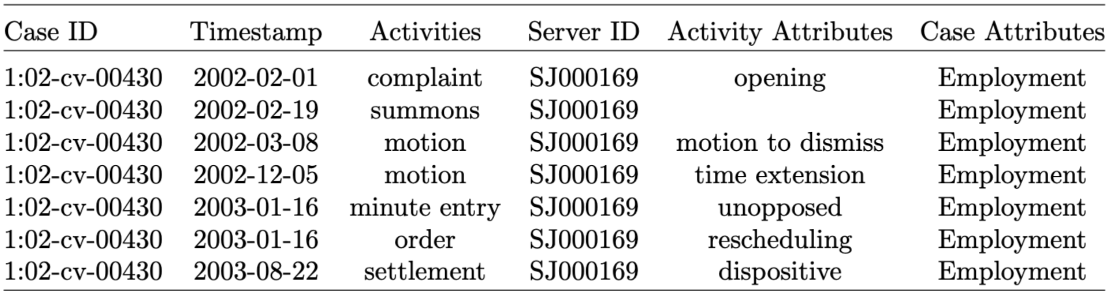
output of this stage consists of three structured datasets: *Cases* (case-level identifiers and attributes: filing/termination dates, nature-of-suit, outcomes); *Labels* (NLP+classifier categories assigned to docket entries; not yet an event log); and *Judges* (judge identities disambiguated and linked to tenure periods, from which caseloads by period are computed; Pah et al. (2021)).

These datasets provide the structured foundation for event log construction in Section 4.2. Because the pipeline is automated, it can be re-run as new dockets arrive, enabling regular updates to Cases, Labels, and Judges datasets. This approach overcomes the issue of adapting the current courts' IT systems for process mining, building on the existing data structure. This capability is an important enabler of the PTL in service systems.

4.2. Event Log Construction

The next stage is to construct an event log that combines the structured datasets into one dataset that automates the process mappings and simulation model construction we describe in the following sections.

Figure 4 Event log structure (illustrative).



Case ID	Timestamp	Activities	Server ID	Activity Attributes	Case Attributes
1:02-cv-00430	2002-02-01	complaint	SJ000169	opening	Employment
1:02-cv-00430	2002-02-19	summons	SJ000169		Employment
1:02-cv-00430	2002-03-08	motion	SJ000169	motion to dismiss	Employment
1:02-cv-00430	2002-12-05	motion	SJ000169	time extension	Employment
1:02-cv-00430	2003-01-16	minute entry	SJ000169	unopposed	Employment
1:02-cv-00430	2003-01-16	order	SJ000169	rescheduling	Employment
1:02-cv-00430	2003-08-22	settlement	SJ000169	dispositive	Employment

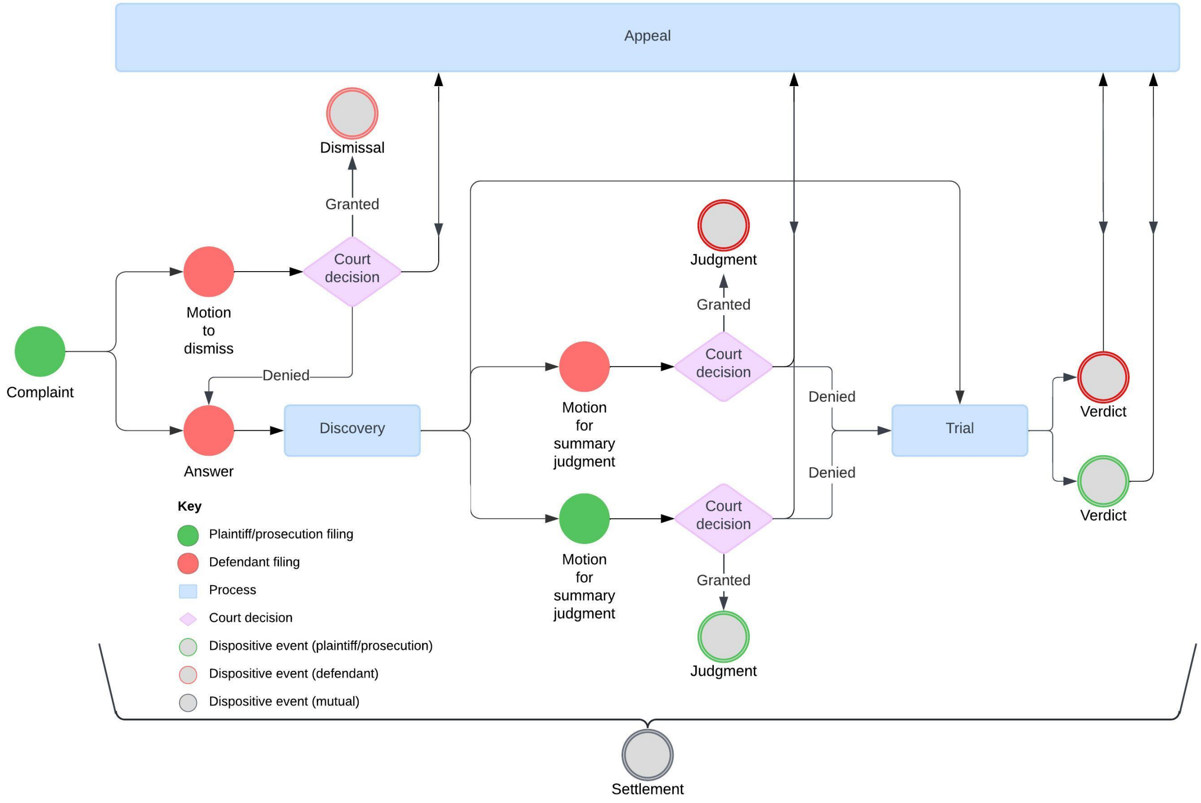
An event log is built around a few simple components, as presented in the example in Figure 4 Each event links to a case identifier, is assigned a canonical event type, and carries a timestamp. When possible, start and completion markers are preserved. A resource field connects the event to the judge responsible at that point in time, and additional attributes capture details such as the type of motion or the outcome of a hearing. Together, these elements represent both the paths of cases and their various components through the system, as well as the distribution of work across judges.

The *Cases* dataset contributes identifiers and static attributes, the *Labels* dataset provides categories assigned to docket entries, and the *Judges* dataset adds disambiguated judicial resources. Integrating these sources produces a single structure that records how cases flow through the court (Azaria et al., 2025).

The event log also incorporates two kinds of features. *Static* features are case-level attributes that remain constant as the case moves through the system (e.g., nature-of-suit, count of plaintiffs). *Dynamic* features are derived from the sequence of events and change as the case progresses (e.g., inter-event times, waiting time to motion disposition, number of active motions). Static and dynamic features together make it possible to describe what a case is and how it evolves.

Raw docket labels are too detailed for direct use as events. Visualizing them will produce a spaghetti process, where the number of paths and loops obscures structure (van der Aalst, 2011). To address this, we apply a nesting approach: labels are grouped into canonical events (e.g., plea, order) and, when appropriate, paired across time (e.g., motion filed → motion decided). These

Figure 5 Civil process highly simplified.

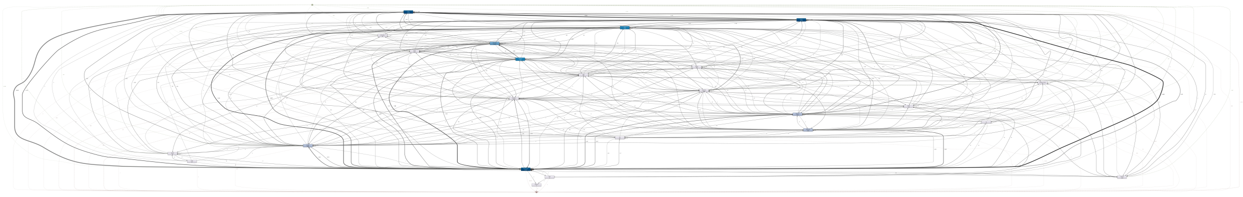


events have various dynamic attributes that can be assigned to them. Some of which are sub-types, some indicate a feature related to flow (e.g., unopposed, dispositive). Canonical events could then be grouped into broader phases (filing, pre-trial, trial). This preserves detail while making the process interpretable (Azaria et al., 2025).

4.3. Process Mapping and Dashboards

With the event log in place, the next stage is supporting descriptive analytics using process discovery and mapping. Process mining techniques allow us to discover the paths cases follow, identify common variants, and locate bottlenecks. This visibility is important: comparing the mined process map, representing performances of the process as reflected in the data, with the perception of court administrators and judges of how they perceive the system to be performing, helps identify misalignments and focus management attention where it is most needed.

Figure 6 **Civil process map (unfiltered).**



Note. For simplicity, the map presents only Civil cases of the Rockford Courthouse.

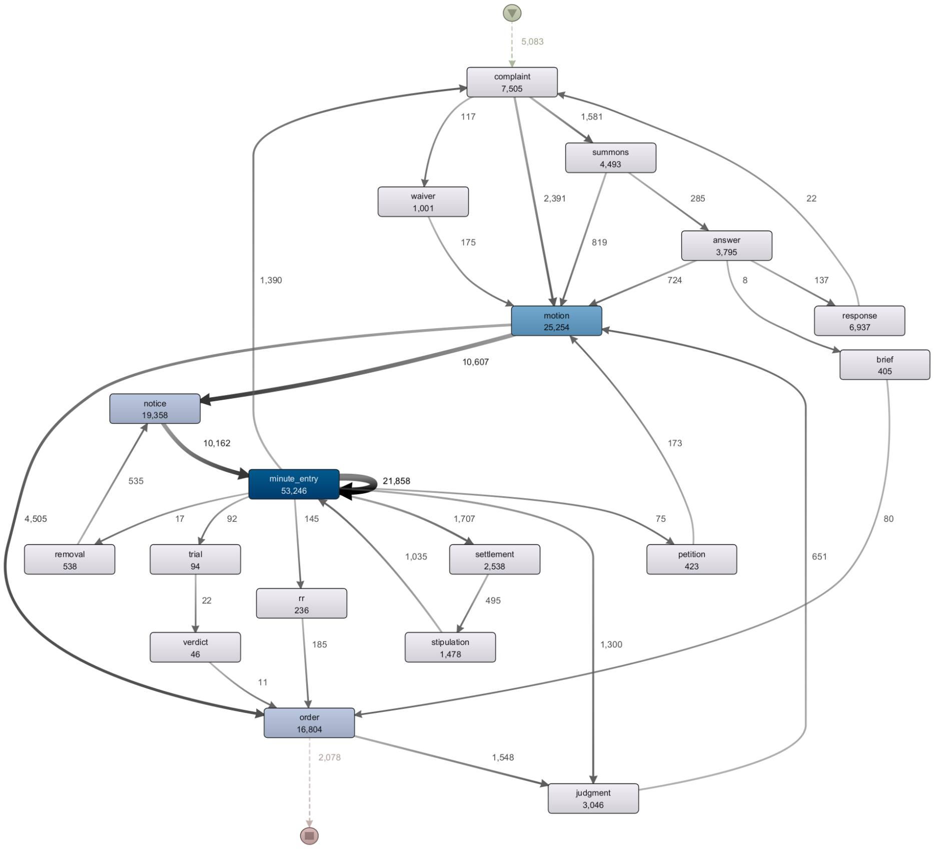
Figure 5 presents a highly simplified chart of the Civil litigation process. This chart, taken from Azaria et al. (2025) represents the process as envisioned by experts. Each case begins with a complaint, followed by the discovery of evidence, and finally trial and a verdict. The chart tracks the different decision points of the judges and the parties, which impact the content of the case and its progression. Another feature of the process is that it can end at any point in its life cycle by settlement, inside or outside of the court. Even from this highly stylized view, it is obvious that cases may take many routes through this process (each route is called a variant in the terminology of process mining). These characteristics help drive the high variance in case progression.

Process mining and mapping can bring clarity to how different case types flow, how long cases spend in each stage, and where delays accumulate. However, as illustrated below, the goal of clarity is often aspirational in practice, requiring substantial further work to be realized.

Reliable maps require care, since raw logs often contain many rare paths that can obscure the main structure. Event-log complexity metrics underscore the importance of abstraction and simplification to keep models interpretable and analytically useful (Benner-Wickner et al., 2014).

Figure 6 shows a screenshot of the portion of the NDIL process produced by a leading process mining tool Disco (Günther and Rozinat, 2012), which has some of the more advanced process mapping capabilities. It presents an unfiltered process map of only the civil cases (criminal cases are excluded) for the years of 2001-2020 handled in the Rockford Courthouse, which has the lightest case load among all courthouses of the NDIL. The figure, based on the nested events

Figure 7 Civil process map (filtered).



Note. For simplicity, the map presents only Civil cases of the Rockford Courthouse.

and carefully constructed event log described in Section 4.2, exemplifies the source of the name spaghetti process. In this dataset there are 6,334 cases and 5,115 case variants - indicating that very few cases take identical path through the process. The resulting process map is unreadable and cannot provide any insights.

To overcome this issue, Figure 7, also taken from Disco, shows a filtered version of the same dataset. In this filtered map, arrows represent transitions between activities, and the numbers printed on the arrows reflect the frequency (i.e. how many times that transition occurred). The thickness and color saturation of each arrow or node encode higher frequency flows—thicker and darker elements indicate more common paths. Because the map is filtered to exclude low-frequency paths, the counts on incoming and outgoing arcs do not always sum perfectly: some transitions have been omitted for clarity.

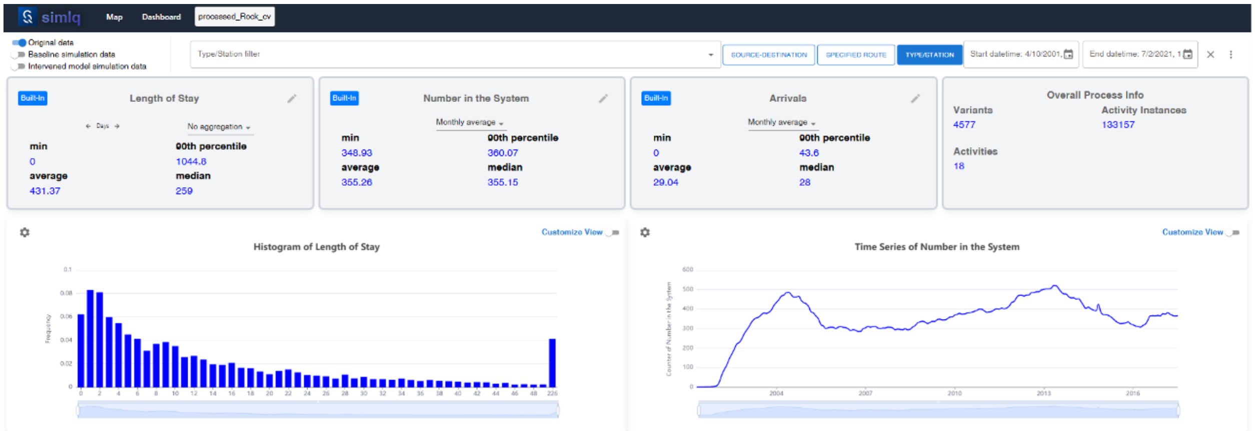
The visualization makes it easier to identify where activity loops and bottlenecks occur, for example, the high-frequency exchange between motions and minute entries, which captures repeated judicial responses and procedural updates.

While some transitions, such as those leading to judgment or settlement, represent terminal outcomes, others reflect interim dynamics that can repeat multiple times within a single case. By filtering for frequency, the model becomes more interpretable and suitable for descriptive analytics. The increased process clarity then supports simulation calibration and comparative analysis across case types or courthouses.

We note that identifying the right level of filtering (i.e., aggregation) of the event log remains a significant challenge for process mining and data-driven simulation platforms. Striking the right balance between losing sight of the key operational aspects (e.g., aggregating to the case level, which is quite common in prior court analysis literature) and using an overly detailed view (as on Figure 6) is of great importance for the resulting PTL to support real-life operations.

The second component of step 3 in the PTL is the dashboard. Dashboards based on event logs provide a systematic way to measure performance. Key performance indicators include clearance rate, time to disposition, age of pending cases, and workload distribution across judges. Traditional reporting is often annual or ad hoc, making it difficult to monitor congestion as it builds. By generating these measures directly from the event log, they can be tracked continuously. This aligns with calls in law-and-economics for systematic, data-driven evaluation of court performance (Ramos-Maqueda and Chen, 2025), and with comparative studies identifying disposition time (DT) as a widely used efficiency measure linked to IT maturity (Mathis and Mussard, 2025). Continuous monitoring enables early detection of backlogs, timely evaluation of policy changes, and ongoing assessment of congestion.

Figure 8 **Illustrative dashboard.**



Start datetime: 4/10/2001, 1
End datetime: 7/2/2021, 1

Length of Stay

min	90th percentile
0	1044.8
average	median
431.37	259

Number in the System

min	90th percentile
348.93	360.07
average	median
355.26	355.15

Arrivals

min	90th percentile
0	43.6
average	median
29.04	28

Overall Process Info

Variants	Activity Instances
4577	133157
Activities	
18	

Note. For simplicity, the Dashboard presents only Civil cases of the Rockford Courthouse.

4.4. Simulation Model Training and Validation

A pivotal step in enabling the PTL process is the predicted analytics phase, which is facilitated by the automated training and validation of the simulation model. This step involves specification and training of the model, and its validation.

The starting point of the process, after the event log is constructed, runs the event log through the process mining tool to construct the baseline scenario that represents the ground truth view of the process in the data. Figure 8 represents key performance indicators for the baseline scenario produced by the SiMLQ platform (we use this platform throughout this and the following section) applied to the NDIL data. The goal of the training and validation steps for the simulation model is to ensure that an adequate level of agreement between the simulated event logs and the baseline scenario has been achieved, i.e., that the queueing network simulation captures the causal structure in the congested system we model.

Following best practices in ML literature, the baseline data sample (i.e., the event log described in the previous section) is first split into *training* and *validation* components. Note that the event log data is a multi-dimensional time series, thus the split must consist of a vertical cut (based on some value of the Timestamp column on Figure 4), with the most recent portion of the data (20% of the total available sample seems to be a common rule of thumb) set aside as the validation sample.

Once the simulation model is trained as described below, it is applied to the validation sample and its accuracy is evaluated. We note that since the job of the simulation model is to produce complete new event log records, i.e., all columns on Figure 4, the evaluation of whether the model is sufficiently accurate is not trivial - we refer the reader to (Chapela-Campa et al., 2025) for an in-depth discussion of various evaluation procedures.

Before turning our attention to training, we briefly mention an important issue of concept drift (Lu et al., 2018), where changes in the underlying data generation process between the training and validation portions make the construction of an accurate simulation model essentially impossible. If adequate model accuracy cannot be achieved and the presence of concept drift is suspected, the size of the validation sample may have to be reduced (with the associated reduction in the potential prediction horizon of the model) to ensure that the data generation process is relatively homogeneous between the training and validation parts. If the suspected cause of the concept drift is seasonality, one must ensure that the training sample is large enough to contain a sufficient number of seasonal cycles. We note that automated detection and learning in the presence of concept drift remains an active area of research in the ML community.

We now turn our attention to model training and describe some key steps involved in this process. The underlying network represented in Figure 6 (with the simplified view on Figure 7), is conceptualized as a queueing network, with each node (activity) representing a queueing station and arcs representing transitions between stations. The entities, represented by case IDs (first column on Figure 4) arrive at the starting node (either according to historical arrival record or an arrival generator, which is part of the simulation model) and then move through the network until the final node is reached. For each station, it is necessary to fit an appropriate queueing model where customer types are assumed to be heterogeneous and service times are non-stationary. Once the case ID exists at a station, it is necessary to predict the routing - which child station will be visited next. Thus, the key steps in simulation model development are:

- **Grouping:** identifying clusters of customers (case IDs) whose service times come from the same distribution. For this step, a variety of clustering techniques (e.g., K-means) can be used.
- **Sojourn Predictions** For each station, we can specify an ML model that predicts the service time for each customer type at each point in time. Note that the set of predictors must include an adequate description of the state of the system (e.g., total number in the system, number at each station) at the time when the sojourn time prediction is made. While many ML algorithms are available to predict sojourn times, for simulation purposes, we require a prediction of the sojourn time distribution, rather than a single point estimate (otherwise simulation model will produce deterministic rather than stochastic outcomes). We find that survival prediction models are particularly suitable for this task, with Random Survival Forest RSF) (Ishwaran et al., 2011) delivering consistently accurate results.
- **Waiting Times** At each station, the waiting times can be modeled through regular discrete event queueing simulators. However, this requires the specification of the service discipline and the number of servers at the station at each point in time. The latter can be particularly challenging since the server concept may differ for different stations (is it the judge? the clerk?), as well as change over time. The approaches in e.g., Veeger et al. (2011); Etman et al. (2006) are helpful in such settings. An alternative workaround is to use a $G/G/\infty$ representation where waits are captured as part of the sojourn time predictors. Predictors of waiting times may also require the prediction of the servers' policy for picking up cases from the queue in front of them. This has been studied, e.g., in Senderovich et al. (2014a) and evidence for cherry picking in MRI settings is given in Lagzi et al. (2023).
- **Routing Prediction** This step can be conceptualized in a variety of ways, ranging from simple *empirical routing* (used together with historical arrival records) simply routes each case id according to its trace in the event log, to $k-$Markov routing, which constructs a probabilistic routing rule

based on k activities previously visited by the case id. Another option is to use more sophisticated routing predictors such as LSTM neural networks.

The end result of the training and validation step is the production of simulated event logs for the validation data set for each simulation iteration. Simulation result for the NDIL data set (20% validation, historical arrivals, empirical routing, RSF sojourn predictor) from SiMLQ is displayed in Figure 9. The simulated data is in Orange color, while the "ground truth" is in blue, as in Figure 8 above. As in any management analytics project, the test set should also be used to choose the best model; here, this is choosing among simulation models that, e.g., use other routing mechanisms. It can be seen that this simulation model is very accurate at the aggregate level (e.g., the average simulated Length of Stay is 431.24 days vs the ground truth value of 431.37). There are, of course, some discrepancies between the simulated and actual distributions of key indicators; however, the overall accuracy of the simulation model is quite high for these data.

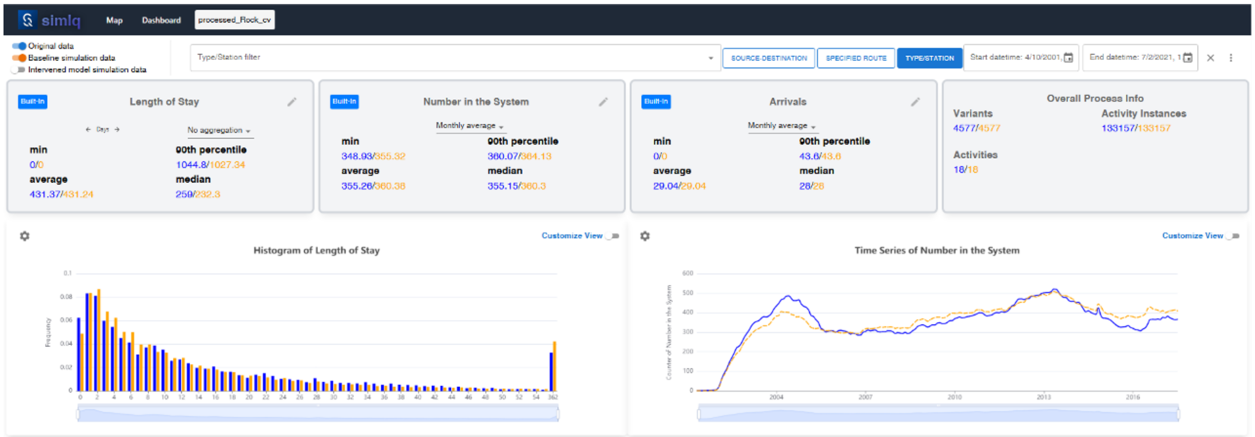
Before leaving this section, it is important to emphasize the key aspects of the simulation model described above:

- Data-driven: the model was automatically constructed from the event log data by SiMLQ with minimal intervention from the user.
- Development speed: the model was trained and validated in a few hours of computation time (caused by the event log complexity), rather than having the development time measured in months for traditional simulation models.
- Ease of updating: as new data (e.g., as a result of process interventions) becomes available, it is joined to the current event log; the model update is then straightforward, supporting the PTL process.

4.5. Counterfactual Intervention Analysis

In this section, we illustrate the key step in the PTL process: using the simulation model described in the previous section to perform comparative analytics.

Figure 9 Illustrative baseline simulation results.



Start datetime: 4/10/2001, 1
End datetime: 7/2/2021, 1

Length of Stay
← Days →
No aggregation
min 0/0
90th percentile 1044.8/1027.34
average 431.37/431.24
median 259/232.3

Number in the System
Monthly average
min 348.93/355.32
90th percentile 360.07/364.13
average 355.26/360.38
median 355.15/360.3

Arrivals
Monthly average
min 0/0
90th percentile 43.6/43.6
average 29.04/29.04
median 28/28

Overall Process Info
Variants 4577/4577
Activity Instances 133157/133157
Activities 18/18

Note. For simplicity, the Dashboard presents only Civil cases of the Rockford Courthouse.

Counterfactual analysis seeks to answer a wide variety of what-if questions with respect to various interventions (endogenous changes), as well as exogenous changes to the process: how would the key performance indicators change if an intervention X was implemented at a certain point in the past? Typical interventions include (but are not limited to):

- Changing processing capacities of certain activities. This is easiest to model as a speedup/ slowdown of the respective stations (nodes) in the process. Note that these changes can be applied universally, i.e., to all customers (case-ids) visiting an activity, or selectively to certain customer groups.
- Changing to the way certain customers (case-ids) are routed. Again, changes in routing can be applied selectively to certain customer groups.
- Changes to resource schedules or availability.
- Changes to resources’ service discipline, i.e., priorities of different customers.

Typical exogenous shocks include changes to the arrival pattern or composition of arrivals by certain characteristics (e.g., an increase in the proportion of criminal cases in the workload of the court).

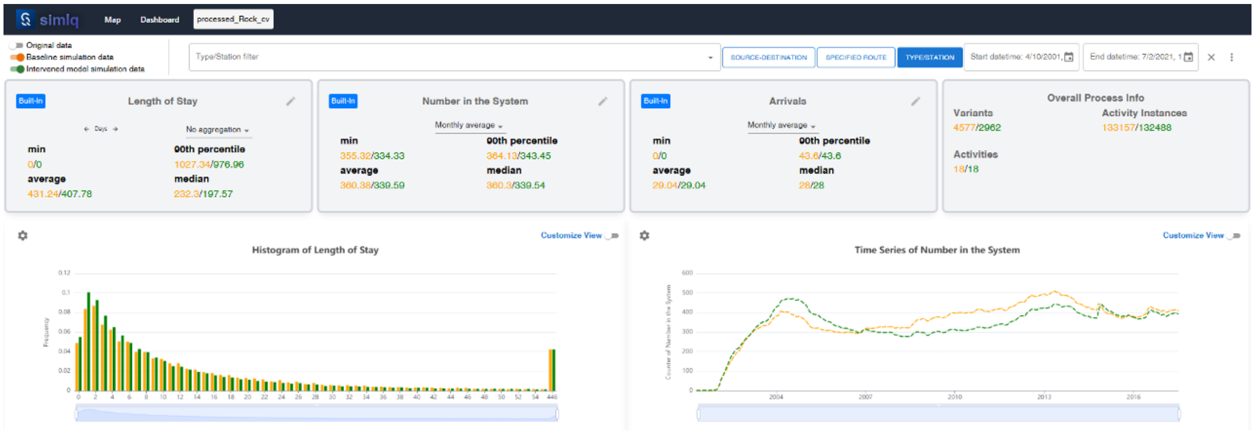
We illustrate this analysis with a relatively simple intervention: *What would be the effects of increasing the number of court days by 20%?*

We represent this intervention by speeding up all activities where the judge is actively involved by 20% (note that most activities in the process do not require court time), which can be thought of as adding another workday every week. SiMLQ makes the specification of this intervention easy: the user selects from a drop-down list of process activities and specifies the speed-up amount for each.

We then re-run the trained simulation model with these changes (specifying the appropriate number of iterations) and compare the result to our baseline simulation scenario described earlier (note that the appropriate apples-to-apples comparison is between the two simulation scenarios, not the intervened scenario vs the ground truth in the data, so as to focus on the estimated impacts of the intervention as separate from model errors). The results are presented in Figure 10, where the output in green represents the intervention scenario and the output in orange the baseline scenario. Several interesting observations emerge from this analysis:

- The reduction average Length of Stay is about 5.5% and the reduction in the average Number in the System (i.e., active cases before the court) is about 10.8%. While both are sizable improvements over the baseline scenario, they fall well short of the 20% extra capacity in this intervention. The gap is due to that only in-court activities were intervened on, while most of the case life-cycle occurs outside the court. Nevertheless, the reduction in Number in the System indicator shows that the model picks up the indirect effects of the intervention: speedier case resolution is expected to reduce average congestion in the system.
- Histogram for the Length of Stay indicates that the intervention is likely to disproportionally benefit cases that already spend very little time in the system. These cases likely require emergency hearings or judgments; the bulk of cases that do not require such rulings will see less of a benefit.
- It is also interesting to observe how the Number in the System will evolve over time. Counterintuitively, within the first few years after the intervention the Number in the System and the

Figure 10 **Illustrative results of counterfactual analysis.**



Length of Stay

min
0/0

90th percentile
1027.34/976.96

average
431.24/407.78

median
232.3/197.57

Number in the System

min
355.32/334.33

90th percentile
364.13/343.45

average
360.38/339.59

median
360.3/339.54

Arrivals

min
0/0

90th percentile
43.6/43.6

average
29.04/29.04

median
28/28

Overall Process Info

Variants
4577/2962

Activity Instances
133157/132488

Activities
18/18

Note. For simplicity, the Dashboard presents only Civil cases of the Rockford Courthouse.

associated congestion is expected to increase - this happens as various hearings are moved up; it may also reflect the fact that speedier access to trial may create less of an incentive to settle outside of court for certain case types (which to model effectively, would likely require input from a domain expert). However, by about 2007 the congestion in the intervened system is significantly lower (our counterfactual intervention is assumed to take place in 2001), and remains so until mid-2015. At the tail end of the data, the orange and green lines are very close: as the cases that derive the most benefit from the intervention are worked out of the system, the impact on the remaining cases is smaller. This illustrates how the data-driven simulation model captures complex dynamics and inter-relationships in the underlying system.

Examining these results, the court administrators may decide whether the expected benefits outweigh the costs of the intervention, as well as prepare for the changes in system congestion described above. Should the intervention be implemented, the new data should be collected as per the PTL process, the simulation model retrained (allowing the various ML components within the model to learn the actual effects of the intervention in the field, and some new counterfactual analyses may be performed.

5. Gaps and Requirements to Reach a Live Digital Twin in Courts

The case study presented above demonstrates the PTL through its stages. This reflects the intended scope of the current work: a proof of concept designed to validate the feasibility of data extraction, process mapping, and simulation modeling using existing court information systems. The study does not extend to the stages of choosing or implementing interventions, as these would require collaboration with judicial authorities and operational access to live management processes.

While courts provide a clear and socially significant illustration of the PTL, they are not necessarily the environment where its benefits are most pronounced. The PTL framework is powerful in high–clock-speed service systems—settings such as emergency departments, call centers, logistics hubs, or online service platforms. In such applications, data are generated continuously and the PTL can operate at (near) real time.

Additionally, the final arrow of Figure 2, which feeds post-intervention outcomes back into the twin, represents the transition from a static analytical model to a continuously learning digital twin. Realizing this feedback loop in judicial settings poses particular challenges. Court information systems were designed primarily for record keeping, not for analytics or operational monitoring. Many other service network systems are designed for financial record keeping rather than operational management. Therefore, most databases capture only one timestamp per activity, e.g., the day the activity was completed. To produce the most value from the PTL, a more detailed timestamp system is required - when the case entered the queue for the activity, when it started the activity, and when it completed the activity. The absence of such fine-grained information is widely recognized as a barrier to data-driven performance evaluation and to understanding how judicial workload unfolds over time (Pereira, 2025).

A further limitation concerns data granularity. Anything that occurs between successive recorded timestamps, such as walking, remains invisible to both the process-mining system and the resulting simulation model. Consequently, the PTL is best suited for timestamp-rich environments in which events are logged at fine temporal resolution. As automatic and sensor-based data collection improves, this constraint will ease, but many modern service systems still lack the temporal detail required to capture intermediate actions and waiting periods accurately.

The challenges of implementing a live digital twin in courts are not purely technical. They span technological, organizational, and governance dimensions that are deeply intertwined.

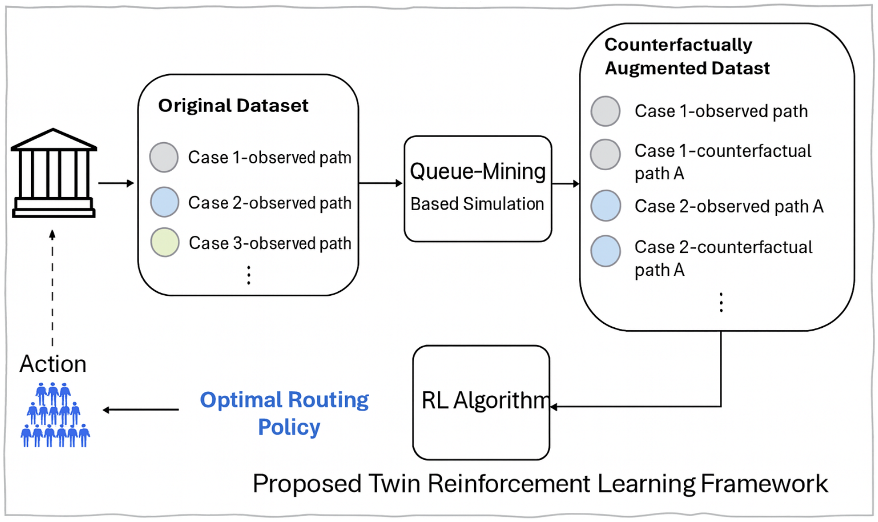
On the technological side, as many other service networks processes, most court systems rely on fragmented information infrastructures, where dockets, case management systems, and administrative records operate in silos. A live twin requires integration across these systems to enable automated data ingestion, calibration, and drift detection. Without this integration, models quickly become outdated as court practices or caseload compositions change.

Organizationally, embedding a digital twin demands shifts in professional routines and local practices. In a judicial environment, Judges, clerks, and administrators must learn to interpret and act on model outputs while maintaining judicial independence and accountability. The goal is not to replace human judgment but to extend its analytical reach, making it easier to anticipate congestion and evaluate potential reforms before they are implemented.

A conceptual challenge also lies in the degree of human involvement allowed during model construction. Current process-mining systems are fully automated such that process discovery and mapping are driven solely by event logs, with no expert input. This is the mirror image of traditional simulation modeling, where every process element was specified manually. The optimal balance may be between both extremes in integrating domain expertise to guide abstraction, validate discovered structures, and ensure that the PTL approach is interpretable and beneficial. In our experience, such a collaboration is still required to improve the causal structure of the simulation model, e.g., when dealing with a concept drift.

Governance considerations are equally central. Judicial data contain sensitive information, and any analytical framework must comply with principles of transparency, privacy, and due process.

Figure 11 **Twin Reinforcement Learning (TRL) schematic.**



Establishing standards for data sharing, model validation, and human oversight is essential for legitimacy. These governance mechanisms form the boundary conditions within which a digital twin can operate as a trusted decision-support system.

When these conditions are met, the PTL framework can evolve beyond diagnosis toward prescriptive decision support. A live twin can serve as a virtual policy laboratory in which alternative interventions are tested through simulation before being introduced in practice. In the longer term, the PTL can evolve from simulating interventions to actively learning which interventions work best. Step 6 of the framework, the evaluation of alternative policies through counterfactual experimentation, can be implemented, e.g., through reinforcement learning (RL). RL views policy design as a sequential decision problem in which an agent interacts with its environment to optimize long-term performance metrics such as congestion, workload balance, or timeliness.

The *Twin Reinforcement Learning (TRL)* framework, illustrated in Figure 11, couples the empirical foundation of a digital twin with the adaptive capability of RL to derive operational policies that are both efficient and institutionally feasible. Unlike classical optimization, which assumes fixed parameters and must be re-calibrated manually, TRL can incorporate counterfactual data and feedback from the twin, enabling safe, data-efficient experimentation without disrupting real operations.

TRL integrates advances in RL (Sutton and Barto, 1998; Dulac-Arnold et al., 2021) and queueing-network control (Dai and Gluzman, 2022; Liu et al., 2022), together with methods for constraint-aware and data-driven scheduling (Senderovich et al., 2019; Bi et al., 2024). Within TRL, the digital twin functions as the environment in which the RL agent learns, capturing the stochastic structure of arrivals, service durations, and resource interactions. Through simulated roll-outs, the agent can explore alternative routing or scheduling policies safely, combining logged event data with simulated experience to maintain causal consistency while improving data efficiency.

At first, the TRL is not an autonomous optimizer but a decision-support mechanism. Human oversight remains integral: judges and administrators can evaluate, accept, or override proposed policies. When their decisions differ from the model's recommendation, the twin records the outcome and retrains, closing the loop between learning and governance. In this way, TRL exemplifies how Step 6 of the PTL can be realized in practice—transforming the framework from a diagnostic tool into an adaptive, continuously learning management system.

By coupling reinforcement learning with a live digital twin, TRL provides a foundation for safe experimentation, policy learning, and evidence-based decision support. For courts and other service systems, it offers a path from reactive reform to proactive system design, advancing the broader goal of data-driven access to justice. After the implementation of TRL as a decision support tool, it can be transformed into an autonomous optimizer that automates specific actions. Such a transformation would complete the PTL framework, where data impacts the model that impacts decisions that impacts the data and so on.

6. Conclusion

The Process Twin Loop (PTL) demonstrates how data-driven modeling can bridge descriptive analysis and prescriptive decision support in complex service systems. By structuring information-system data into event logs, process maps, and simulation models, the PTL provides a replicable pathway for organizations to transform existing data into operational insight and policy experimentation. Applied to the court system, the PTL illustrates how a digital-twin approach can reveal congestion dynamics, enable virtual reform testing, and lay the groundwork for continuous learning.

The case study presented here ends at Step 5 of the PTL, yet it outlines the necessary components for realizing a live, adaptive system. Future work will extend these stages toward real-time data integration, policy optimization, and live feedback—advancing from a static analytical model to an evolving digital twin capable of guiding resource allocation and procedural reform.

Michael Pidd's classic definition of a model as "an external and explicit representation of part of reality as seen by the people who wish to use that model to understand, to change, to manage, and to control that part of reality in some way or other" (Pidd, 1999) remains foundational. Yet the rise of digital twins challenges this separation between the model and reality itself. In service systems, the digital twin becomes an *embedded* representation—continuously linked to the underlying process through data streams and algorithmic feedback. Moreover, as ML and artificial intelligence increasingly assume roles in analysis and control, some of the functions once reserved for "people" in Pidd's framing are now partially automated.

In this sense, digital twins redefine what it means to model: they transform models from external artifacts of reflection into active participants in organizational behavior. For service systems such as courts, this redefinition opens new possibilities for evidence-based management, enabling institutions to learn, adapt, and improve through the very data they generate.

References

Agmon-Gonnen M (2004) Judicial independence: The threat from within. *Hamishpat (Hebrew)* 18(2).

Apromore (2024) Build a digital twin with process mining. `https://apromore.com/build-a-digital-twin-model-with-process-mining`, accessed 2025-10-09.

Azaria S (2023) *Operations management in court systems: lessons learned from the delay of Justice.* Ph.D. thesis, Tel Aviv University. Coller School of Management, Available at: `https://tau.primo.exlibrisgroup.com/permalink/972TAU_INST/quev9q/alma9933587306104146`.

Azaria S, Alexander C, Baron O, Krass D (2025) Access to data for access to justice: Unpacking judicial congestion using an event log. *Available at SSRN 5254276* .

Azaria S, Ronen B, Shamir N (2023) Justice in time: A theory of constraints approach. *Journal of Operations Management* 69(7):1202–1208, Available at: `http://dx.doi.org/10.1002/joom.1234`, published.

Azaria S, Ronen B, Shamir N (2024) Alleviating court congestion: The case of the jerusalem district court. *INFORMS Journal on Applied Analytics* 54(3):267–281, Available at: `http://dx.doi.org/10.1287/inte.2023.0026`, published.

Azaria S, Shamir N (2025) The congestion vortex: Empirical evidence of the adverse effects of delay on workload in court systems. *Production and Operations Management* Available at: `https://ssrn.com/abstract=5104446`, revise and Resubmit.

Baron O (2021) Business analytics in service operations—lessons from healthcare analytics. *Naval Research Logistics* 68(1):1–17, Available at: `http://dx.doi.org/10.1002/nav.21975`.

Benner-Wickner M, Book M, Brückmann T, Gruhn V (2014) Examining case management demand using event log complexity metrics. *2014 IEEE 18th International Enterprise Distributed Object*

Computing Conference Workshops and Demonstrations, 108–115, Available at: http://dx.doi.org/10.1109/EDOCW.2014.25.

Bi J, Ma Y, Zhou J, Song W, Cao Z, Wu Y, Zhang J (2024) Learning to handle complex constraints for vehicle routing problems. *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 93479–9509, Available at: https://papers.nips.cc/paper_files/paper/2024/hash/bi2024-learning-vrp.html.

Castro R, Guccio C, Pignataro G (2015) Evaluating the efficiency of italian courts: Do judges' characteristics matter? *European Journal of Law and Economics* 39(2):431–452, Available at: http://dx.doi.org/10.1007/s10657-013-9402-8.

Celonis (2024) Process simulation documentation. https://docs.celonis.com/en/process-simulation.html, accessed 2025-10-09.

CEPEJ (2016) European judicial systems: Efficiency and quality of justice (2016 edition, 2014 data). Technical report, European Commission for the Efficiency of Justice (CEPEJ)Council of Europe, Available at: https://www.coe.int/en/web/cepej/european-judicial-systems-efficiency-and-quality-of-justice.

Chapela-Campa D, Benchekroun I, Baron O, Dumas M, Krass D, Senderovich A (2025) A framework for measuring the quality of business process simulation models. *Information Systems* 127:102447.

Church TW, Carlson A, Lee JL, Tan T (1978) *Justice Delayed: The Pace of Litigation in Urban Trial Courts* (Williamsburg, VA: National Center for State Courts), Available at: https://www.ojp.gov/ncjrs/virtual-library/abstracts/justice-delayed-pace-litigation-urban-trial-courts-0.

Coviello D, Ichino A, Persico N (2015) The inefficiency of worker time use. *Journal of the European Economic Association* 13(5):906–947, Available at: http://dx.doi.org/10.1111/jeea.12129.

Dai JG, Gluzman M (2022) Queueing network controls via deep reinforcement learning. *Stochastic Systems* 12(1):30–67, Available at: `http://dx.doi.org/10.1287/stsy.2021.0025`.

Dalton PS, Singer T (2014) The efficiency of judicial systems: Evidence from a survey of european courts. *Applied Economics Letters* 21(17):1229–1233, Available at: `http://dx.doi.org/10.1080/13504851.2014.910601`.

Davenport TH (1998) Putting the enterprise into the enterprise system. *Harvard Business Review* 76(4):121–131, Available at: `https://hbr.org/1998/07/putting-the-enterprise-into-the-enterprise-system`.

Dulac-Arnold G, Levine N, Mankowitz DJ, Li J, Paduraru C, Gowal S, Hester T (2021) Challenges of real-world reinforcement learning: Definitions, benchmarks and analysis. *Machine Learning* 110(9):2419–2468, Available at: `http://dx.doi.org/10.1007/s10994-021-05961-4`.

Dumas M, Rosa ML, Mendling J, Reijers HA (2013) *Fundamentals of Business Process Management* (Springer), Available at: `http://dx.doi.org/10.1007/978-3-642-33143-5`.

Etman LFP, Veeger CPL, Lefeber E, Adan IJBF, Rooda JE (2006) Aggregate modeling of semiconductor equipment using effective process times. Perrone LF, Wieland FP, Liu J, Lawson BG, Nicol DM, Fujimoto RM, eds., *Proceedings of the 2006 Winter Simulation Conference*, 1827–1834 (IEEE), Available at: `http://dx.doi.org/10.1109/WSC.2006.323230`.

Grieves M (2002) Digital twin: Manufacturing excellence through virtual factory replication. Presentation, University of Michigan, 2002, origin of the digital twin concept.

Grieves M (2014) Digital twin: Manufacturing excellence through virtual factory replication. Technical report, NASA, Available at: `https://ntrs.nasa.gov/citations/20150001423`, nASA report expanding the digital twin model.

Günther CW, Rozinat A (2012) Disco: Discover your processes. Lohmann N, Moser S, eds., *Proceedings of the Demonstration Track of the 10th International Conference on Business Process Management (BPM 2012)*, volume 940, 40–44 (CEUR-WS.org), Available at: `http://ceurws.org/Vol-940/`.

Haag S, Anderl R (2018) Digital twin – proof of concept. *Manufacturing Letters* 15(3):64–66, Available at: `http://dx.doi.org/10.1016/j.mfglet.2018.02.006`.

Ishwaran H, Kogalur UB, Chen X, Minn AJ (2011) Random survival forests for high-dimensional data. *Statistical Analysis and Data Mining: The ASA Data Science Journal* 4(1):115–132.

Kapelko M (2025) Evaluating input- and output-specific inefficiency in courts of justice: An empirical study of polish district courts. *International Transactions in Operational Research* 32(5):2767–2797, Available at: `http://dx.doi.org/10.1111/itor.13152`.

Lagzi S, Quiroga BF, Romero G, Howard N, Chan TCY (2023) Negative externality on service level across priority classes: Evidence from a radiology workflow platform. *Journal of Operations Management* 69(8):1257–1281, Available at: `http://dx.doi.org/10.1002/joom.1278`.

Liu B, Xie Q, Modiano E (2022) Rl-qn: A reinforcement learning framework for optimal control of queueing systems. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems* 7(1):1–35, Available at: `http://dx.doi.org/10.1145/3514247`.

Lu J, Liu A, Dong F, Gu F, Gama J, Zhang G (2018) Learning under concept drift: A review. *IEEE transactions on knowledge and data engineering* 31(12):2346–2363.

Mathis K, Mussard S (2025) Measuring court efficiency: Disposition time and it maturity in comparative perspective. *European Journal of Law and Economics* 60:145–172, Available at: `http://dx.doi.org/10.1007/s10657-025-09853-z`.

Mitsopoulos M, Pelagidis T (2010) Judicial system performance and economic development: Some tentative lessons from a cross-country comparison. *European Journal of Law and Economics* 29(1):61–79, Available at: `http://dx.doi.org/10.1007/s10657-009-9124-4`.

Negri E, Fumagalli L, Macchi M (2017) A review of the roles of digital twin in cps-based production systems. *Procedia Manufacturing* 11:939–948, Available at: `http://dx.doi.org/10.1016/j.promfg.2017.07.198`.

Pah A, Schwartz D, Sanga S, Clopton Z, DiCola P, Mersey R, Alexander C, Hammond K, Amaral L (2020) How to build a more open justice system. *Science* 369:134–136, Available at: `http://dx.doi.org/10.1126/science.aba6914`.

Pah AR, Rozolis CJ, Schwartz DL, Alexander CS, Consortium SO, et al. (2021) Preside: A judge entity recognition and disambiguation model for us district court records. *2021 IEEE International Conference on Big Data (Big Data)*, 2721–2728 (IEEE).

Pereira MA (2025) Clearance rates and disposition times: Not the whole story of judicial efficiency. *International Review of Law and Economics* 83:106283, Available at: `http://dx.doi.org/10.1016/j.irle.2025.106283`.

Pidd M (1999) Just modeling through: A rough guide to modeling. *Interfaces* 29(2):118–132.

Provine DM (1988) *Judging Credentials: Nonlawyer Judges and the Politics of Professionalism* (University of Chicago Press).

Ramos-Maqueda M, Chen DL (2025) The data revolution in justice. *World Development* 186:106834, ISSN 0305-750X, Available at: `http://dx.doi.org/10.1016/j.worlddev.2024.106834`.

Reinkemeyer L (2020) Process mining in action. *Process mining in action principles, use cases and outloook* 11(7):116–128.

Senderovich A, Booth KEC, Beck JC (2019) Learning scheduling models from event data. *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, 401–409, Available at: https://ojs.aaai.org/index.php/ICAPS/article/view/3534.

Senderovich A, Weidlich M, Gal A, Mandelbaum A (2014a) Mining resource scheduling protocols. *Business Process Management: BPM 2014*, volume 8659 of *Lecture Notes in Computer Science*, 200–216 (Cham: Springer), Available at: http://dx.doi.org/10.1007/978-3-319-10172-9_16.

Senderovich A, Weidlich M, Gal A, Mandelbaum A (2014b) Queue mining–predicting delays in service processes. *International conference on advanced information systems engineering*, 42–57 (Springer).

Senderovich A, Weidlich M, Gal A, Mandelbaum A (2015) Queue mining for delay prediction in multi-class service processes. *Information systems* 53:278–295.

Sutton RS, Barto AG (1998) *Reinforcement Learning: An Introduction* (Cambridge, MA: MIT Press).

Tacha DR (1995) Judicial independence: The need for institutionalized responsibility. *Ohio State Law Journal* 58(1):1–9, Available at: https://kb.osu.edu/handle/1811/64571.

Tao F, Zhang H, Liu A, Nee AYC (2019) Digital twin in industry: State-of-the-art. *IEEE Transactions on Industrial Informatics* 15(4):2405–2415, Available at: http://dx.doi.org/10.1109/TII.2018.2873186.

Van Der Aalst W (2012) Process mining: Overview and opportunities. *ACM Transactions on Management Information Systems (TMIS)* 3(2):1–17.

Van Der Aalst W (2016) Data science in action. *Process mining: Data science in action*, 3–23 (Springer).

van der Aalst WM (2018) Process mining and simulation: A match made in heaven! *SummerSim*, 4–1.

van der Aalst WMP (2011) Process mining: Discovering and improving spaghetti and lasagna processes. Chawla N, King I, Sperduti A, eds., *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2011)*, 13–20 (Paris, France: IEEE), Available at: `http://dx.doi.org/10.1109/CIDM.2011.6129461`.

Veeger CPL, Etman LFP, Lefeber E, Adan IJBF, van Herk J, Rooda JE (2011) Predicting cycle time distributions for integrated processing workstations: An aggregate modeling approach. Jain S, Creasey RR, Himmelspach J, White KP, Fu M, eds., *Proceedings of the 2011 Winter Simulation Conference*, 2081–2092 (IEEE), Available at: `http://dx.doi.org/10.1109/WSC.2011.6147934`.

Vitkauskas D, Dikov G (2017) *Protecting the Right to a Fair Trial Under the European Convention on Human Rights* (Council of Europe), Available at: `https://edoc.coe.int/en/a-human-rights-rule-of-law/fundamental-rights/7226-protecting-the-right-to-a-fair-trial-under-the-european-convention-c` `html`.

Voigt S (2016) Determinants of judicial efficiency: A survey. *European Journal of Law and Economics* 42(2):183–208, Available at: `http://dx.doi.org/10.1007/s10657-016-9531-6`.

Zhang H, Ma L, Sun J, Lin H, Thürer M (2019) Digital twin in services and industrial product service systems. *Procedia CIRP* 83:574–579, Available at: `http://dx.doi.org/10.1016/j.procir.2019.04.009`.