



Hybrid decision support systems for predicting train delay codes in a socio-techno-economic system

Martin Svensson, Anton Borg & Per Lingvall

To cite this article: Martin Svensson, Anton Borg & Per Lingvall (17 Mar 2026): Hybrid decision support systems for predicting train delay codes in a socio-techno-economic system, Journal of Business Analytics, DOI: [10.1080/2573234X.2026.2642030](https://doi.org/10.1080/2573234X.2026.2642030)

To link to this article: <https://doi.org/10.1080/2573234X.2026.2642030>



© 2026 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.



Published online: 17 Mar 2026.



Submit your article to this journal [↗](#)



Article views: 118



View related articles [↗](#)



View Crossmark data [↗](#)

Hybrid decision support systems for predicting train delay codes in a socio-techno-economic system

Martin Svensson^a, Anton Borg^b and Per Lingvall^c

^aDepartment of Industrial Economics and Management, Blekinge Institute of Technology, Karlskrona, Sweden; ^bDepartment of Computer Science, Blekinge Institute of Technology, Karlskrona, Sweden; ^cSwedish Transport Administration, Gothenburg, Sweden

ABSTRACT

Introduction: In complex operational environments, hybrid decision-making frameworks offer a means to integrate human expertise – characterized by contextual sensitivity, adaptability, and experiential knowledge – with the objective, standardized precision of machine-based systems.

Method: This study develops a decision-support structure by comparing supervised Machine Learning (ML) models; random forest (RF), support vector machine (SVM) and a bidirectional encoder representation from transformer-based (KB/BERT) model – using hierarchical and flat approaches against a manual classification process, involving more than 200 train delay codes across 10 days. ML models are trained on same-day delay data and evaluated against the outcomes from a multi-actor decision process.

Results: Hierarchical models outperform flat ones, achieving near-human assessors on basic level coding (Level 1 and 2), though with greater variability (mean F1-scores (50–91 per cent)), compared to manual classification (mean F1-scores (87–98 per cent)) at the most granular level (Level 3) of prediction. “Simpler” models also outperform the more complex KB/BERT.

Practical Implications: We discuss the functionality and accuracy of ML-based hybrid decision-support systems (HDSS), noting the need for trade-offs between precision and accuracy. ML models demonstrate potential to complement – not replace – human expertise, particularly with uncertainty estimation tools that mitigate classification risks and support decision-making. We conclude with implications for data representation in the design of HDSS within socio-techno-economic contexts.

ARTICLE HISTORY

Received 27 March 2025
Accepted 4 March 2026



KEYWORDS

Hybrid decision-making; classification; natural language processing; railway management; supervised learning; train delay attribution

1. Introduction

The preference and reliance on algorithms in decision-making (Logg et al., 2019) enable the computation tasks beyond human capabilities, making work processes more efficient and effective (Kulkarni et al., 2017; Van den Broek et al., 2021). Algorithm-based decision-making also offers the possibility of continuously upgrading process power and memory (Sotola, 2012) and presents an opportunity to reduce the influence of human biases (Edwards & Rodriguez, 2019; Hardin et al., 2017), thereby standardizing and providing them with an objective foundation. The increased accuracy of algorithmic judgment, relative to human judgment (Dawes et al., 1989), has also enabled a transition from descriptive to predictive analytics (Gunaratne et al., 2018; Newell & Marabelli, 2015) and to prescriptive approaches. Nowadays, Artificial Intelligence (AI) and ML-agents retrieve, organize, analyze, and classify data, enabling algorithms to automate decisions in organizations (Davenport, 2018).

However, on the one hand, the potential for AI and ML approaches to fully automate decision-making processes is contested. Previous research suggests that substantial variation in work tasks (Autor & Handel, 2013) and the fact that work adapts to computerization (Spitz-Oener, 2006) may downplay the possibilities for automating decision-making. Algorithm-based decision-making is restricted to the algorithms used for predictions (Kulkarni et al., 2017) as well as to the restrictions humans impose when collecting data for model training (Ghasemaghaei et al., 2018). Automated solutions have also been shown to scale poorly, stagnate, be difficult to evaluate, and exhibit low overall performance due to a mismatch between the context in which they are developed and in which they are deployed. Furthermore, evidence suggests a lack of logical reasoning and the potential to identify causal relations (Holmberg et al., 2020). Humans, on the other hand, have expertise, experience, and knowledge that machines currently lack, which are essential for solving real-life problems (Demartini, 2015; Kahneman,

CONTACT Martin Svensson  martin.svensson@bth.se  Department of Industrial Economics and Management, Blekinge Institute of Technology, Valhallvägen 1, 371 79 Karlskrona, Sweden

© 2026 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.

This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives License (<http://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited, and is not altered, transformed, or built upon in any way. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.

2011). Humans are capable of (re)interpreting, supplementing, (re)framing, or even substituting information (Günther et al., 2017; Shollo et al., 2015). Humans also benefit from intuitive approaches, relative to analytical approaches, in domain-specific tasks (Dane & Pratt, 2007; Dane et al., 2012) and in cooperation may aggregate heterogeneous knowledge to rich representations of decision-making problems (Dellermann et al., 2019).

Thus, the tension between machine and human capabilities has opened the arena for Human-Centred AI (HCAI) as an augmenting or controlling technology (Schmager et al., 2023). Involvement of humans span hybrid-based decision-support systems (HDSS), which emphasize general collaboration between machines and humans, to more specific approaches of “human in the loop”, emphasizing how humans oversee all or most of the stages in a decision-making sequence (Munro, 2021), but without settling on the extent to which human involvement is ideal.

The role of human involvement is particularly important when developing HDSS for socio-technical systems (Saward & Stanton, 2018) since the outcomes of ML models depend on the data they are trained on. A slight drift in model construction can lead to biased predictions (Edwards & Rodriguez, 2019). Additionally, in stage-gated decision-making processes, the available data at a certain point in time may not fully reflect the scope of the actual problem. Moreover, it may also be challenging to account for all

relevant information at a specific point in time for a decision when human cognition is intertwined with manual work within technical systems. In summary, developing HDSS requires not only changes in technical solutions, but also consideration of work processes to develop procedures for collecting data to be used for predictive modelling.

Related to considerations of the degree of human involvement, socio-technical systems and the possibility of mimicking human decision-making in a stage-gated process, we analyze data on train delays from the Swedish Transportation Administration with the purpose of developing a hybrid-based decision-support system for classification decisions of train delays. We use data generated by train dispatchers (TKL) to provide provisional delay codes for the 10-day multi-stage and multi-actor process (see the current process in orange and our suggested process in blue in Figure 1). Free-text data, generated by train dispatchers on the first day (Day 0) of the process, is used to predict the outcome of the full 10-day process. The model predictions are made on Day 0 with the intent to make objective decisions, free resources among TKL and other stakeholders to be used for additional information gathering, identification of complicated cases or error detection, as well as to make the socio-technical process economically efficient.

The Swedish Transport Administration has a history of delay attribution coding even before it became a mandatory EU directive in 2012. Delay attribution codes are used simultaneously to assess

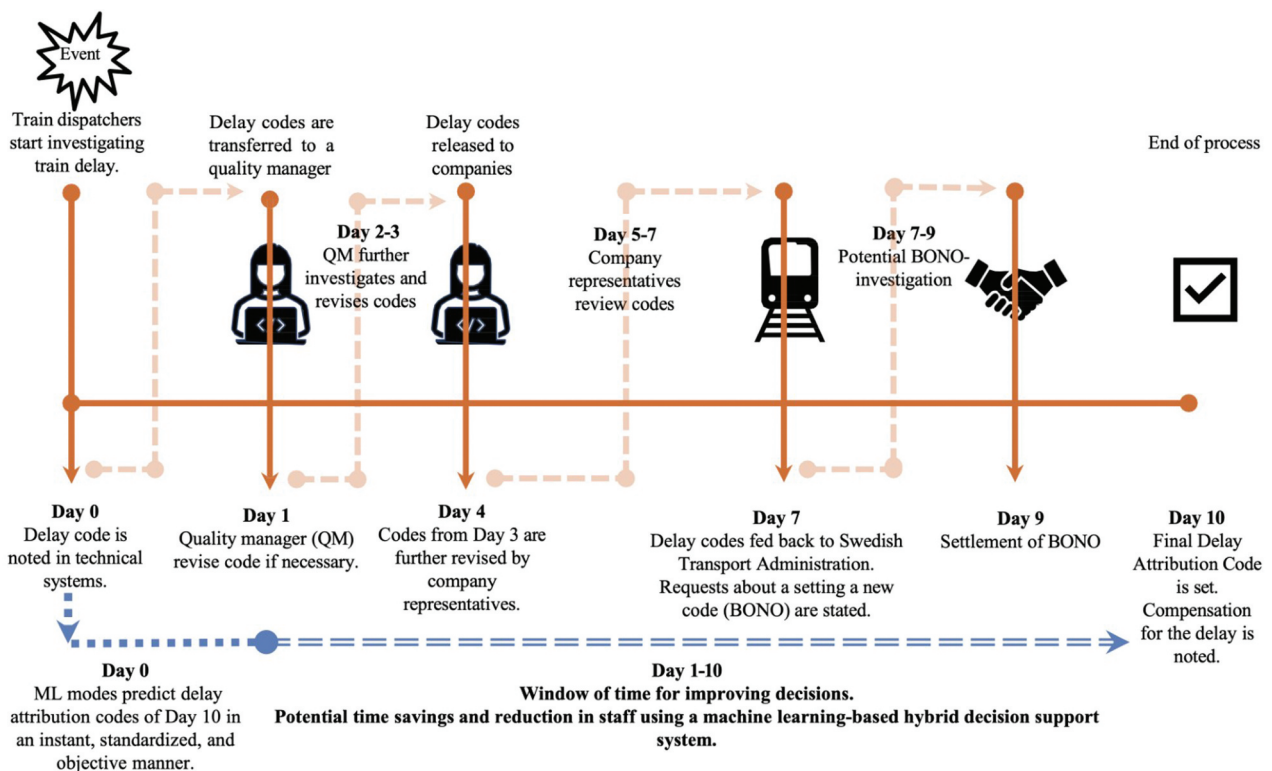


Figure 1. The decision making process of delay attribution coding.

the quality of the Swedish railway system and to identify the responsible party that caused the delay. Previous audits have identified issues with the quality of the coding process, and early research has revealed inconsistencies in the manual coding procedure (Nyström, 2008). Thus, a stretched technological frontier, including real-time data-driven decision-making and ML models offers new opportunities for dynamic, standardized, and objective HDSS that may simultaneously help address previous problems in the delay attribution process as well as contribute to the literature on HCAI in HDSS. In this study, the following research question guides our work: *To what extent can machine learning-based methods (flat and hierarchical multi-class text classification) predict delay attribution codes?*

Three key contributions are presented in the paper: first, a practical hybrid decision-support architecture is introduced, which combines dispatcher expertise with machine-learning classifiers (RF, SVM, KB-BERT) to provide provisional delay-attribution codes, accelerate the 10-day workflow, and flag uncertain cases. Second, an empirical evaluation of Swedish railway data demonstrates that hierarchical text classification outperforms flat approaches. Third, it shows that simple, feature-based models (RF, SVM) can outperform a sophisticated BERT-based model in this domain, underscoring that for highly structured, domain-specific vocabularies with limited data, lightweight models may be more effective than large pre-trained language models.

The study is outlined as follows: related work spans decision challenges in the delay attribution context, automation and HDSS in railway management and ML approaches used for text classification. This section is followed by the methodology, which includes a description of the data and pre-processing, the algorithms and evaluation metrics used, as well as the experimental setup and statistical evaluation. We present the results of the experiments and statistical analysis. Finally, the results are discussed in terms of how the analytics of assigning delay codes may be (semi-) automated and how this affects operational performance. We end by discussing limitations and future research before concluding the study.

2. Related work

2.1. The decision-making context of delay attribution coding

Since 2012, there has been an EU-directive that stipulates a systematic follow-up of train delays (Trafikverket, 2020, 2023). In Sweden, the follow-up is managed by the Swedish Transport Administration (Trafikverket, 2023), which registers and assigns delay attribution codes to train delays (Joborn & Ranjbar,

2022). First, the party that causes a delay pays a fee (quality charge) to the other party (Trafikverket, 2020). The second part is called the “right of recourse” and was introduced in 2018 as a complement to quality fees to reduce operational management and infrastructure-related train delays. The “right of recourse” means that railway companies can request compensation for their own and their customers’ additional costs if a delay is caused by an infrastructure manager. As an example of costs, the administrative expenses of the quality fee system increased from 2.5 million Swedish Krona (approximately \$200,000) in 2012 to 9.6 million Swedish Krona in 2019 (approximately \$900,000), corresponding to a 285 per cent increase in 2019 fixed prices. The Swedish Transport Administration’s quality fee payments for delays increased from 21 million SEK (approximately \$2 million) in 2012 to 192 million SEK (approximately \$19 million) in 2019, representing an 801 per cent increase in 2019 fixed prices. In 2023, the total costs associated with quality fees were 412 million Swedish Krona (approximately \$40 million), divided between the Swedish Transport Administration (273 million SEK, approximately \$26 million) and the railway operators (139 million SEK, approximately \$13 million).

The process for determining the delay attribution code involves multiple actors and spans 10 days, with activities distributed across various information systems, as shown in Figure 1. Deciding the delay attribution code is difficult for at least three reasons. First, approximately 200 delay attribution codes exist, organized in a three-level hierarchical structure (see Section Data and Data Preprocessing). Second, when dispatchers make initial decisions, they often act on incomplete data. The technical systems provide information, but overall observability is reduced (Brehmer, 1992) since dispatchers are geographically separated from the actual events on the tracks. Third, the data is manually collected and documented, containing domain-specific syntax, and the event descriptions are restricted to how the train dispatchers perceive the event.

The train dispatcher must set the code to at least the second level in the code hierarchy (Joborn & Ranjbar, 2022). If they have sufficient information and have the time, they also set it to the third level. They also need to conduct the necessary investigations to identify the cause of the delay, e.g., by contacting the train driver, consulting with maintenance contractors, or searching for additional information in relevant systems. In addition to assigning a code to the delay, the train dispatcher provides a brief description of the event. According to internal guidelines, the description should be concise, short, and neutral. When describing the event, the dispatcher has limited time to conduct the investigation, since other tasks

take precedence, which adds pressure to an already intensive workload.

The working environment is rich in interruptions, which negatively influence the work of a complex cognitive nature (Speier et al., 2003). The decision-making process also includes psychological aspects that underpin judgements (such as, e.g., attention, stress tolerance, cognitive overload (Kata & Poleszak, 2021) or even cognitive biases (Edwards & Rodriguez, 2019)), which in turn influences the selection of information used to describe events (Endsley, 1995).

The train dispatchers' code is then reviewed by the quality managers over the following three days (see QM in Figure 1). Short event descriptions are used to assess the codes and serve as input to search for additional information if the codes are unsatisfactorily decided. When experts have reviewed, revised, or confirmed the existing code, it is released to the companies operating the trains. If the train company has objections or additional information about the case, they report back before the end of the eighth day following the event. The final delay attribution code, and its corresponding quality fee, is then set in the system by the end of the 9th day following the event (see Figure 1).

Consequently, registration of train delays is a manual process that requires dispatchers to gather and compile initial information about a case and manually assign a code for the delay. This can be a time-consuming and error-prone process (Mullenbach et al., 2018), as well as a personalized process that varies depending on the dispatcher's experience. Furthermore, there is a large decision space (Simon, 1990, 2000), in which information can be either lacking or highly context-sensitive or ambiguous, making it more difficult for dispatchers to navigate. The design of the work process and the type of problem determine whether classification of train delays is automated or supported (Mullenbach et al., 2018). Decision support may significantly reduce the workload of several stakeholders. Using a machine learning-based decision-support system to classify the delay code would also provide a secondary input for dispatchers, potentially reducing classification errors. In addition to outlining the specific decision challenges in the context of the delay attribution coding, we also review literature on train delays.

2.2. Punctuality and delays in railway Transportation

Punctuality and delays have become measures of the operations' reliability and performance (Veiseth et al., 2007) and, therefore, critical elements to manage (Schöbel, 2009). However, the terminology, analysis, and countermeasures of delays vary across studies. Early on, Gylee (1994) stated that primary delays

have the greatest direct impact and secondary delays propagate from primary ones, creating a "cascading effect" (Dingler et al., 2010). Higgins et al. (1995) suggested a three-way categorization: track-related delays (e.g., infrastructure issues causing slowdowns or stops), train-dependent delays (e.g., locomotive failures), and terminal/scheduled-stop delays (e.g., loading/unloading, connections, fueling, crew). Müller-Hannemann and Schnee (2009) categorized delays by motivation, including operational disruptions, accidents, equipment malfunctions, construction and repair work, and extreme weather (snow/ice, floods, landslides); they emphasize the importance of real-time information systems for both passengers and operators to adapt during disturbances.

Besides matters of definition, research on punctuality and delays also addresses causality (variables that influence outcomes) of and the potential to predict train delays (see Arshad & Ahmed, 2019; Joborn & Ranjbar, 2022; Laifa et al., 2021; Liu et al., 2023; Spaninger et al., 2022; Tiong et al., 2023; Zhou et al., 2021). The possibility of finding causal patterns and making predictions varies between and across studies, due to differences in terminology and the setup of the operational environment. For instance, Grechi and Maggi (2018) reviewed delay categorizations and found external factors such as weather, accidents, and operational disturbances are among the most significant causes of delays. From a Swedish context, Palmqvist et al. (2017) analyzed 32 million Swedish train movements and found that timetable margins, traffic volume, weather conditions, infrastructure elements, and individual rolling stock usage explained significant variance in punctuality. Another recent Swedish study, identified snow on track as the most critical incident factor, resulting in the highest normalized delay minutes per train and the most significant increase in the odds of delay for individual trains (Mukunzi & Palmqvist, 2024). Furthermore, Johansson et al. (2026) showed that primary delays (entry, running, dwell) accounted for more than 40 percent of total delays – underscoring the need to reducing them to raise punctuality in Swedish metropolitan areas.

Moreover, a study using data of 48,000 freight trains within EU states that 40 per cent of departures, 30 per cent of runtimes, and 20 per cent of dwell times are delayed. Early trains are common: 80 percent are ready to depart early, and 60 percent do so, while 40 percent of runtimes and 75 percent of dwell times are shorter than scheduled (Palmqvist et al., 2022). Thus, causal analysis and the prediction of delays are complex matter since there is variation in causes, consequences of delays, and mitigation strategies for causes and consequences (König, 2020). Furthermore, the information describing delays may be incomplete at the time of decision-making, which in turn makes

instant classification of the delay type complex (König, 2020).

In summary, punctuality and delay studies are important facets when studying railway management. Most studies focus on system-level metrics and causal analysis – at the operational and physical layers – using regression, survival models, or simulation-based methods. Still, there are also several studies addressing types of delays and their consequences. In contrast to existing studies that address train delays from an operational side, our study targets how to improve the internal information process of delay attribution coding. By introducing ML-based hierarchical classification, we enhance internal workflows rather than understanding causes of or predicting train delays.

2.3. Automation and hybrid decision-support systems (HDSS) in railway management

Although automation historically focused on physical or mechanical tasks (Duncheon, 2002; Groover, 2016; Marsh & Mannari, 1981), it has progressed to involve information and control automation (Parasuraman et al., 2000; Sheridan et al., 2002) and information/cognitive automation (Frohm, 2008). However, in railway research, automation most often concerns the operation of trains,¹ rather than back-office processes (Brandenburger & Naumann, 2018).

Automation of railway operation also provides standardized information, which further enables automation of more peripheral work processes. However, sparse research addresses how other automation processes may improve performance beyond the direct operation of trains. Thus, research on automation processes in the railway industry must be conducted in tandem with decision-making studies to develop a better understanding of the human-technology interface (Kecklund et al., 2011; Naghiyev, 2017).

Decision-Support Systems (DSS) are computer-based systems designed to support decision-makers in solving semi-structured or unstructured problems by integrating data, models, and user-friendly interfaces (Power, 2002; Shim et al., 2002). Since their emergence in the 1960s, DSS has evolved from rule-based systems to more sophisticated platforms incorporating simulation, optimization, and AI (Arnott & Pervan, 2005). Traditional DSS often overemphasizes machine capabilities while underutilizing human cognitive strengths, resulting in systems that lack contextual awareness, adaptability, and user trust, thereby neglecting the social and organizational dimensions of decision-making (Marakas, 2003; Silver, 1991).

Recent literature highlights the integration of ML into DSS. ML enables real-time data analysis, predictive modelling, and scenario simulation, allowing for data-driven decision-making. However, these systems

are not without limitations. Decision systems supported by ML depend on the quality of data for training and testing, posing challenges related to data quality, including accuracy, completeness, consistency, timeliness, validity, and uniqueness (Askham et al., 2013).

Systems using ML for predictive analytics often operate as “black boxes”, limiting transparency and user trust (Kostopoulos et al., 2024). The trade-off between accuracy and explainability remains a central challenge (Doshi-Velez & Kim, 2017). HDSS represents a paradigm shift toward systems that synergistically integrate human and machine intelligence. Cao (2023) defines HDSS as socio-technical systems that combine the complementary strengths of human intuition and machine computation to support complex decision-making.

Such an integrated approach combines algorithmic and human decision-making, distinguishing hybrid systems from fully automated, human-out-of-the-loop systems (HOOTL). Hybrid decision-making spans a spectrum of systems, encompassing human agents to retain decision-making autonomy while relying on algorithmic or automated information gathering, to systems where humans serve as rubber-stamping mechanisms with nominal control or responsibility for decisions, a concept termed “quasi-automation” (Wagner, 2019). In other words, a hybrid decision-making mode includes human oversight (employing such as humans-in-the-loop (Munro, 2021) (HITL), humans-on-the-loop (HOTL), or humans-in-command (HIC)). Further examples of this are seen in a taxonomy of HDSS, in terms of the degree of human-AI interaction patterns: 1) human oversight, 2) Socratic AI (AI defers to humans when uncertain), and 3) interactive collaboration (Punzi et al., 2023). They argue that true synergy requires bidirectional communication, where both agents adapt and learn from each other. Without trust calibration between humans and AI, both over-reliance and under-reliance on AI can degrade decision quality (Kares et al., 2023). Thus, the DSS must be context-sensitive and aligned with human cognitive models (Rezaeian et al., 2025).

2.4. Machine learning approaches to text classification

Machine Learning-based text classification has been successfully applied in various domains to categorize documents based on their content (Kowsari et al., 2019; Sebastiani, 2002). Early contributions laid out the foundation for traditional techniques such as Term Frequency-Inverse Document Frequency (TF-IDF) and Support Vector Machines (SVM) in text classification tasks (Joachims, 1998; Russell & Norvig, 2010). Recently, Deep Learning techniques (e.g., BERT) have significantly impacted this domain,

showcasing the efficacy of Deep Learning approaches for text classification (Borg et al., 2021; Kim, 2014; Remmer et al., 2021). Automated text classification has been shown to work in specialized domains, such as medical text classification (Catling et al., 2018; Denec et al., 2024; Evans et al., 2020; Mullenbach et al., 2018; Remmer et al., 2021), and email classification (Borg et al., 2021). Such domains can have unstructured text that differs from other documents, e.g., emails have been argued to be a separate type of document (Baron, 1998). For tasks that involve many potential classes, hierarchical approaches (Kowsari et al., 2019; Perotte et al., 2014; Sebastiani, 2002) improve the accuracy and efficiency of text classification models. Hierarchical approaches streamline the classification process by breaking it down into manageable steps, making it more efficient, especially when dealing with many potential categories. Similarly, dividing data into related blocks has been shown to improve text classification performance in some cases, e.g., using language-specific BERT-based models (Remmer et al., 2021).

Although research predicting delays in rail traffic and airlines appears to be an emerging area of study, there seems to be a lack of research on classifying the different types or reasons for delays. The closest study we have found focuses on using text data for delay predictions in Chinese high-speed trains (Liu et al., 2023). Classifying different types of delays and, by extension, the reasons for the delay, is an essential step towards mitigating delays as well as developing an economically sound routine to manage them using texts from delay reports, together with the existing hierarchical delay attribution codes, holds promise for semi-automating the delay classification process. This is supported by the findings from ICD classifications (Perotte et al., 2014). Similar to emails (Borg et al., 2021) and medical documents (Mullenbach et al., 2018; Remmer et al., 2021), the unstructured text of the delay reports can be considered a distinct type of documentation. They are, at least in this case, written using domain-specific terminology and syntax. However, decision processes in socio-technical systems, such as those involved in delay attribution coding, are fraught with difficulties that may undermine the possibility of full automation. In the case of delay attribution coding, human judges interpret the circumstances of railway operation and select information to be included in the delay reports. Moreover, they can correct the process within 10 days. Thus, a semi-automated or hybrid decision-making mode seems to be an alternative.

2.5. Summary

Developing DSS for the delay attribution process may simultaneously increase both operational and

economic efficiency. However, decision processes in socio-technical systems, such as those involved in delay attribution coding, are fraught with difficulties. The decision-making process for delay attribution coding is complex, time-consuming, and error-prone as human judgment introduces variability into the technical systems. The process spans 10 days, involves several actors, and encompasses ambiguous and uncertain information, often lacking clarity due to the reduced observability of the tracks, necessitating the postponement of immediate decision-making.

Given that the decision space spans multiple levels and delay attribution codes within each level, we model this problem as a hierarchical multi-class text-classification problem. Hierarchical solutions have been suggested to perform better than flat approaches in other domains (Perotte et al., 2014; Sebastiani, 2002). The problem is investigated through a systematic classification process across multiple levels, where models are trained using data sampled based on the presence of the parent code. This hierarchical approach enables classification of instances into increasingly specific categories, resulting in a more nuanced and accurate classification process.

Given the unstructured text in the event descriptions for each delay event, this study aims to investigate the feasibility of classifying delay attribution codes to at least the second level, to provide decision support to the involved actors in the Swedish railway industry. The investigation is limited to data available in the unstructured text of the reports and, as such, does not involve additional information that dispatchers have available but have chosen not to include in the reports.

3. Method

This section describes the data collection and preprocessing, the algorithms, evaluation metrics, experiment setup, and statistical tests used.

3.1. Data and data preprocessing

The data used in this study were provided by the Swedish Transport Administration and include internal unstructured text describing the cause of the delay and the code for the delay. The features of the data are shown in Table 1.

As previously stated, there are ≈ 200 delay attribution codes organized in a three-level hierarchy, with five top-level codes, multiple refinement codes on the second level for the first-level code, and additional refinement codes on the third level for the level 2 code (Joborn & Ranjbar, 2022).

- **Operational Management (D):** disruptions caused by, for example, the Swedish Transport

Table 1. Description of data in datasets 1 and 2. Please note that the label for day 0 might differ from day 10.

Feature	Description
<i>eventcode</i>	Event identification, e.g., 439,394
<i>text</i>	Free-text describing the event and consequences
<i>label</i>	delay attribution code for day 0, e.g., DPR 03
<i>n1₀</i>	Day 0 first level code, e.g., D
<i>n2₀</i>	Day 0 second level code, e.g., PR
<i>n3₀</i>	Day 0 third level code, e.g., 03
<i>n1₁₀</i>	Day 10 first level code, e.g., J
<i>n2₁₀</i>	Day 10 second level code, e.g., PR
<i>n3₁₀</i>	Day 10 third level code, e.g., 05

Administration's own prioritization, incorrect handling, or incorrect traffic information.

- **Consequential cause (F):** disturbances caused by, for example, an expected connection, round-trip times or lack of available track.
- **Infrastructure (I):** damage to signaling and electrical installations, track and track switches, as well as culverts, tunnels, and bridges. This also includes disturbances caused by track work and weather phenomena such as solar curves.
- **Railway company (J):** disruptions caused by, for example, locomotive and machine faults, terminal and platform management, or vehicle or staff shortages.
- **Accidents/incidents and external factors (O):** disruptions caused by, for example, weather, unauthorized persons on tracks, accidents, sabotage, and trains that arrive late to Sweden from other countries.

Thus, the reason for a delay spans from the trains' braking system, lack of staff operating the trains, to environmental circumstances, such as storms, snow, or rain. After deciding on the level one delay attribution code (D, F, I, J, or O), additional letters may be added for a level two code (e.g., JTP or FAT), and numbers for a level three code, further specifying the code. e.g., for the delay attribution code JTP13, J stands for Railway company, TP stands for Terminal/Platform, and 13 is a code for a request. In this case, the delay attribution code refers to an incident on the terminal or platform, causing the railway company to request a delay. The total number of codes the dispatcher may choose from exceeds 200 codes. Given the total number of delay attribution codes, they have not been translated.

The data were collected nationwide from Sweden and consist of events that impacted trains during these days. This dataset shares similarities with a prior dataset (Joborn & Ranjbar, 2022), with a key distinction being the inclusion of data points for each day over the 10-day process. This inclusion enhances the dataset's capacity to identify variations and changes in the collected data over time, particularly concerning the delay attribution code and unstructured text. This study primarily focuses on day 0 and day 10. The

collected dataset contains 34,901 observations and 182 delay attribution codes. Figure 2 shows the number of observations for delay codes.

Each row contains an event, with the delay attribution code, and unstructured free-text explaining what happened and its consequences. The delay attribution code is described in both condensed and verbose forms (feature $n1$, $n2$, $n3$). In the latter, feature $n1$ denotes level one of the delay attribution code and corresponds to the first letter of the condensed form, feature $n2$ denotes level two of the delay attribution code and corresponds to the next two letters of the condensed form, and feature $n3$ denotes level three of the delay attribution code and corresponds to the last characters of the condensed form. The delay attribution code registered on the same day as the event and on day 10 after the event is available in the dataset, as per the process used.

The dataset was pre-processed by removing duplicate entries, those with no free-text, and only keeping rows where the label occurred at least 100 times. In this dataset, removing classes with fewer datapoints than the threshold reduced the initial dataset by 3%. In comparison, only 65% of all datapoints had free-text descriptions. Having a minimum size threshold was done to have adequate data to train, calibrate, and evaluate each class (e.g., 66 fault codes had fewer than 10 observations on the third level) (Flach, 2012). However, it should be noted that the filtering step introduces biases toward more frequent categories. This is particularly pronounced on level 3, where the number of delay attribution codes is highest. Rare codes are excluded, thereby simplifying the class space. The resulting dataset may consequently under-represent the heterogeneity of real-world delay scenarios and limit the applicability of the trained system to cases involving uncommon codes. A larger dataset would alleviate some of these issues by providing more examples for the rare categories; however, even with expanded data, some categories would remain rare due to their infrequent occurrence. It is likely that the classification problem will continue to exhibit class imbalance, potentially leading to bias in favor of the common categories and limiting the model's ability to generalize rare delay scenarios.

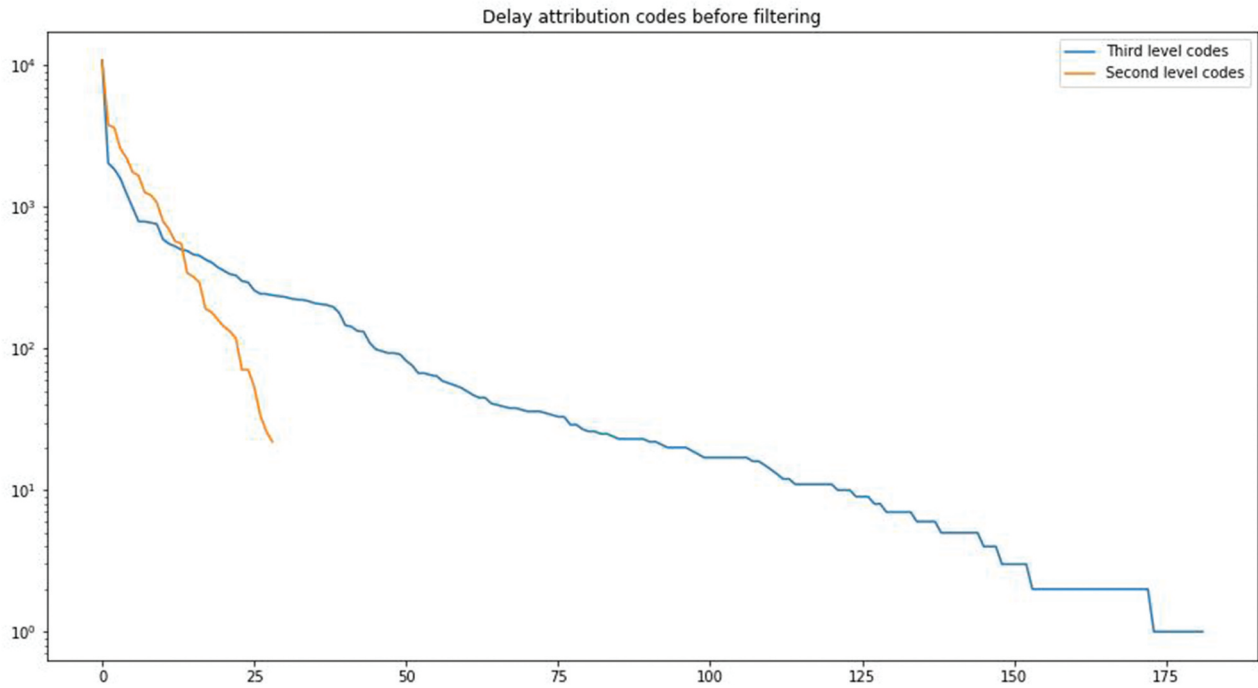


Figure 2. Number of observations for delay attribution codes before preprocessing. 45 delay attribution codes have more than 100 observations.

Filtering reduced the dataset from its original 34,901 instances to 21,484 instances and 43 delay attribution codes on level 3, with most of the reduction was due to a lack of free-text (i.e., had missing data).

Furthermore, punctuation and line breaks were removed from the free-text, and the text was transformed to lowercase. The term *sth* is used throughout the texts but written differently; *sth <speed>*, *sth <speed>*, *sth <speed>km*, or similar variations. To address this inconsistency, we remove the spacing variations, ensuring a standardized format like *sth <speed>* where the numbers are connected to the *sth* keyword. Individual train identifiers occur frequently in the free-text, and in some cases, only train identifiers are entered. These were replaced with a placeholder, *TRAINNR*. There are 8032 instances where the text consists only of train identifiers.

The free-text was tokenized during the experiment using TF-IDF. TF-IDF is a simple, yet useful way of transforming free-text into a structured format useful for machine learning algorithms (Russell & Norvig, 2010). It has been shown to enable high-performance text classification (Sebastiani, 2002). While newer natural language processing methods use transformers or deep learning to capture context and understanding better, TF-IDF is a reasonable approach for initial evaluations (Russell & Norvig, 2010). In the experiment, stop-words were removed, the texts were transformed into 1-, 2-, and 3-grams based on the words in the texts, and the top 1000 features ordered by term frequency across the corpus were retained. Initial studies were conducted with more features kept, but these yielded worse results, and therefore, the 1000-feature

threshold was kept. An example of the preprocessing would be the following, fictional delay report: “An early freight train got stuck on the track, the driver could diagnose and solve the problem themselves but 123,456 was delayed”. This is transformed into the following cleaned text: “an early freight train got stuck on the track, the driver could diagnose and solve the problem themselves, but TRAINNR was delayed”. For the specific report, the TF-IDF transformation returns a vector of 1000 feature values, where positive values represent features present, e.g., [0.0 0.0 0.218780 ...].

3.2. Algorithms

The following section presents the machine-learning algorithms used to classify delay-attribution codes from delay reports. It describes the core algorithms evaluated: Random Forests, Support Vector Machines, and a lightweight few-shot classifier built on the Swedish-specific KB/BERT language model, as well as a Uniform baseline that serves as a reference point. To address the inherent class imbalance and to provide calibrated uncertainty estimates, we integrate Conformal Prediction into each model’s decision process. Implementation details, including library choices, parameter settings, and data handling procedures, are also outlined.

A popular algorithm is Random Forest (RF) (Cutler et al., 2007). RF is an ensemble learning method that constructs multiple decision trees during training and outputs the class that is the majority decision of the individual trees (Flach, 2012). Each tree in the

ensemble is built from a random sample of the training data using a randomly selected subset of features, and the final prediction is determined by aggregating the individual tree predictions. This approach helps mitigate over-fitting and improves the model's generalizability, making Random Forest a versatile and widely used algorithm for various machine learning tasks, including classification (Cutler et al., 2007). The ability of Random Forest to handle high-dimensional data and effectively handle outliers and missing values contributes to its popularity in different domains (Kowsari et al., 2019).

Support Vector Machines (SVM) have been a popular and well-performing algorithm when it comes to text classification (Flach, 2012). SVM is a powerful supervised learning algorithm used for classification and regression tasks. It can handle complex data distributions, making it a versatile and effective tool for various machine learning tasks, including text categorization (Flach, 2012; Kowsari et al., 2019; Sebastiani, 2002). In the context of classification, SVM aims to find the optimal hyperplane that separates different classes by maximizing the margin, i.e., the distance between the hyperplane and the nearest data points of each class.

The KB/BERT Few-Shot algorithm is a lightweight, transfer-learning classifier that leverages a pre-trained KB/BERT language model (Malmsten et al., 2020) to obtain dense sentence embeddings, and then trains a tiny supervised head (in this case, an RBF-kernel SVM) on the labelled training set. Few-Shot Learning is based on the premise that prior knowledge can complement supervised information when learning the classification task (Wang et al., 2020). In this case, prior knowledge is based on language knowledge in the KB/BERT language model developed specifically for Swedish (Malmsten et al., 2020), a model that has been used in similar tasks (Remmer et al., 2021). In practice, this means that texts are encoded in mini-batches (batch size of 32) through the KB/BERT model, producing vectors that capture contextual knowledge from the underlying model. KB/BERT creates the embeddings on the cleaned data, and for the previous fictional example resulting in a vector of 768 values $[-0.53826773 \ 0.32077646 \ -0.1275206 \ \dots \ 0.12960844]$. After that, a supervised head is fitted on the embeddings of the training set, yielding a decision function. Because the bulk of the representation power resides in the frozen KB/BERT encoder, the method should achieve strong few-shot performance while requiring only modest computational resources when training the supervised head. It should be noted that: (1) the public KB/BERT model was not fine-tuned on the domain-specific data, which may restrict its adaptation to the taxonomy; (2) possible domain-specific vocabulary might not translate to the prior knowledge

captured in the KB/BERT model, potentially limiting the encoder's generalization

As a baseline model, a Uniform Classifier is used. It randomly predicts class labels based on the distribution of the training data, without learning any patterns from the input features. This approach is particularly useful for assessing the performance of other, more complex models, serving as a reference point for evaluating the effectiveness of the applied machine learning techniques.

Given that some classes are rarely occurring, conformal prediction is used to improve classification results by estimating uncertainty and removing predictions that are uncertain (Angelopoulos & Bates, 2021). Conformal Prediction is a ML framework that provides a method for assigning reliable confidence measures to individual predictions (Vovk et al., 2005). By constructing prediction regions that accommodate a predefined significance level, Conformal Prediction enables the assessment of credibility predictions, allowing for quantification of the uncertainty associated with the predictions made by machine learning models. This framework is particularly useful for enhancing the reliability of predictions and has found applications in various domains, including text classification, healthcare, and computer vision, where the accurate assessment of prediction credibility is crucial in decision-making processes (Angelopoulos & Bates, 2021).

In this study, the algorithms are implemented using the scikit-learn library for Python (Pedregosa et al., 2011). The default parameters are left unchanged except for two parameters: SVM uses the SVC implementation with an RBF kernel, and Random Forest is set to use 500 estimators. The Uniform classifier uses the DummyClassifier with a strategy set to "uniform". KB/BERT is available via the transformers package (Wolf et al., 2020).

3.3. Experiment setup

In our experiment, we use two approaches: a flat and a hierarchical approach. These are further described below. The experiments are evaluated using the F1-score. The F1-score is the harmonic mean of precision and recall, where precision is $TP / (TP + FP)$ and recall is $TP / (TP + FN)$, given that TP is true positives, FP is false positives, and FN is false negatives. As such, the F1-score is calculated as per the following equation:

$$F1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

In a multi-class setting, this is calculated per class and averaged. The train dispatcher process (TKL) is the optimal performance and is based on the manual

classification effort. This is done by computing the F1-score using the true labels for day 0 and day 10, giving an indication of how well the operatives can classify the delay attribution codes on day 0. The F1-score is described in more detail in the following section.

3.3.1. Flat classification

The flat classification approach (i.e., flat) concerns whether we can classify the text regarding the second and third-level delay attribution code. In this scenario, the text is transformed using TF-IDF, the algorithm trained with the processed text, and the second or third-level delay attribution code as class labels. For the two levels of class-labels, the following is done: A 10-fold cross-conformal validation strategy is used, i.e., an ordinary 10-fold cross-validation, but the training set is split equally into a training and calibration set. The model is trained on the remaining training data, calibrated on the calibration set, and finally, the model is evaluated on the test set. The experiment is evaluated using the F1-score.

3.3.2. Hierarchical classification

In the second approach, delay attribution codes are classified at varying levels of granularity using the verbose delay attribution codes represented as n_1 , n_2 , and n_3 . A hierarchical classification system comprising three levels is implemented, enabling classification of instances into increasingly specific categories, allowing for a more nuanced and accurate classification process (Kowsari et al., 2019; Sebastiani, 2002). This is similar to Stacking SVM (Kowsari et al., 2019; Sun & Lim, 2001). However, instead of having binary-class classifiers for each leaf, multi-class classifiers are used for each node. This approach is utilized for both the SVM and the Random Forest algorithm. A visual representation of the hierarchical approach is shown in Figure 3, illustrating how classifiers are trained for each level.

At the initial level, classification is attempted based on n_1 class labels. This involves transforming the text using TF-IDF, training an algorithm on the

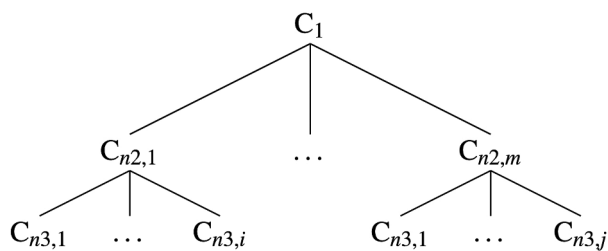


Figure 3. The hierarchical classification approach visualized. Each node C is a multi-class classifier, trained using a specific delay attribution code and its sub-classes as classification targets.

processed text, and utilizing n_1 codes recorded on day 0 (referred to as feature n_{10}) as class labels. Model evaluation is performed using 10-fold cross-validation, with an equal division of the training set into training and calibration subsets. The model is trained on the remaining training data, and predictions are calibrated using the calibration set (i.e., a cross-conformal approach). Performance assessment employs the previously outlined evaluation metric and includes evaluation against both the n_1 codes recorded on day 0 and day 10 (i.e., features n_{10} and n_{110}).

In the second level of the classification hierarchy, a comparable approach to the first level is applied. However, in this context, a distinct model is constructed for each class within n_{10} . Data samples are extracted for each class within n_{10} , and the corresponding labels from n_{20} are used as target labels. Given that n_{10} contains four distinct target labels, this approach yields four distinct models, each tailored to one of the four datasets.

Advancing to the third level, a similar evaluation methodology is retained. For every class within n_{20} , data samples are collected, and a model is developed using n_{30} as target labels. Similarly to before, performance evaluation entails assessing this model against both n_{30} and n_{310} target labels. This multi-level approach enables a progressive classification of delay attribution codes, accommodating increasing levels of specificity.

3.4. Statistical evaluation

When evaluating the difference between a flat versus a hierarchical approach, Kruskal-Wallis and the Conover post-hoc test are used to determine where, if any, statistical significance is manifested (Sheskin, 2003). To evaluate the performance of the approaches for the classification of delay attribution codes on Level 1, Kruskal-Wallis is used to investigate whether there is a statistically significant difference between the algorithms for day 0 and day 10. Conover post-hoc test is used to determine how the difference manifests (Sheskin, 2003).

For the classification of delay attribution codes on Level 2 and 3, the different delay attribution codes investigated are seen as different datasets (e.g., D, I, J, O for Level 2). As such, Friedman's test is used to investigate whether there is a statistically significant difference between the algorithms, and a Nemenyi post-hoc test to determine how the difference manifests across the approaches (Demšar, 2006; Sheskin, 2003). Friedman's Test ranks based on the mean evaluation of metrics and is, consequently, more conservative than Kruskal-Wallis. It is expected that Kruskal-Wallis will detect more differences than Friedman's test (Sheskin, 2003).

4. Results

In the following subsections, the experimental results are presented together with statistical analysis. In the next subsection, results comparing the Flat versus the Hierarchical approach are presented, for level two and three. In the subsequent subsections, the results for the Hierarchical approach are presented for each hierarchical level.

4.1. Flat vs hierarchical classification

Comparing a flat vs a hierarchical classification approach indicates that the hierarchical approach performs better on levels two and three. This can be observed in Table 2. In this scenario, classification on level one between the two approaches is equivalent and, as such, is not interesting to compare.

While the hierarchical approach generally yields a higher F1-score than the flat approach, the standard deviation is also higher. This indicates that some classes have a higher performance than the mean. This is not unexpected, as the hierarchical approach yields means across the different classes.

For the Random Forest and SVM algorithms, the hierarchical approach performs better than the flat approach. However, the hierarchical approach has a higher standard deviation, indicating that some classes might be more difficult to classify than others. For Random Forest at Level 2, the hierarchical approach outperforms the flat approach (F1-score of 0.860 ± 0.102 vs. 0.777 ± 0.010), and at Level 3, the hierarchical approach maintains a competitive performance (F1-score of 0.733 ± 0.113 vs. 0.708 ± 0.009). SVM performs similarly at Level 2, where the hierarchical approach outperforms the flat approach (F1-score of 0.876 ± 0.100 vs. 0.797 ± 0.008). While at Level 3, a similar trend is observed, with the hierarchical approach showing improved performance (F1-score of 0.737 ± 0.114 vs. 0.723 ± 0.009).

The better performance observed with the hierarchical approach can be largely attributed to the reduced effective solution space at each decision point. In the flat configuration, each of the three level-wise classifiers must discriminate among the

full complement of categories, many of which are semantically similar and often overlap in feature space (e.g., the JPR and DPR categories). Consequently, flat models are forced to learn highly discriminative boundaries across a dense, partially overlapping class landscape, thereby increasing the risk of misclassification. By contrast, the hierarchical approach decomposes the problem: at any given node, only a subset of child classes is considered. This restriction not only shrinks the solution space and thereby simplifies the decision boundary but also decreases the potential overlap that can occur in the flat strategy.

Looking at the uniform classifier, the solution space for the hierarchical models is much smaller than for the flat approach. Since the uniform classifier randomly sets the class, the score should be approximately $1/|c|$ where c is the number of classes in the solution space. At Level 2, the hierarchical approach significantly outperforms the flat approach (F1-score of 0.335 vs. 0.075). At Level 3, a similar pattern is observed, with the hierarchical approach showing improved performance (F1-score of 0.313 vs. 0.023). However, the standard deviation for the hierarchical solution is much higher, indicating greater variance in the space for the hierarchical solution.

A Kruskal-Wallis test showed significant differences in means for both the third level ($H = 455.1741$, $p < 0.001$) and the second level ($H = 208.1459$, $p < 0.001$). A Conover's post-hoc test indicates that the difference between the uniform classifier and the other approaches ($p < 0.001$), is statistically significant at both the second and the third level, see Tables 3 and 4. There is a statistically significant difference between Random Forest flat and Hierarchical ($p < 0.05$) and the Hierarchical SVM ($p < 0.001$), on the second level. The test does not find any statistically significant differences between the flat SVM and the Random Forest approaches, for either level. The hierarchical SVM performs better on the second level. There is a significant difference between the flat and the hierarchical approach for the SVM algorithm ($p < 0.05$). This indicates that the hierarchical approach performs statistically significantly better than the flat approach for SVM and Random Forest, on the second level. KB/BERT Hierarchical, on the second level, didn't perform significantly better than the Uniform classifier. However, the KB/BERT Flat performed similarly to the Flat Random Forest and SVM. But the Hierarchical Approaches performed better than KB/BERT ($p < 0.05$). On the third level, the flat KB/BERT did not perform significantly worse than Random Forest or SVM, independent of the approach.

A notable difference between the two approaches is that the hierarchical approach finishes in half the time that the flat approach takes, ≈ 8 minutes vs ≈ 16 minutes, at least in this experimental setting.

Table 2. Mean F1-score (and standard deviation) per algorithm for the two different approaches.

Algorithm		Level 2	Level 3
Uniform Classifier	Flat	0.075 (0.004)	0.023 (0.002)
	Hierarchical	0.335 (0.113)	0.313 (0.157)
Random Forest	Flat	0.777 (0.010)	0.708 (0.009)
	Hierarchical	0.860 (0.102)	0.733 (0.113)
SVM	Flat	0.797 (0.008)	0.723 (0.009)
	Hierarchical	0.876 (0.100)	0.737 (0.114)
TKL	Flat	0.972 (0.004)	0.962 (0.004)
	Hierarchical	0.963 (0.034)	0.922 (0.066)
KB/Bert	Flat	0.771 (0.006)	0.687 (0.006)
	Hierarchical	0.270 (0.140)	0.239 (0.152)

Table 3. Conover post-hoc test results comparing performance between flat and hierarchical classifiers on the second level.

	KB/Bert, F	KB/Bert, H	Uniform, F	Uniform, H	Random forest, F	Random Forest, H	SVM, F	SVM, H	TKL, F	TKL, H
KB/Bert, F	1.0	<0.01	<0.01	<0.01	0.805	<0.05	0.805	<0.01	<0.01	<0.01
KB/Bert, H		1.0	0.108	0.178	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
Uniform, F			1.0	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
Uniform, H				1.0	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
Random Forest, F					1.0	<0.05	0.805	<0.01	<0.01	<0.01
Random Forest, H						1.0	0.433	0.340	<0.01	<0.01
SVM, F							1.0	<0.05	<0.01	<0.01
SVM, H								1.0	<0.01	<0.01
TKL, F									1.0	0.805
TKL, H										1.0

Table 4. Conover post-hoc test results comparing performance between flat and hierarchical classifiers on the third level.

	KB/Bert, F	KB/Bert, H	Uniform, F	Uniform, H	Random forest, F	Random Forest, H	SVM, F	SVM, H	TKL, F	TKL, H
KB/Bert, F	1.0	<0.01	<0.01	<0.01	1.0	0.332	1.0	0.211	<0.01	<0.01
KB/Bert, H		1.0	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
Uniform, F			1.0	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
Uniform, H				1.0	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
Random Forest, F					1.0	1.0	1.0	1.0	<0.01	<0.01
Random Forest, H						1.0	1.0	1.0	<0.01	<0.01
SVM, F							1.0	1.0	<0.01	<0.01
SVM, H								1.0	<0.01	<0.01
TKL, F									1.0	1.0
TKL, H										1.0

Independent of the approach, KB/BERT takes multiple hours to finish.

4.2. Hierarchical classification

4.2.1. Level 1

The mean F1 score for the different algorithms is presented in Table 5. The results indicate that the difference between Random Forest and SVM is minor, but that both algorithms perform significantly better than the uniform classifier. On Day 0, Random Forest had a mean F1-score of 0.890 (± 0.008), and on Day 10, it had an F1-score of 0.889 (± 0.008). Similarly, SVM exhibited strong performance, with F1-scores of 0.901 (± 0.006) on Day 0 and 0.899 (± 0.006) on Day 10. The TKL classifier is the performance of the manual classification process (i.e., the F1-score calculated on the delay attribution code for day 0 compared to day 10). The results indicate that the models can correctly classify the delay attribution code for level 1 using only

the unstructured text from day 0 in $\approx 89\%$ of the tested instances. Further, the results in Table 5 indicate that the differences in performance between day 0 and day 10 are minimal. A Kruskal–Wallis test revealed a significant difference in means ($H = 82.165$, $p < 0.01$). A Conover’s post-hoc test indicates that the difference between all five algorithms is statistically significant ($p < 0.001$), as shown in Table 6. This indicates that the medians of the different samples differ from one another. Given the small standard deviation shown in Table 5, this isn’t surprising.

4.2.2. Level 2

The mean F1 score for the different algorithms is presented in Table 7. The results indicate that the SVM and Random Forest perform significantly better than the Uniform Classifier, but worse than the manual classifier (TKL). In examining the performance of Random Forest and SVM across various classes (D, I, J, O) at Day 0 and Day 10, both

Table 5. Mean F1-score for the different algorithms on days 0 and 10 for the first-level delay attribution codes. Standard deviation within parentheses.

Algorithm day	Uniform classifier	Random forest	SVM	KB/Bert	TKL
0	0.260 (0.006)	0.890 (0.008)	0.901 (0.006)	0.881 (0.006)	0.992 (0.003)
10	0.260 (0.006)	0.889 (0.008)	0.899 (0.006)	0.879 (0.005)	

Table 6. Conover post-hoc test results for the first level delay attribution codes. $p < 0.01$ denotes where statistically significant differences were found.

	KB/Bert	Uniform	Random forest	SVM	TKL
KB/Bert	1.000	<0.01	<0.01	<0.01	<0.01
Uniform	<0.01	1.000	<0.01	<0.01	<0.01
Random Forest	<0.01	<0.01	1.000	<0.01	<0.01
SVM	<0.01	<0.01	<0.01	1.000	<0.01
TKL	<0.01	<0.01	<0.01	<0.01	1.000

Table 7. Mean F1-score (and standard deviation within parentheses) for the approaches when classifying the second-level delay attribution codes.

Code	Algorithm	Uniform classifier	Random forest	SVM	KB/Bert	TKL
Code	Day					
D	0	0.344 (0.007)	0.938 (0.007)	0.946 (0.007)	0.792 (0.000)	0.992 (0.002)
	10	0.342 (0.007)	0.936 (0.007)	0.943 (0.008)	0.791 (0.001)	
I	0	0.359 (0.029)	0.899 (0.023)	0.895 (0.020)	0.624 (0.004)	0.924 (0.023)
	10	0.347 (0.028)	0.835 (0.039)	0.833 (0.036)	0.573 (0.019)	
J	0	0.167 (0.006)	0.690 (0.015)	0.712 (0.018)	0.264 (0.002)	0.942 (0.005)
	10	0.168 (0.009)	0.684 (0.016)	0.707 (0.016)	0.258 (0.010)	
O	0	0.469 (0.051)	0.912 (0.030)	0.953 (0.029)	0.851 (0.009)	0.995 (0.012)
	10	0.464 (0.055)	0.909 (0.030)	0.956 (0.028)	0.848 (0.011)	

algorithms demonstrate competitive results. Random Forest demonstrates strong performance, achieving high F1-scores, particularly in classes D and O, ranging from 0.938 to 0.936 and 0.912 to 0.909, respectively, at both evaluation time points. Similarly, SVM demonstrate commendable performance, achieving competitive F1-scores, especially in classes such as D and O, ranging from 0.946 to 0.943 and 0.953 to 0.956. However, for the delay attribution code J, both Random Forest and SVM perform worse than for the other delay attribution codes (0.690 and 0.712, respectively, at day 0, indicating that the algorithms have trouble separating some of the classes for this code. Similarly, KB/BERT performs poorly on the J attribution codes (0.26), notably lower than that of SVM and Random Forest. The uniform classifier also has a lower mean, indicating that the

J code has a larger solution set (i.e., more classes) than the other delay attribution codes.

A Friedman test revealed a statistically significant difference between the algorithms, $\chi^2(29) = 30.8, p < 0.01$. A Nemenyi post-hoc test indicates that there is a statistically significant difference between the uniform classifier and Random Forest ($p < 0.05$), SVM ($p < 0.01$), and TKL ($p < 0.01$) but not for KB/BERT, see Table 8. Further, there is a statistically significant difference between TKL and KB/BERT ($p < 0.01$).

4.2.3. Level 3

The mean F1 score for the different algorithms for the third level of the delay attribution codes is presented in Table 9. The delay attribution codes in Table 9 are grouped by the level 2 parent category. In the table, the level 2 codes presented concerns delays attributed to

Table 8. Nemenyi post-hoc results for the second-level delay attribution codes. $p < 0.05$ and $p < 0.01$ denotes where a statistical significant difference exists.

	Uniform	Random Forest	SVM	TKL	KB/Bert
Uniform	1.000	<0.05	<0.01	<0.01	0.687
Random Forest	<0.05	1.000	0.900	0.175	0.508
SVM	<0.01	0.900	1.000	0.508	0.175
TKL	<0.01	0.175	0.508	1.000	<0.01
KB/Bert	0.687	0.508	0.175	<0.01	1.000

Table 9. Mean F1-score (and standard deviation within parentheses) for the approaches when classifying the third-level delay attribution codes.

Code	Algorithm	Uniform Classifier	Random Forest	SVM	KB/Bert	TKL
Code	Day					
DPS	0	0.497 (0.014)	0.874 (0.012)	0.882 (0.005)	0.881 (0.002)	0.954 (0.013)
	10	0.490 (0.015)	0.898 (0.020)	0.907 (0.013)	0.908 (0.011)	
IBT	0	0.602 (0.106)	0.733 (0.050)	0.742 (0.051)	0.733 (0.023)	0.787 (0.089)
	10	0.489 (0.143)	0.538 (0.106)	0.543 (0.119)	0.529 (0.095)	
IBÖ	0	0.455 (0.069)	0.880 (0.037)	0.888 (0.030)	0.725 (0.016)	0.945 (0.038)
	10	0.440 (0.060)	0.848 (0.038)	0.848 (0.044)	0.715 (0.021)	
ISA	0	0.200 (0.032)	0.682 (0.048)	0.708 (0.031)	0.311 (0.031)	0.915 (0.031)
	10	0.186 (0.035)	0.622 (0.054)	0.644 (0.039)	0.294 (0.023)	
JDM	0	0.175 (0.034)	0.645 (0.052)	0.624 (0.044)	0.370 (0.004)	0.992 (0.009)
	10	0.174 (0.034)	0.642 (0.053)	0.619 (0.046)	0.368 (0.006)	
JIA	0	0.167 (0.046)	0.581 (0.032)	0.585 (0.036)	0.363 (0.003)	0.874 (0.017)
	10	0.153 (0.042)	0.503 (0.029)	0.504 (0.034)	0.365 (0.011)	
JPR	0	0.165 (0.030)	0.581 (0.027)	0.585 (0.031)	0.238 (0.010)	0.913 (0.025)
	10	0.157 (0.030)	0.561 (0.029)	0.563 (0.036)	0.243 (0.014)	
JPS	0	0.286 (0.043)	0.842 (0.019)	0.829 (0.041)	0.692 (0.004)	0.979 (0.015)
	10	0.285 (0.042)	0.828 (0.021)	0.813 (0.042)	0.673 (0.015)	
JUF	0	0.264 (0.027)	0.729 (0.044)	0.733 (0.028)	0.545 (0.003)	0.911 (0.025)
	10	0.239 (0.027)	0.658 (0.054)	0.664 (0.038)	0.487 (0.019)	
OMÄ	0	0.316 (0.074)	0.771 (0.044)	0.784 (0.044)	0.484 (0.043)	0.953 (0.022)
	10	0.316 (0.074)	0.771 (0.044)	0.784 (0.044)	0.484 (0.043)	

Railway Companies (J), and more specifically events related to locomotive or engine wagon-related issues (DM), incidents that occur before departure (IA) or during transport (UF), prioritization decisions made by the company (PR), or Personnel-related events (PS). The infrastructure category (I) includes railway work or transport disruptions (BT), railway superstructure-related issues (BÖ), or signal installations and signaling systems issues (SA). Operational Management (D) delays that involve personnel issues are captured with a personnel sub-code (PS), while the accident category (O) relates to human incidents (MÄ). The results indicate that the SVM and Random Forest perform significantly better than the Uniform Classifier, but worse than the manual classifier (TKL). Random Forest and SVM demonstrate competitive performance across different delay attribution codes, with F1-scores ranging from 0.503 to 0.898 and 0.504 to 0.907, respectively. Similar to level 2, both Random Forest and SVM have difficulties when classifying most J-based delay attribution codes (e.g., JIA and JPR) compared to the other delay attribution codes, indicating that the algorithms have trouble separating some of the classes for the specific delay attribution code. The uniform classifier also has a lower mean, indicating that the J code has a larger solution set (i.e., more classes) than the other delay attribution codes. At level 3, it is likely that the overlap between different delay attribution codes is higher than on the previous levels, e.g., between IBT- and IBT40. Additionally, the dash delay attribution code (“-”) is also present at level 3, acting as the delay attribution code when train dispatchers cannot classify on the third level. This also increases the chances of overlap between delay attribution codes. While most classes have similar performance between day 0 and day 10, this is not the case for the IBT code. The IBT code performs similarly to the Uniform Classifier at day 10. However, this is not unexpected as the solution set for IBT codes increases from two codes (IBT-, IBT40) to 6 codes (IBT-, IBT21, IBT22, IBT30, IBT40). Since the model has not been able to train on the new classes, it cannot be expected to correctly classify them either, thus decreasing the performance. It should also be noted that the TKL performance for the IBT code is lower than that of the other codes, indicating that it is a challenging code to classify.

Similar to Level 2, there seem to be attribution codes that KB/BERT is unable to classify correctly,

e.g., ISA (F1-scores ranging from 0.294 to 0.311) and JPR (F1-scores ranging from 0.243 to 0.238).

A Friedman test revealed a statistically significant difference between the algorithms across the different delay attribution codes, $\chi^2(99) = 73.12$, $p < 0.01$.

A Nemenyi post-hoc test indicates that there is a statistically significant difference between the uniform classifier and Random Forest ($p < 0.05$), SVM ($p < 0.01$), and TKL ($p < 0.01$), see Table 10. Furthermore, there is a statistically significant difference between TKL and Random Forest ($p < 0.01$) and KB/BERT ($p < 0.05$). Similarly, SVM performed significantly better than KB/BERT ($p < 0.05$), but not Random Forest. And more interestingly, no significant difference was detected between TKL and SVM ($p = 0.07$). KB/BERT did not perform significantly better than the uniform classifier ($p = 0.115$), which can be explained by the model’s poor performance on, e.g., the ISA fault attribution code.

5. Discussion and implications

Returning to our research question of “*To what extent can machine learning-based methods (flat and hierarchical multi-class text classification) predict delay attribution codes*”, our results affirm the viability of employing a machine learning-based approach to classify delay codes. The hierarchical approach outperforms the flat approach, and the SVM model generally performs better than the RF and KB/BERT models. Based on the short event descriptions made by train dispatchers on Day 0, we predict the codes that are set on Day 10 in the manual process with an accuracy of approximately 90 percent for Level 1 (Table 5) and slightly less for Level 2 codes (accuracy ranges between 71 and 95 percent) (Table 7). Setting Level 2 codes is also a requirement for train dispatchers, while quality managers may need to code down to Level 3 during the 10-day process to determine the basis for economic compensation. Moreover, it is essential to note that the TKL responses serve as the “ground truth” against which we compare the ML models, but they are not necessarily an objectively correct classification of codes. Rather, it is the outcome of the complete process in which multiple actors have been involved to “negotiate” an outcome. Thus, by providing ML models with free-text descriptions of the events, train dispatchers will receive instant

Table 10. Nemenyi post-hoc test for the third-level delay attribution codes. $p < 0.05$ and $p < 0.01$ denotes where a statistical significant difference exists.

	Uniform	Random Forest	SVM	TKL	KB/Bert
Uniform	1.000	<0.01	<0.01	<0.01	0.115
Random Forest	<0.01	1.000	0.724	<0.01	0.374
SVM	<0.01	0.724	1.000	0.070	<0.05
TKL	<0.01	<0.01	0.070	1.000	<0.01
KB/Bert	0.115	0.374	<0.05	<0.01	1.000

classifications of the codes for Day 10 on Day 0 (with an accuracy of up to 90 percent for Level 3 codes, see [Figure 1](#)). In other words, since ML models can instantly classify codes, the time window for quality managers and other stakeholders could be reduced, and resources could be allocated more effectively. However, to fully automate decision-making, the ML approach needs to be able to classify Level 3 codes with similar accuracy as Level 1 and 2. The SVM model performed best, reaching up to or close to 90 percent accuracy for codes like DPS and IBÖ, but also showed lower overall accuracy and more variability for the remaining predictions on Level 3 ([Table 9](#)) than for Levels 1 and 2. In turn, these findings hold further academic, methodological, organizational, and practical implications, which are further discussed below.

5.1. Methodological implications: decision accuracy and robustness of the machine learning approach

In the result section, we observed that the hierarchical models consistently outperform flat counterparts. Specifically, the hierarchical SVM demonstrates a significant improvement over the flat SVM and flat Random Forest models, while the hierarchical Random Forest outperforms the flat Random Forest. This performance gain is expected, given the reduced solution space of the hierarchical models; approximately six possible classes compared to the flat models, approximately 43 classes. However, a trade-off emerges, as evidenced by the hierarchical model's higher standard deviation ([Table 2](#)). The higher variance is anticipated, reflecting greater variability across different classes. Interestingly, KB/BERT did not perform well. Especially, the hierarchical approach performed poorly. Even though the flat KB/BERT approach performed similarly to the flat SVM, the Hierarchical SVM performed better than KB/BERT. This suggests that the text embeddings are more susceptible to class overlap than the TF-IDF tokenization. Using language knowledge does not seem to aid classification, rather, for some classes, KB/BERT generates embeddings that the classification head is unable to distinguish between the classes.

[Table 9](#) reveals variations in classification ease across classes, with some, such as IBÖ, exhibiting clearer distinctions compared to more challenging cases, like JPR. It is essential to recognize that certain classes share common features, and the event descriptions may overlap in content. This is most easily exemplified in the third-level codes, where codes may contain numeric representations or a dash ('-'), with the dash indicating a default class when train dispatchers are unable to specify a code.

When conducting our experiments, we also assessed the impact of excluding instances where only numeric data were available. This involved retaining approximately 13,500 instances from the original dataset, which originally consisted of approximately 21,500 instances. Intriguingly, the results of this evaluation revealed minimal differences in the performance of the algorithm's F1-score; the models displayed similar proficiency even after removing these specific data points. The results are available in [Appendix A](#).

This finding prompts consideration of the models' treatment of instances with only numeric data during the training process. Notably, the limited performance disparity suggests that the models may not be effectively incorporating or discerning patterns from instances solely comprised of numeric values. While removing such instances does not significantly impact overall performance metrics, it raises questions about the models' ability to generalize and derive meaningful insights from these particular data points. Further investigation into the mechanisms of feature importance and model interpretability could shed light on the extent to which these instances contribute to the overall learning process. Importantly, we recognize that this exclusion encompasses thousands of instances, each representing delay scenarios that require accurate classification, such as train delays caused by other trains. The sheer volume underscores the importance of understanding how the models handle these cases and the potential implications of excluding them. Understanding the role of such instances in the training dynamics is essential for refining the model's capacity to discern relevant information from diverse inputs, thereby enhancing its adaptability and effectiveness in real-world scenarios.

Moreover, while there is a risk of misclassification, especially when the model lacks sufficient training data or encounters novel patterns, this could be mitigated by regular model updates, continuous training, and manual handling of edge cases. We recognize the importance of maintaining a delicate balance and avoiding over-reliance on models by encouraging train dispatchers to critically assess predictions, particularly in ambiguous situations where contextual insights are paramount. This is especially important, as models may struggle to understand context beyond the provided features, thereby missing domain-specific insights known to train dispatchers. We discuss these and other organizational and practical implications below.

5.2. Implications and generalizability of machine learning models for the hybrid decision support system

Developing a decision-support system for the delay code attribution process is limited by the work of

train dispatchers. They generate subjective data that underpins the decision process. The train dispatchers set delay codes and write brief event descriptions, which are used to predict day 10. Setting the codes requires information from several systems and interactions with multiple individuals. Such an approach intertwines human cognition with communication between individuals, both within and outside technical systems, rendering decision-making a matter of socio-technical systems (Saward & Stanton, 2018). Despite relying on the train dispatchers' initial event descriptions, the ML-based model used in this study provides a standardized and objective HDSS, rather than serving as a means to fully substitute "humans in the loop" (Munro, 2021). The interaction pattern with the ML models would therefore best be categorized as a "human oversight" or interactive collaboration model (Punzi et al., 2023).

Besides the model's potential in predicting delay codes, it can be adjusted to complement train dispatchers' work in a way that is overall beneficial for the organization. For instance, the ML model can assist in identifying and prioritizing instances with a higher likelihood of misclassification, allowing train dispatchers to focus their attention where it matters the most. The model may also be complemented with certainty estimation measures, facilitated by conformal prediction² and thereby adds an additional layer of value to the collaborative interaction between the socio-technical system (the model) and the train dispatchers. By offering a measure of certainty or confidence in its predictions, the model empowers train dispatchers to make more informed decisions. The uncertainty measures could also be visualized, separating codes with high certainty of correct classification from those with a low. Such a risk visualization becomes especially important when the model's confidence is lower, prompting train dispatchers to exercise heightened diligence and potentially seek additional information before assigning a code. Thus, the uncertainty estimates together with our evaluation of sets of models are expected to simultaneously: help guide train dispatchers' attention to problematic instances, to facilitate trust calibration between the ML models and humans (Kares et al., 2023), and to "unboxing" the ML models, contributing to Explainable AI (Kostopoulos et al., 2024).

In other words, the models offer benefits to multiple stakeholders. First, from an operational perspective, the model should not be considered a replacement of train dispatchers in the delay attribution process, but rather to complement them, serving as a "second opinion". The ML support aligns closely with human assessors for Level 1 and Level 2 codes. Thus, there is potential for automating Level 1 and 2 codes, while Level 3 codes may require hands-on work. Since the ML-model predicts the code Day 0

the refinement of Level 3 codes (or any other codes that show risk of being misclassified), may be done in the days following the event, potentially reducing the 10-day time window of the process as well as providing for more efficient use of resources (see Figure 1, the dotted lines at the bottom of the figure).

Second, the model serves as a useful tool for quality evaluators, highlighting instances of disagreement between the model's predictions and train dispatchers' decisions. A disagreement between the model and dispatchers could indicate a potential change in the final code on day 10. However, this assumes that the code in question belongs to a class for which the model has demonstrated a low classification error rate (i.e., indicating a higher certainty). Third, this functionality proves particularly beneficial for new employees navigating the intricacies of the delay attribution coding process, as well as in situations where certain codes have historically exhibited a higher error rate. Thus, the model may help limit the decision space (Simon, 1990).

Fourth, the model can serve as a continuous learning resource for training dispatchers, enabling them to refine their coding decisions in real-time based on the latest patterns and trends captured by the algorithm. Finally, the results of our investigation point to the fact that hierarchical models are faster than their flat counterparts. The latter is beneficial from an organizational perspective, as accurately predicting codes and providing guidance on code uncertainty enables the organization to reduce time spent on routine tasks, thereby increasing performance and strengthening operational capabilities (Wu et al., 2010).

Supervised ML and NLP have been applied in healthcare to classify incident types and severity from free-text reports (Evans et al., 2020) or discharge summaries (Remmer et al., 2021). National critical incident report analyses (CIRS) demonstrate that topic modeling and LLMs can extract themes, such as medication errors, from large text repositories, thereby aiding triage when human review is limited (Denec et al., 2024). Hierarchical structures in clinical coding improve weighted F1-scores for disease-category prediction, particularly for rare conditions, compared to flat labels (Catling et al., 2018). Similarly, in logistics, unsupervised NLP models identify risk themes and temporal patterns—e.g., pandemic impacts or fuel price volatility – often before structured systems detect disruptions (see Sadeek & Hanaoka, 2023). In claims processing, NLP automates entity extraction and claim categorization, reducing manual errors, and improving throughput (Sadeek & Hanaoka, 2023).

These cases parallel our setting in three ways: (1) hierarchical coding for accountability, (2) delayed validation as new evidence emerges, and (3) reliance on free-text narratives. Our hierarchy-aware approach

can pre-assign medical codes from incident narratives, flag ambiguous cases for review, and propagate certainty across taxonomic levels. Similarly, classifiers trained on operational narratives can assign provisional “cause codes” for shipment delays and update labels as telemetry or partner confirmations arrive – analogous to staged validation with dispatchers and experts (Sadeek & Hanaoka, 2023). Hierarchical modeling is critical for improving recall on rare classes while maintaining auditability for high-stakes decisions.

6. Limitations and future work

The analysis highlights instances where codes are infrequent, such as the absence of F codes in the dataset, and cases like IBT codes, where models struggle due to the limited number of instances. Notably, we observe a substantial discrepancy between day 0 and day 10 for IBT codes, likely attributed to the small sample size (200 instances). The impact of sample sizes becomes particularly evident in our evaluation. Classes with limited instances, such as IBT, are more sensitive to dataset variations, which affects the performance metrics of the models. Moreover, when instances are sparse, models may encounter challenges in effectively generalize patterns. Conversely, for classes with larger sample sizes, the models benefit from a more robust training set, enabling better adaptability and yielding more stable performance across different evaluation points. Understanding the influence of sample sizes on class-specific model performance is crucial for refining the training approach. For classes with limited instances, strategies such as oversampling, data augmentation, or targeted collection efforts may be considered to address the imbalance and enhance model learning. In other words, the observation highlights the necessity of a strategic dataset curation, ensuring representative samples across all relevant classes to foster a balanced and comprehensive model training process.

Another noteworthy aspect discovered during the investigation is the presence of several events characterized by only one or two numeric values. These numerical entries often correspond to train identifications, signifying, e.g., instances where one train has caused a delay in subsequent trains. This phenomenon was not confined to any specific delay attribution code. However, since the text associated with these instances consists solely of numeric values, the models struggle to interpret the underlying meaning of the numeric values. Unlike the models, train dispatchers possess the ability to discern the significance of these numbers as they draw on knowledge beyond the confines of the unstructured text.

This observation underscores the critical importance of ensuring the feature of separability across

classes and enabling models to grasp the contextual meaning of the text. For effective machine learning, features should be designed to capture distinct characteristics that enable models to accurately discriminate between various classes. In the context of our investigation, the presence of only numeric values in certain instances poses a challenge as it hampers the model’s capacity to discern the nuanced differences between events. To address this limitation, the event description should be complemented with additional contextual information. This can be achieved either during the text creation phase by train dispatchers, who can provide relevant insights alongside the numeric values, or through post-processing steps involving rule-based additions to enrich the dataset. In other words, the single numeric values that are stated in the information train dispatchers create are, to this point, an intangible aspect of their decision-making routine.

Besides the limitations of data and analytics a complementary research track is encouraged. Since the data to be used for analytics is generated by the dispatchers, we also suggest an organizational change perspective that concerns how to best write a free-text that captures the necessary details about train delays. An updated routine beyond short, concise, and neutral may be necessary.

7. Conclusion

The results suggest the feasibility of the hierarchical classification approach based on the performance of Support Vector Machines (SVM) and Random Forests at different levels. The SVM algorithm, with F1-scores of 0.876 (Level 2) and 0.737 (Level 3), and the Random Forest algorithm, with scores of 0.860 (Level 2) and 0.733 (Level 3), performed statistically significantly better than their flat counterparts on Level 2. These results underscore the ability of these algorithms to capture hierarchical relationships within the delay attribution coding process. The hierarchical framework, extending to Level 3, demonstrates potential to discern granular distinctions in delay attribution codes. The success of SVM and Random Forest in navigating the hierarchical structure suggests the broader applicability and feasibility of this approach, enabling delay attribution code classification systems for this type of decision task. In comparison, for KB/BERT, the flat approach performed better than the Hierarchical. However, compared to KB/BERT, SVM still performed better (Flat KB/BERT, F1-score of 0.687, vs Hierarchical SVM, F1-score of 0.737). The hierarchical KB/BERT performed similarly to the uniform classifier with an average F1-score of 0.239 on the third-level classification.

Moreover, the outcome of employing a hierarchical approach with SVM, Random Forest, compared to

both manual train dispatchers and a Uniform Random Classifier, for third-level classification, indicates the feasibility of this methodology. Despite that, when compared to the outcome of the 10-day process of the train dispatchers (TKL), SVM and Random Forest demonstrate lower accuracy and more variability; they also show statistically significant improvements compared to the Uniform Classifier, and in most cases KB/BERT. For instance, in the case of the DPS delay attribution code at Day 0, SVM achieves an F1-score of 0.882, surpassing the Uniform Classifier's 0.497. This signifies the potential of SVM and Random Forest to improve delay attribution code classification efficiency, even if they currently fall short of the train dispatchers' performance on the finest level of analysis. Furthermore, our findings underscore the complexity in classifying certain delay attribution code classes, exemplified by, e.g., the JPR codes on level 3, which is challenging for the SVM model and the train dispatchers.

Incorporating modern language representation techniques, using a pre-trained transformer model based on BERT, did not yield improved classification from the unstructured text. However, it should be noted that the domain-specific language used in the text could be difficult to interpret for general large language models.

Overall, the findings show that the models can complement train dispatchers in their classification tasks but not replace them. The ML approach benefits various stakeholders, including new employees and quality evaluators, and strengthens organizational performance by reducing time spent on routine tasks and enhancing operational capabilities. Additionally, the current research could be extended to explore ways of enhancing the robustness and interpretability of the classification approach. Additionally, exploring certainty estimation methods, particularly those that leverage conformal prediction, offers an avenue to quantify the model's confidence in its predictions. This could provide valuable insights into scenarios where the model is uncertain or prone to misclassification. Moreover, investigating techniques for enhancing the explainability of the model's classifications would contribute to building trust and understanding among end-users.

Notes

1. GOA – Grade of Automation – is a four-level standard (IEC 62290) used to classify the degree to which train operations are automated. ATP – Automatic Train Protection – Refers to a safety system that automatically adjust to speed limits and prevents collisions or red signal overruns. ATO – Automatic Train Operation. – refers to driving functions of the train, such as acceleration, braking, and maintaining speed and schedule. The ATO system is fully automatic in GoA 4 (fully automated operation).

2. A method for estimating the uncertainty of predictions (Angelopoulos & Bates, 2021). The method enables the calculation of the statistical significance of the predictions with regard to a calibration set.

Acknowledgments

We would like to thank the anonymous reviewers, our contacts at the Swedish Transport Administration for their help and domain knowledge as well as Johanna Törnqvist Krasemann for initiating the project.

Author contributions

Anton Borg: Conceptualization, Methodology, Data curation, Writing Original draft preparation, Visualization. Martin Svensson: Conceptualization, Writing Original draft preparation, Methodology. Per Lingvall: Data Collection, Reviewing.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

The work was supported by the Trafikverket [TRV 2021/79668].

References

- Angelopoulos, A. N., & Bates, S. (2021). A gentle introduction to conformal prediction and distribution-free uncertainty quantification. arXiv preprint arXiv: 2107.07511.
- Arnott, D., & Pervan, G. (2005). A critical analysis of decision support systems research. *Journal of Information Technology*, 20(2), 67–87. <https://doi.org/10.1057/palgrave.jit.2000035>
- Arshad, M., & Ahmed, M. (2019). Prediction of train delay in Indian Railways through machine learning techniques. *International Journal of Computer Sciences & Engineering*, 7(2), 405–411. <https://doi.org/10.26438/ijcse/v7i2.405411>
- Askham, N., Cook, D., Doyle, M., Fereday, H., Gibson, M., Landbeck, U., & Schwarzenbach, J. (2013). *The six primary dimensions for data quality assessment*. DAMA UK working group.
- Autor, D. H., & Handel, M. J. (2013). Putting tasks to the test: Human capital, job tasks, and wages. *Journal of Labor Economics*, 31(S1), S59–S96. <https://doi.org/10.1086/669332>
- Baron, N. S. (1998). Letters by phone or speech by other means: The linguistics of eMail. *Language and Communication*, 18(2), 133–170. [https://doi.org/10.1016/S0271-5309\(98\)00005-6](https://doi.org/10.1016/S0271-5309(98)00005-6)
- Borg, A., Boldt, M., Rosander, O., & Ahlstrand, J. (2021, March). e-Mail classification with machine learning and word embeddings for improved customer support. *Neural Computing and Applications*, 33(6), 1881–1902. <https://doi.org/10.1007/s00521-020-05058-4>
- Brandenburger, N., & Naumann, A. (2018). Towards remote supervision and recovery of automated railway

- systems: The staff's changing contribution to system resilience. In *2018 international conference on intelligent rail transportation (icirt)* (pp. 1–5). <https://ieeexplore.ieee.org/abstract/document/8641576>
- Brehmer, B. (1992). Dynamic decision making: Human control of complex systems. *Acta Psychologica*, 81(3), 211–241. [https://doi.org/10.1016/0001-6918\(92\)90019-A](https://doi.org/10.1016/0001-6918(92)90019-A)
- Cao, L. (2023). *Designing human-centered hybrid decision support systems* [Ph.D. Thesis]. University of Gothenburg, Department of Applied Information Technology. <https://gupea.ub.gu.se/handle/2077/75568>
- Catling, F., Spithourakis, G. P., & Riedel, S. (2018). Towards automated clinical coding. *International Journal of Medical Informatics*, 120, 50–61. <https://doi.org/10.1016/j.ijmedinf.2018.09.021>
- Cutler, D. R., Edwards, T. C., Jr., Beard, K. H., Cutler, A., Hess, K. T., Gibson, J., & Lawler, J. J. (2007). Random forests for classification in ecology. *Ecology*, 88(11), 2783–2792. <https://doi.org/10.1890/07-0539.1>
- Dane, E., & Pratt, M. G. (2007). Exploring intuition and its role in managerial decision making. *Academy of Management Review*, 32(1), 33–54. <https://doi.org/10.5465/amr.2007.23463682>
- Dane, E., Rockmann, K. W., & Pratt, M. G. (2012). When should I trust my gut? Linking domain expertise to intuitive decision-making effectiveness. *Organizational Behavior and Human Decision Processes*, 119(2), 187–194. <https://doi.org/10.1016/j.obhdp.2012.07.009>
- Davenport, T. H. (2018). From analytics to artificial intelligence. *Journal of Business Analytics*, 1(2), 73–80. <https://doi.org/10.1080/2573234X.2018.1543535>
- Dawes, R. M., Faust, D., & Meehl, P. E. (1989). Clinical versus actuarial judgment. *Science*, 243(4899), 1668–1674. <https://doi.org/10.1126/science.2648573>
- Dellermann, D., Lipusch, N., Ebel, P., & Leimeister, J. M. (2019). Design principles for a hybrid intelligence decision support system for business model validation. *Electronic Markets*, 29(3), 423–441. <https://doi.org/10.1007/s12525-018-0309-2>
- Demartini, G. (2015). Hybrid human-machine information systems: Challenges and opportunities. *Computer Networks*, 90, 5–13. <https://doi.org/10.1016/j.comnet.2015.05.018>
- Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7, 1–30.
- Denecke, K., & May, R., LLMHealthGroup, & Rivera Romero, O. (2024). Potential of large language models in health care: Delphi study. *Journal of Medical Internet Research*, 26, e52399.
- Dingler, M., Koenig, A., Sogin, S., & Barkan, C. P. (2010). Determining the causes of train delay. In *AREMA Annual Conference Proceedings*. Lanham, MD: The American Railway Engineering and Maintenance-of-Way Association.
- Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv: 1702.08608.
- Duncheon, C. (2002). Product miniaturization requires automation-but with a strategy. *Assembly Automation*, 22(1), 16–20. <https://doi.org/10.1108/01445150210697096>
- Edwards, J. S., & Rodriguez, E. (2019). Remedies against bias in analytics systems. *Journal of Business Analytics*, 2(1), 74–87. <https://doi.org/10.1080/2573234X.2019.1633890>
- Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors*, 37(1), 32–64. <https://doi.org/10.1518/001872095779049543>
- Evans, H. P., Anastasiou, A., Edwards, A., Hibbert, P., Makeham, M., Luz, S., Sheikh, A., Donaldson, L., & Carson-Stevens, A. (2020). Automated classification of primary care patient safety incident report content and severity using supervised machine learning (ML) approaches. *Health Informatics Journal*, 26(4), 3123–3139. <https://doi.org/10.1177/1460458219833102>
- Flach, P. (2012). *Machine learning: The art and science of algorithms that make sense of data*. Cambridge University Press.
- Frohm, J. (2008). *Levels of automation in production systems*. Chalmers University of Technology Göteborg.
- Ghasemaghahi, M., Ebrahimi, S., & Hassanein, K. (2018). Data analytics competency for improving firm decision making performance. *Journal of Strategic Information Systems*, 27(1), 101–113. <https://doi.org/10.1016/j.jsis.2017.10.001>
- Grechi, D., & Maggi, E. (2018). The importance of punctuality in rail transport service: An empirical investigation on the delay determinants. *European Transport-Transporti Europei*, 70, 1–23.
- Groover, M. P. (2016). *Automation, production systems, and computer-integrated manufacturing*. Pearson Education India.
- Gunaratne, J., Zalmanson, L., & Nov, O. (2018). The persuasive power of algorithmic and crowdsourced advice. *Journal of Management Information Systems*, 35(4), 1092–1120. <https://doi.org/10.1080/07421222.2018.1523534>
- Günther, W. A., Mehrizi, M. H. R., Huysman, M., & Feldberg, F. (2017). Debating big data: A literature review realizing value from big data. *Journal of Strategic Information Systems*, 26(3), 191–209. <https://doi.org/10.1016/j.jsis.2017.07.003>
- Gylee, M. (1994). Punctuality analysis—a basis for monitoring and investment in a liberalized railway system. In: *Proceedings of seminar held at the 22nd PTRC*, European Transport Forum, Warwick.
- Hardin, A., Looney, C. A., & Moody, G. D. (2017). Assessing the credibility of decisional guidance delivered by information systems. *Journal of Management Information Systems*, 34(4), 1143–1168. <https://doi.org/10.1080/07421222.2017.1394073>
- Higgins, A., Ferreira, L., & Kozan, E. (1995). Modelling single line train operations. *Transportation Research Record, Journal of the Transportation Research Board, Railroad Transportation Research*, 1489, 9–16.
- Holmberg, L., Davidsson, P., & Linde, P. (2020). A feature space focus in machine teaching. In De La Prieta F. (Ed.), *2020 IEEE international conference on pervasive computing and communications workshops (percom workshops)* (pp. 1–2). Springer, Cham. https://doi.org/10.1007/978-3-030-51999-5_5
- Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In C. Nédellec & C. Rouveirol (Eds.), *Machine learning: ECML-98* (pp. 137–142). Springer Berlin Heidelberg.
- Joborn, M., & Ranjbar, Z. (2022). Understanding causes of unpunctual trains: Delay contribution and critical disturbances. *Journal of Rail Transport Planning & Management*, 23, 100339. <https://doi.org/10.1016/j.jrtpm.2022.100339>
- Johansson, I., Sipilä, H., & Palmqvist, C. W. (2026). Simulating railway punctuality in three Swedish

- metropolitan regions. *Transportation*, 1–20. <https://doi.org/10.1007/s11116-025-10716-4>
- Kahneman, D. (2011). *Thinking, fast and slow*. Macmillan.
- Kares, F., König, C. J., Bergs, R., Protzel, C., & Langer, M. (2023). Trust in hybrid human-automated decision-support. *International Journal of Selection and Assessment*, 31(3), 388–402. <https://doi.org/10.1111/ijsa.12423>
- Kata, G., & Poleszak, W. (2021). Cognitive functioning and safety determinants in the work of a train drivers. *Acta Neuropsychologica*, 19(2), 277–289. <https://doi.org/10.5604/01.3001.0014.9958>
- Kecklund, L., Mowitz, A., & Dimgard, M. (2011). Human factors engineering in train cab design-prospects and problems. *Human Modelling in Assisted Transportation: Models, Tools and Risk Methods*, 327–333.
- Kim, Y. (2014). Convolutional neural networks for sentence classification. CoRR, Abs/1408. 5882. <http://arxiv.org/abs/1408.5882>
- König, E. (2020, June). A review on railway delay management. *Public Transport*, 12(2), 335–361. <https://doi.org/10.1007/s12469-020-00233-1>
- Kostopoulos, G., Davrazos, G., & Kotsiantis, S. (2024). Explainable artificial intelligence-based decision support systems: A recent review. *Electronics*, 13(14), 2842. <https://doi.org/10.3390/electronics13142842>
- Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. *Information*, 10(4), 150. <https://doi.org/10.3390/info10040150>
- Kulkarni, U., Robles-Flores, J. A., & Popovic, A. (2017). Business intelligence capability: The effect of top management and the mediating roles of user participation and analytical decision making orientation. *Journal of the Association for Information Systems*, 18(7), 516–541. <https://doi.org/10.17705/1jais.00462>
- Laifa, H., Khcherif, R., & Ben Ghezalaa, H. H. (2021). Train delay prediction in Tunisian railway through LightGBM model. *Procedia Computer Science*, 192, 981–990. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 25th International Conference KES2021. <https://doi.org/10.1016/j.procs.2021.08.101>
- Liu, Q., Wang, S., Li, Z., Li, L., Zhang, J., & Wen, C. (2023, March). Prediction of high-speed train delay propagation based on causal text information. *Railway Engineering Science*, 31(1), 89–106. <https://doi.org/10.1007/s40534-022-00286-x>
- Logg, J. M., Minson, J. A., & Moore, D. A. (2019). Algorithm appreciation: People prefer algorithmic to human judgment. *Organizational Behavior and Human Decision Processes*, 151, 90–103. <https://doi.org/10.1016/j.obhdp.2018.12.005>
- Malmsten, M., Börjeson, L., & Haffenden, C. (2020). Playing with words at the National Library of Sweden - making a Swedish BERT. <https://arxiv.org/abs/2007.01658>
- Marakas, G. M. (2003). *Decision support systems in the 21st century* (Vol. 134). Prentice Hall Upper Saddle.
- Marsh, R. M., & Mannari, H. (1981). Technology and size as determinants of the organizational structure of Japanese factories. *Administrative Science Quarterly*, 26(1), 33–57. <https://doi.org/10.2307/2392598>
- Mukunzi, G., & Palmqvist, C.-W. (2024). The impact of railway incidents on train delays: A case of the Swedish railway network. *Journal of Rail Transport Planning & Management*, 30, 100445. <https://doi.org/10.1016/j.jrtpm.2024.100445>
- Mullenbach, J., Wiegrefe, S., Duke, J., Sun, J., & Eisenstein, J. (2018, June). Explainable prediction of medical codes from clinical text. In M. Walker, H. Ji, & A. Stent (Eds). *Proceedings of the 2018 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long papers)* (pp. 1101–1111). Association for Computational Linguistics. <https://doi.org/10.18653/v1/N18-1100>
- Müller-Hannemann, M., & Schnee, M. (2009). Efficient timetable information in the presence of delays. In *Robust and Online Large-Scale Optimization: Models and Techniques for Transportation Systems* (pp. 249–272). Berlin, Heidelberg Berlin, Heidelberg: Springer.
- Munro, R. (2021). *Human-in-the-loop machine learning: Active learning and annotation for human-centered AI*. Manning.
- Naghiyev, A. (2017). *Human factors of train driving with in cab control and automation technology* [Unpublished doctoral dissertation]. University of Nottingham.
- Newell, S., & Marabelli, M. (2015). Strategic opportunities (and challenges) of algorithmic decision-making: A call for action on the long-term societal effects of ‘datification’. *Journal of Strategic Information Systems*, 24(1), 3–14. <https://doi.org/10.1016/j.jsis.2015.02.001>
- Nyström, B. (2008). A methodology for measuring the quality of deviation reporting: Applied to railway delay attribution. *International Journal of Quality & Reliability Management*, 25(7), 656–673. <https://doi.org/10.1108/02656710810890863>
- Palmqvist, C.-W., Lind, A., & Ahlqvist, V. (2022). How and why freight trains deviate from the timetable: Evidence from Sweden. *IEEE Open Journal of Intelligent Transportation Systems*, 3, 210–221. <https://doi.org/10.1109/OJITS.2022.3160546>
- Palmqvist, C. W., Olsson, N., & Hiselius, L. (2017). Some influencing factors for passenger train punctuality in Sweden. *International Journal of Prognostics and Health Management*, 8(3). <https://doi.org/10.36001/ijphm.2017.v8i3.2649>
- Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 30(3), 286–297. <https://doi.org/10.1109/3468.844354>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12(8), 2825–2830. <https://doi.org/10.5555/1953048.2078195>
- Perotte, A., Pivovarov, R., Natarajan, K., Weiskopf, N., Wood, F., & Elhadad, N. (2014). Diagnosis code assignment: Models and evaluation metrics. *Journal of the American Medical Informatics Association*, 21(2), 231–237. <https://doi.org/10.1136/amiajnl-2013-002159>
- Power, D. J. (2002). Decision support systems: concepts and resources for managers. *Studies in Informatics and Control*, 11(4), 349–350.
- Punzi, C., Setzu, M., Pellungrini, R., Giannotti, F., & Pedreschi, D. (2023). Towards synergistic human-AI collaboration in hybrid decision-making systems. In Meo R. & Silvestri F. (Eds.), *Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, Springer, Cham. (pp. 268–275 https://doi.org/10.1007/978-3-031-74627-7_20)

- Remmer, S., Lamproudis, A., & Dalianis, H. (2021). Multi-label diagnosis classification of Swedish discharge summaries-ICD-10 code assignment using KB-BERT. In *Proceedings of the international conference on recent advances in natural language processing (ranlp 2021)* (pp. 1158–1166).
- Rezaeian, O., Bayrak, A. E., & Asan, O. (2025). Explainability and AI confidence in clinical decision support systems: Effects on trust, diagnostic performance, and cognitive load in breast cancer care. *International Journal of Human-Computer Interaction*, 1–21. <https://doi.org/10.1080/10447318.2025.2539458>
- Russell, S. J., & Norvig, P. (2010). *Artificial intelligence a modern approach*. Upper Saddle River, EUA: Prentice-Hall.
- Sadeek, S. N., & Hanaoka, S. (2023). Assessment of text-generated supply chain risks considering news and social media during disruptive events. *Social Network Analysis and Mining*, 13(1), 96. <https://doi.org/10.1007/s13278-023-01100-0>
- Saward, J. R., & Stanton, N. A. (2018). Individual latent error detection: Simply stop, look and listen. *Safety Science*, 101, 305–312. <https://doi.org/10.1016/j.ssci.2017.09.023>
- Schmager, S., Pappas, I., & Vassilakopoulou, P. (2023). Defining human-centered AI: A comprehensive review of HCAI literature. In *Mediterranean conference on information systems (mcis)*. <https://aisel.aisnet.org/mcis2023/13>
- Schöbel, A. (2009). Capacity constraints in delay management. *Public Transport*, 1(2), 135–154. <https://doi.org/10.1007/s12469-009-0010-0>
- Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys (CSUR)*, 34(1), 1–47. <https://doi.org/10.1145/505282.505283>
- Sheridan, T. B., Sheridan, T. B., & Maschienenbauingenieur, K., Sheridan, T. B., & Sheridan, T. B. (2002). *Humans and automation: System design and research issues* (Vol. 280). Human Factors and Ergonomics Society.
- Sheskin, D. J. (2003). *Handbook of parametric and nonparametric statistical procedures*. Chapman and hall/CRC.
- Shim, J. P., Warkentin, M., Courtney, J. F., Power, D. J., Sharda, R., & Carlsson, C. (2002). Past, present, and future of decision support technology. *Decision Support Systems*, 33(2), 111–126. [https://doi.org/10.1016/S0167-9236\(01\)00139-7](https://doi.org/10.1016/S0167-9236(01)00139-7)
- Shollo, A., Constantiou, I., & Kreiner, K. (2015). The interplay between evidence and judgment in the IT project prioritization process. *Journal of Strategic Information Systems*, 24(3), 171–188. <https://doi.org/10.1016/j.jsis.2015.06.001>
- Silver, M. S. (1991). Decisional guidance for computer-based decision support. *MIS Quarterly*, 15(1), 105–122. <https://doi.org/10.2307/249441>
- Simon, H. (2000). Theories of bounded rationality. *Central Currents in Social Theory: The Roots of Sociological Theory*, 1700–1920.
- Simon, H. A. (1990). Invariants of human behavior. *Annual Review of Psychology*, 41(1), 1–20. <https://doi.org/10.1146/annurev.ps.41.020190.000245>
- Sotala, K. (2012). Advantages of artificial intelligences, uploads, and digital minds. *International Journal of Machine Consciousness*, 4(1), 275–291. <https://doi.org/10.1142/S1793843012400161>
- Spanninger, T., Trivella, A., Büchel, B., & Corman, F. (2022). A review of train delay prediction approaches. *Journal of Rail Transport Planning & Management*, 22, 100312. <https://doi.org/10.1016/j.jrtpm.2022.100312>
- Speier, C., Vessey, I., & Valacich, J. S. (2003). The effects of interruptions, task complexity, and information presentation on computer-supported decision-making performance. *Decision Sciences*, 34(4), 771–797. <https://doi.org/10.1111/j.1540-5414.2003.02292.x>
- Spitz-Oener, A. (2006). Technical change, job tasks, and rising educational demands: Looking outside the wage structure. *Journal of Labor Economics*, 24(2), 235–270. <https://doi.org/10.1086/499972>
- Sun, A., & Lim, E.-P. (2001). Hierarchical text classification and evaluation. *IEEE International Conference on Data Mining*. 29 November–2 December 2001, San Jose, California. https://ink.library.smu.edu.sg/sis_research/976
- Tiong, K. Y., Ma, Z., & Palmqvist, C.-W. (2023). A review of data-driven approaches to predict train delays. *Transportation Research Part C: Emerging Technologies*, 148, 104027. <https://doi.org/10.1016/j.trc.2023.104027>
- Trafikverket. (2020). En punktligare tågtrafik - sammanställning av trafikverkets åtgärder 2017–2019. Rapport, 2020. (179). <https://data.riksdagen.se/fil/94561D44-1FA7-450D-9AA6-52D716F40A52>
- Trafikverket. (2023). Järnvägsnätsbeskrivningen. https://bransch.trafikverket.se/contentassets/a7154cd2d5ca423dacb011e0da75b673/jnb_2023_2023-12-15-am15.pdf
- Van den Broek, E., Sergeeva, A., & Huysman, M. (2021). When the machine meets the expert: An ethnography of developing AI for hiring. *MIS Quarterly*, 45(3), 1557–1580. <https://doi.org/10.25300/MISQ/2021/16559>
- Veiseth, M., Olsson, N., & Saetermo, I. A. F. (2007). Infrastructure's influence on rail punctuality. *WIT Transactions on the Built Environment*, 96, 481–491. <https://doi.org/10.2495/UT070451>
- Vovk, V., Gammerman, A., & Shafer, G. (2005). *Algorithmic learning in a random world* (Vol. 29). Springer.
- Wagner, B. (2019). Liable, but not in control? Ensuring meaningful human agency in automated decision-making systems. *Policy & Internet*, 11(1), 104–122. <https://doi.org/10.1002/poi3.198>
- Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, 53(3), 1–34. <https://doi.org/10.1145/3386252>
- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., & Rush, A. M. (2020, October). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: System demonstrations* (pp. 38–45). Association for Computational Linguistics. Online. <https://www.aclweb.org/anthology/2020.emnlp-demos.6>
- Wu, S. J., Melnyk, S. A., & Flynn, B. B. (2010). Operational capabilities: The secret ingredient. *Decision Sciences*, 41(4), 721–754. <https://doi.org/10.1111/j.1540-5915.2010.00294.x>
- Zhou, P., Chen, L., Dai, X., Li, B., & Chai, T. (2021). Intelligent prediction of train delay changes and propagation using RVFLNS with improved transfer learning and ensemble learning. *IEEE Transactions on Intelligent Transportation Systems*, 22(12), 7432–7444. <https://doi.org/10.1109/TITS.2020.3002785>