Predicting Delay Occurrence at Freight Rail Sidings

Juan Carlos Martínez Mori
Department of Civil and Environmental Engineering
University of Illinois at Urbana-Champaign
jcmarti2@illinois.edu

William Barbour
Department of Civil and Environmental Engineering
University of Illinois at Urbana-Champaign

Shankara Kuppa
CSX Transportation

Daniel B. Work (corresponding author)
Department of Civil and Environmental Engineering and Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
dbwork@illinois.edu

Abstract

Human dispatchers make freight rail dispatch decisions in real-time based on factors such as network traffic, network topology, and train characteristics. These decisions have significant influence on train delays and rail capacity, which motivates the development of tools to predict their effects. This article presents a machine learning framework to predict the occurrence of delay-inducing meets at sidings using an encoding of network state that incorporates information available to dispatchers at the time of prediction. Support vector classifiers (SVCs) are trained and predictions are compared to a simple deterministic baseline technique that uses only location information and treats trains equally. Testing is performed using historical data from a rail network in Tennessee, USA. Preliminary findings indicate that SVCs are able to exploit critical information beyond just train locations that is present in the network state to predict the occurrence of delays at sidings. The SVCs far outperform the baseline technique to which we compare and show that factors such as train length, train priority, and track occupancy influence delay occurrence in the case study.

1 Introduction

1.1 Motivation and related work

The demand for freight rail transportation in the United States is expected to grow significantly in the coming years, leading to long, heavy trains and increased network congestion (*1, 2*). To provide adequate service, it will be critical for railroad operators to make the most of their infrastructure capacity. Therefore, quantifying the influence of the rail network characteristics, the presence of heterogeneous traffic, the design of train schedules, and the incidence of disturbances are active areas of research (*3, 4, 5, 6*).

In the United States, railroads are primarily composed of single track segments that serve bidirectional traffic. Short areas of double track segments called *sidings* are used to resolve conflicts between trains traveling in opposing directions, called *meets*, or to allow *overtakes* between trains traveling in the same direction. Ideally, conflicting trains could meet at sidings according to a predetermined optimal schedule and experience globally minimized delay. However, operational constraints and unforeseen disturbances such as delays from the train yards, mechanical failure, unplanned work, and inclement weather inevitably cause the system to deviate from an original plan. Consequently, human dispatchers, in some cases aided by computers, are used to make dispatch decisions involving sidings in real-time. This real-time dispatching can result in *delay-inducing meets* between trains, which we define to incur a delay longer than a specified threshold.

The decisions of which trains are deviated to a siding, as well as for how long they are delayed, are largely based on dispatcher judgment and depend on a variety of network state factors. These include the topology of the network and the state of the traffic. Furthermore, these decisions are not isolated, as they may influence network capacity (*7*) and cause knock-on delays that propagate through the network (*8, 9, 10*).

Extensive research has been done to develop decision support systems for real-time dispatching (*11, 12, 13, 14, 15, 16, 17*). However, the dispatch strategy that is used in practice may vary over time and by location. Further, dispatchers may deviate from the plan or computer recommendation to account for circumstances not considered by the decision support system. Therefore, predicting whether a given train will experience delay often involves understanding the dispatching strategy that is used and the human dispatcher following such strategy strictly.

The most closely related work is on data driven methods that predict train event times (*18*), delays at stations (*19, 20*), and overall runtimes (*21*). In this article, we instead use a data driven technique to explore the factors that produce delays at sidings. Given a rail network state, such tool can predict whether a train will experience delay at a given siding, regardless of the dispatch strategy used. This quantifies the effects of dispatch decisions and could be used, for example, to improve real-time arrival predictions or to predict grade crossing blockages (e.g., if a street intersects a railroad at a siding). Furthermore, trained models could be used to detect unusual behavior or to gain intuition on the factors that drive delays at sidings.

1.2 Contributions and outline

The main contribution of this article is the development of a machine learning classification framework to predict whether a train will experience a delay-inducing meet at a given siding, greater than a given threshold. This classification of delay occurrence is a simplified precursor to the more challenging problem of delay estimation, which depends on externalities not explainable by attributes captured in our current dataset. For example, the delay magnitude is in some cases skewed towards very high values by rare events such as mechanical failures and additional crew calls.

The remainder of this work is organized as follows. In the next section we explain our methodology by formulating the train meet problem, summarizing the predictors, and defining the performance metrics. Then, we test the solution approach using historical data from the CSX Transportation Nashville division, evaluate its preliminary performance, and interpret the model. Lastly, we draw conclusions and discuss future research possibilities.

2 Methodology

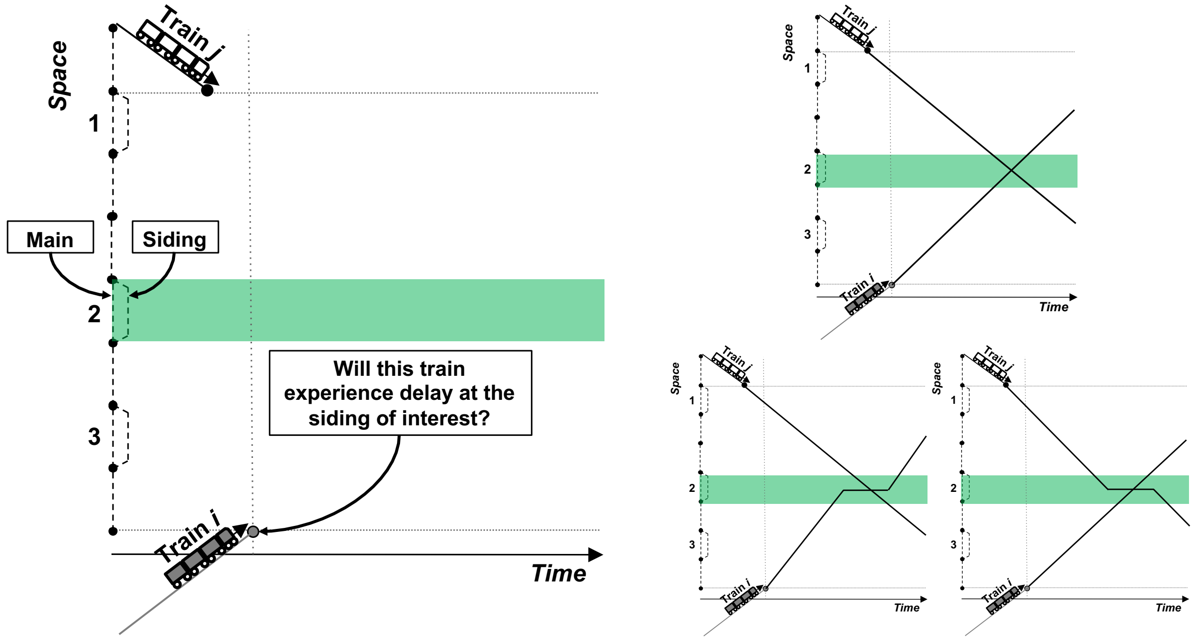
2.1 Problem formulation

Our goal is to predict whether a train will experience a delay-inducing meet at a given siding through a prediction framework that captures the essential aspects of the network state. Well in advance of a train i reaching a particular siding of interest, the dispatcher must determine a strategy. Conflicts may arise as in Figure 1, where train i (grey) and train j (white) approach the siding of interest (#2, shaded). Alternatives include controlling train speeds to perform a delay-free meet at siding #2 (if possible), delaying one of the trains at the siding, or resolving the conflict elsewhere (e.g., sidings #1 or #3). Decisions are based on a variety of factors including track occupancy, network topology, and train characteristics such as length and priority.

In this work, trains report their position at discrete locations along the track called *on-station points* (OS-points) and marked as black dots in Figure 1. OS-points are located almost exclusively at the beginning and ends of sidings. For this reason, we say train i follows a discretized trajectory $T_i = [(d_{i,1}, t_{i,1}), (d_{i,2}, t_{i,2}), \dots, (d_{i,k}, t_{i,k}), \dots]$, where $d_{i,k}$ is the location of the kth trajectory point at time $t_{i,k}$. We consider a *segment* to be comprised of the track between a pair of adjacent OS-points. Note that under this definition, a segment may include both the main-line track and the siding, if present.

We consider the network state at any given point in time to be the occupancy of individual segments, along with the characteristics of the train on the segment if it is occupied. The network state with respect to segment occupancy is extracted from the train trajectories and the occupying train characteristics are determined from a database of train operations. We limit our analysis to the cases where a single train meet is caused by two trains traveling in opposing directions. This is because the cases without train conflicts are trivial, as the candidate train is highly unlikely to be stopped at the siding and experience delay. Cases involving overtakes or more than one conflict resolved at a single siding occur less frequently and do not constitute sufficient data for the training process.

In practice, dispatchers solve line of road conflicts by planning over some reasonable spatial horizon. To model this, we define a topological *visibility window* comprised of the segment containing the siding of interest, u segments upstream, and v segments downstream, totalling $u + 1 + v$ segments. For example, if the dispatcher makes decisions using only the network shown in Figure 1, the visibility window would have $u = 3$ and $v = 4$, and would include sidings #1, #2, and #3.



(a) Problem formulation.

(b) Three possible solutions.

FIGURE 1 : (a) Potential scenario faced by train dispatchers. (b) A dispatcher may try to control train speeds to have a delay-free meet (top), or may have to delay either train *i* (bottom left) or train *j* (bottom right). Alternatively, a dispatcher may opt to solve the conflict elsewhere.

Dispatchers make decisions on train meets well in advance of the actual occurrence and give track clearance when a train is several segments away from the siding. To model this, we make the prediction of delay occurrence for train *i* at τ minutes prior to the true time, t_{i,k^*}, at which train *i* reaches the siding of interest k^*. This results in the prediction for train *i* being made based on the network state at clock time $t_{i,k^*} - \tau$, mimicking an online operational tool without knowledge of the future system state.

The distribution of travel times across a segment that contains a siding is typically narrow around the statistical mode. Inevitably, some trains will experience delays and, consequently, the distribution has a long tail of high travel times (i.e., it is skewed to the right). In this article, we say that a train experienced delay at a siding if it took time greater than Δ minutes to traverse it. The threshold Δ for a particular siding is set to $\Delta = Q_3 + 1.5 \times IQR$, where Q_3 is the third quartile and IQR is the interquartile range of historical travel times. That is, we determine the occurence of delay based on the difference between a train's runtime across a segment and the typical running time of all trains across the segment.

We treat delay occurrence as a classification problem with class label $y = 1$ if a train experiences a delay at a siding of interest, and $y = -1$ otherwise. Given mined and preprocessed historical data containing train trajectories, train characteristics, and the true labels of delay occurrence, we test two classifiers, both subject to the visibility window and the time offset. Specifically, we consider *i*) a naive delay minimizing baseline classifier, and *ii*) a *support vector classifier* (SVC) that learns from observations that incorporate more comprehensive network state information.

2.2 Summary of predictors

2.2.1 Baseline classifier

The baseline classifier is a heuristic technique that considers only the location and directionality of trains from the network state. The baseline is a deterministic algorithm and is intentionally simplistic such that it neglects the additional factors that we use in the SVC and treats all trains equally. The location and direction of candidate train i and any other trains in the visibility window are determined at time $t_{i,k^*} - \tau$ from their respective trajectories. We consider the cases with a single opposing train j in the visibility window at the time of prediction.

First, the classifier determines if the conflict is to be resolved at the siding of interest k^* based on the nearest siding to a linearly interpolated meeting point. If this is not the case, the output is $\hat{y} = -1$ (no delay occurs at the siding), as the conflict is resolved elsewhere. Otherwise, it estimates which of the trains i or j arrives at k^* first based on the current location of each train. The train that is estimated to arrive first is assumed to experience delay at the siding. If the candidate train is to arrive first, the output is $\hat{y} = 1$ (i.e., it experiences delay), otherwise it is $\hat{y} = -1$ (i.e., it does not experience delay).

The baseline is delay minimizing under the assumption that the opposing trains run at the same constant speed. This assumption is made because instantaneous speed data is not reported at the OS-points used in this research and train maximum speeds are similar on this portion of the network. Additionally, because the baseline intentionally ignores the other factors we know the railroad uses to make decisions, the predictive impact of these factors is present in the performance gains achieved by the SVC over the baseline.

2.2.2 Support vector classifier

Clearly a better rule based baseline could be developed, at a substantial cost in model complexity. The data driven approach (SVC) attempts to learn these rules implicitly from the data, simplifying the model construction. We present this method by first describing the construction of the required data set and then providing a brief overview of the algorithm.

We use a training data set $D_{tr} = \{X_{tr}, Y_{tr}\}$, where $X_{tr} := [x_1, \ldots, x_i, \ldots, x_m]$ is a constructed set of m training feature vectors $x_i \in \mathbb{R}^n$, and $Y_{tr} := [y_1, \cdots, y_i, \cdots, y_m]^\top$ is a vector of m true delay-inducing meet labels $y_i \in \{1, -1\}$ corresponding to the vectors in X_{tr}. For each candidate train $i = 1, \ldots, m$, we obtain the true label y_i directly from the data set and encode the system state in a feature vector x_i as shown in Figure 2. In this work, each segment-wise partition of a feature vector x_i encodes train characteristics by including the five attributes described in Table 1. Any segment without a train present will have its feature vector partition set to zeroes. Feature vectors are scaled identically using element-wise min-max normalization.

The feature occupancy (O) is associated with each segment, where the segment is said to be occupied if a train is present and unoccupied otherwise. The feature relative direction (D) is associated with an occupied segment, and enables the algorithm to locate the candidate and conflicting train within the visibility window. The features priority (P), train length (L), and crew time remaining (R) are considered relevant train attributes that may influence a dispatcher's decision. For example, given two conflicting trains of different priority levels, a dispatcher may prefer to delay the train with the lowest priority, even if this choice is not delay minimizing. Further, a dispatcher's decision may be constrained by physical attributes such as a train being too long to

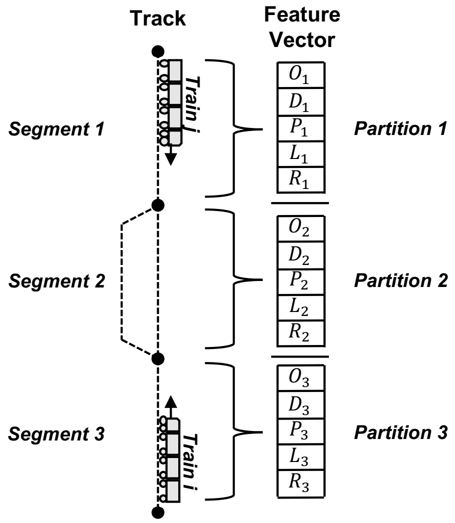


FIGURE 2 : A feature vector is composed of partitions that correspond to track segments in the visibility window. Each cell in the feature vector corresponds to a feature described in Table 1. Each feature vector has $u + 1 + v$ partitions.

fit in a particular siding. Lastly, a dispatcher's decision may also be constrained by regulations that impose a limit on the amount of time for which a crew may work. That is, a train with a crew whose legal work time limit is about to expire is likely to be sent to a siding in order to wait for a relief crew or to be expedited to avoid needing a relief crew.

The features summarized in Table 1 are by no means an exhaustive encoding of the network state, but are selected after consideration of their contribution to the predictive capability of the proposed model. Considering that each additional feature category adds $u + 1 + v$ dimensions to the overall feature vector, we are cautious not to unnecessarily increase the dimensionality of the feature space.

Given the training data set, we calibrate a *support vector classifier* (SVC) (*22*). For completeness, we summarize the foundation of this algorithm. Readers looking for a comprehensive explanation of SVCs are directed to (*23*).

The goal of training a *hard margin* SVC is to obtain the maximum margin hyperplane that separates data points of one class from data points of another, provided the data is linearly separable. Meanwhile, a *soft margin* SVC allows some classification error by the hyperplane, which is necessary in the more common case where the true labels are not linearly separable. The soft margin classifier is trained by solving the primal problem on the training data:

$$\begin{aligned} \underset{\omega, b, \zeta}{\text{minimize}} \quad & \frac{1}{2}\|\omega\|_2^2 + C\sum_{i=1}^{m}\zeta_i \\ \text{subject to} \quad & y_i(\omega^T x_i + b) \geq 1 - \zeta_i \\ & \zeta_i \geq 0,\ i = 1, \dots, m, \end{aligned} \tag{1}$$

where ω is a vector of weights that is perpendicular to the hyperplane, b is a bias term, and

TABLE 1 : Features in segment-wise partitions of a feature vector.

Feature	Description
Occupancy (O)	Set to 0.5 if occupied by traffic, set to 1.0 if occupied by candidate train.
Relative direction (D)	Set to 0.5 if occupying train runs on the same direction as the candidate train, set to 1.0 if occupying train runs on the opposite direction to the candidate train.
Priority (P)	Priority ranking on a 1-4 scale based on train type.
Train length (L)	Total length of locomotive and cars.
Crew time remaining (R)	Legal amount of work time remaining for train crew.

$\zeta_i,\ i = 1,\dots,m$ are slack variables that penalize incorrect classification by a factor of C. The purpose of the penalty factor C is to balance model fit and model complexity quantified by $\|\omega\|_2^2$. The problem's dual form is

$$\begin{aligned} \underset{\alpha}{\text{minimize}} \quad & \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j x_i^T x_j - \sum_{i=1}^{m}\alpha_i \\ \text{subject to} \quad & \sum_{i=1}^{m}\alpha_i y_i = 0 \\ & 0 \le \alpha_i \le C,\ i = 1,\dots,m, \end{aligned} \tag{2}$$

where $\alpha_i,\ i = 1,\dots,m$, are the dual variables. This formulation rests on the *Karush-Kuhn-Tucker* (KKT) conditions for the Lagrangian formulation of the primal problem, which also imply that

$$\omega = \sum_{i=1}^{m}\alpha_i y_i x_i, \tag{3}$$

and that α_i is nonzero only when x_i is a *support vector*. The bias term b is implicitly determined and can also be found with the KKT conditions. In practice, the number of support vectors is typically much smaller than m, reducing the number of operations needed to compute ω. Once a solution is found, the decision function is

$$f(x) = \omega^T x + b, \tag{4}$$

$$= \sum_{i=1}^{m}\alpha_i y_i x_i^T x + b. \tag{5}$$

Class label predictions $\hat{y}_i$ are assigned according to a discrimination threshold h, which is a free parameter tuning the model towards desirable application-specific results. An observation x_i is labeled as 1 if $f(x_i) > h$ and as -1 otherwise. In the case of predicting train delay at sidings in practice, it would likely be preferable to overestimate delay in lieu of underestimating it and choosing h to shift the classification boundary provides for this flexibility. Note that this is unrelated to the delay threshold parameter Δ described in the problem formulation.

It is often the case that the vectors $x_i \in X_{tr}$ are not linearly separable in $\mathbb{R}^n$. In such cases, we can use a decision function of the form

$$f(x) = \sum_{i=1}^{l} \alpha_i y_i \Phi(x_i)^T \Phi(x) + b, \tag{6}$$

where the function $\Phi : \mathbb{R}^n \longrightarrow \mathbb{R}^N$ takes a feature vector of n dimensions to a higher dimensional space of N dimensions, for some $N >> n$, where the class labels are linearly separable (i.e., the hard margin case) or can be linearly separated with fewer misclassification errors (i.e., the soft margin case). Note that we are only interested in the result of the dot product $\Phi(x_i)^T \Phi(x)$, rather than on an explicit definition of Φ. Therefore, we can instead use a kernel function such that $K(x_i, x) = \Phi(x_i)^T \Phi(x)$ to obtain

$$f(x) = \sum_{i=1}^{l} y_i \alpha_i K(x_i, x) + b. \tag{7}$$

This technique is commonly referred to as the *kernel trick* and has been widely studied (*24*). In the linear case, the kernel function is simply $K(x_i, x) = x_i^T x$. An example of a nonlinear kernel is the *radial basis function* (RBF) kernel

$$K(x_i, x) = \exp\left(-\frac{\|x_i - x\|^2}{2\sigma^2}\right), \tag{8}$$

where σ is a free parameter that controls the decay rate.

2.3 Performance metrics

A train that is delayed at a siding ($y = 1$) belongs to the *positive* labels (P). Similarly, a train that is not delayed at a siding ($y = -1$) belongs to the *negative* labels (N). It follows that the classification output $\hat{y}$ belongs to one of four categories. If a positive instance is correctly labeled as positive, the output belongs to the *true positives* (TP). On the other hand, if a positive instance is incorrectly labeled as negative, the output belongs to the *false negatives* (FN). Likewise, if a negative instance is correctly labeled as negative, the output belongs to the *true negatives* (TN). Lastly, if a negative instance is incorrectly labeled as positive, the output belongs to the *false positives* (FP). We now introduce the performance metrics based on the defined categories.

Many meets can be set up to occur in synchronization and incur very little delay, especially when the network is less congested. It follows that class labels as defined by our delay threshold are typically unbalanced; most class labels are $y = -1$. This indicates that simple performance metrics such as accuracy will not be very informative; good performance in the more populated class may mask under-performance in the less populated class. Therefore, we opt to use the *true positive rate* (TPR) and *false positive rate* (FPR) as our performance metrics, as they are insensitive to class imbalances. These are defined as

$$\text{TPR} = \frac{\text{TP}}{\text{P}}, \tag{9}$$

$$\text{FPR} = \frac{\text{FP}}{\text{N}}. \tag{10}$$

In this context, TPR is the fraction of trains that were correctly predicted to be delayed at the siding from the total number of trains that were truly delayed at the siding. Meanwhile, FPR is the fraction of trains that were incorrectly predicted to be delayed at the siding from the total number of trains that were truly not delayed at the siding. These two metrics indicate how the positive predictions (TP and FP) are distributed among the true labels (P and N).

We are also interested in measuring the relevance of the predictions made. Therefore, we consider the *precision* and *recall* statistics, defined as

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}, \tag{11}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}. \tag{12}$$

In this context, precision is the fraction of trains that were correctly predicted to be delayed at the siding from the total number of trains that were predicted to be delayed at the siding. Note that recall is the same as TPR, since TP + FN = P. Thus, we can think of precision as a measure of prediction quality and of recall as a measure of prediction completeness within the positive class.

The baseline predictor will result in one value for each of the performance metrics outlined because it does not use the discrimination threshold parameter h. The SVC can produce various values for each of the performance metrics outlined as h is varied and shifts the decision boundary. This results in a *receiver operating characteristic* (ROC) curve as TPR and FPR respond to h as well as a precision-recall curve as the precision and recall respond. The appropriate value for h depends on the application of the classifier, and should be chosen based on curves generated with the training data. For example, road safety applications at grade crossings need to achieve low FN and high TPR to avoid cases where delays are unforseen.

Note that a perfect predictor would have TPR = 1, FPR = 0, precision = 1, and recall = 1.

ROC and precision-recall curves can be used to identify suboptimal models (when a model's curve is dominated by another model's curve). Further, the *area under the curve* (AUC) can be used to compare models without fixing the discrimination threshold h. A perfect predictor would achieve a ROC AUC = 1 and precision-recall AUC = 1, so AUC values that approach one are desirable.

Since the baseline will produce a single point in the FPR-TPR plot, we say that the baseline is outperformed if a model matches its FPR at a higher TPR (or conversely, matches its TPR at a lower FPR). Likewise in the precision-recall plot, we say that the baseline is outperformed if a model matches its precision at a higher recall (or conversely, matches its recall at a higher precision).

3 Results and evaluation

3.1 Case study description

We evaluate the baseline and the proposed methodology using historical data from the CSX Transportation Nashville division. Specifically, we predict the occurrence of delay-inducing meets at a siding in the Chattanooga subdivision in Tennessee. We consider a data set filtered for the cases of strictly pairs of opposing trains in the entire visibility window, resulting in almost 1000 cases from historical data. We define a visibility window with parameters $u = 5$, and $v = 6$, as schematically represented in Figure 3. This window includes sixty features, one siding north of the siding of interest (siding #3-S), and three sidings south of the siding of interest (sidings #8-S, #10-S, and #11-S). These parameters were chosen in order to have a reasonably large visibility window without reaching the *yards* at the ends of the line, where the boundary conditions are unavailable due to varying departure times.

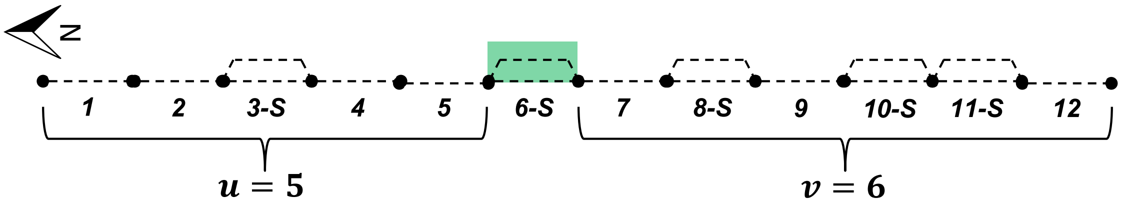


FIGURE 3 : Schematic representation of the network near the siding of interest (#6-S, shaded). There are 5 + 1 + 6 = 12 segments in the visibility window, resulting in $12 \cdot 5 = 60$ features.

In order to account for possible traffic heterogeneity by direction, as well imbalance caused by the network topology, distinct baseline classifiers and SVCs are considered for each combination of prediction direction and τ. From this data set, we calculate $\Delta = 8$ minutes from the travel time distribution.

We evaluate the performance of the predictors on the testing data at time offset τ of ten, twenty, and thirty minutes. In the case of the SVCs, seventy percent of the data is used for training, $D_{tr} = \{X_{tr}, Y_{tr}\}$, and thirty percent of the data is held out for testing, $D_{ts} = \{X_{ts}, Y_{ts}\}$. The SVC parameters ($C$ and σ) for each of these combinations is obtained through a parameter grid search using 10-fold cross validation on the training data, with the objective of maximizing ROC AUC.

3.2 Preliminary results

Preliminary performance of the delay occurrence predictors using the testing data with $\tau = 20$ minutes and $h = 0$ (in the case of the SVCs) is presented in Table 2. We see that the baseline predictor tends to have a low TPR (recall), low precision, and high FPR. This is undesirable, as it means that it incorrectly predicts the occurrence of delays at siding for a large fraction of the trains that did not actually experience it, while it correctly predicts the occurrence of delays at siding for only a small fraction of the trains that actually experienced it. On the other hand, both the linear SVC and the RBF SVC far outperform the baseline for both northbound and southbound directions. Overall, this supports the idea that the siding delay result is the combination of many factors not captured by the simplistic baseline. We also see that in general, the SVCs achieve a better performance on the northbound predictions.

TABLE 2 : Predictor performance with $\tau = 20$ minutes and $h = 0$.

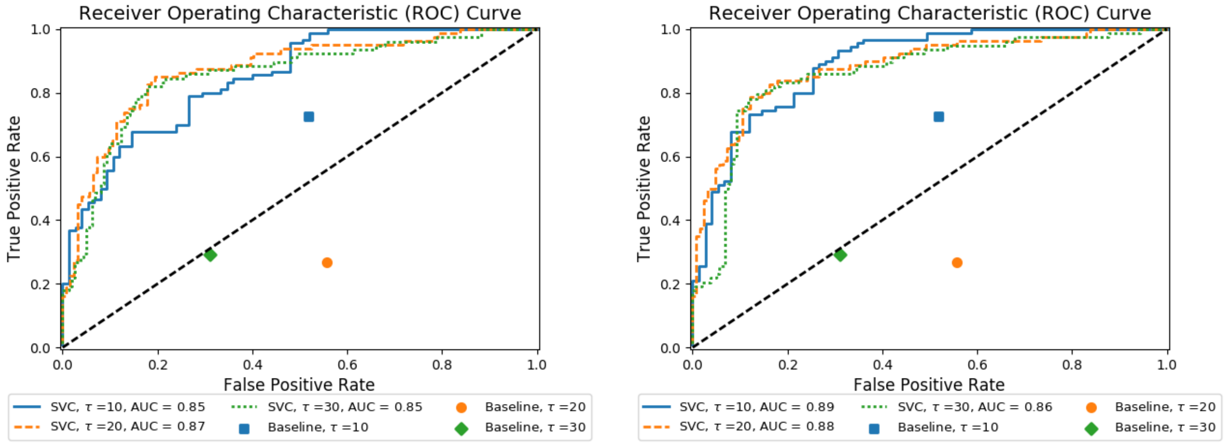
Direction	Predictor	TPR	FPR	Precision	Recall
Northbound	Baseline	0.27	0.56	0.10	0.27
	Linear SVC	0.85	0.22	0.72	0.85
	RBF SVC	0.84	0.19	0.74	0.84
Southbound	Baseline	0.27	0.23	0.10	0.27
	Linear SVC	0.72	0.32	0.41	0.72
	RBF SVC	0.71	0.27	0.45	0.71

A more comprehensive comparison of the predictors is facilitated by Figure 4 and Figure 5, where ROC curves for the SVC and the baseline are plotted for the northbound and southbound directions, respectively. We see from both figures that the ROC AUC achieved by both the baseline and the SVCs tends to improve as τ decreases. This is expected, as lower values of τ lead to a network state that is an artifact of a dispatching decision that has already been made. However, the SVCs maintain a reasonable performance even at $\tau = 30$ minutes, unlike the baseline. In other words, the SVCs are able to exploit the information in the network state encoding well before the delay at siding event occurs. For instance, in Figure 4a, the $\tau = 10$ minutes baseline achieves a TPR of 0.73 at a FPR greater than 0.5, while the linear northbound SVC with $\tau = 30$ minutes matches that TPR with a FPR lower than 0.2 (over a 60% FPR improvement even with a larger τ). In practice, this means that the SVC is able to match the rate of correct predictions for trains that experienced delay while making less mistakes on the trains that did not experience delay.

The performance of the SVC further improves when a RBF kernel is used, as seen by comparing Figure 4a with Figure 4b, both for northbound trains. A similar tendency can be seen by comparing Figure 5a with Figure 5b for southbound trains. For example, in Figure 5, the ROC AUC of the SVC with $\tau = 30$ minutes improves from 0.71 when the linear kernel is used to 0.76 when the RBF kernel is used, which is a 7.04% ROC AUC increase.

The models can be further evaluated by the precision-recall curves as in Figure 6. We see that the baseline in the northbound direction is outperformed by the SVC predictors at all values of τ tested, which indicates that the SVCs make more relevant siding delay occurrence predictions. For example, in Figure 6a, the $\tau = 10$ minutes baseline achieves a recall of 0.73 at a precision lower than 0.3, while the linear northbound SVC with $\tau = 30$ minutes matches that recall at a precision greater than 0.7 (over a 133% precision improvement even with a larger τ). As before, the SVC further benefits from an RBF kernel, as seen by comparing the ROC AUC of the curves in Figure 6a and Figure 6b. For example, there is a 4.71% precision-recall ROC AUC improvement when the RBF kernel is used with $\tau = 10$ minutes.

Note that the baseline achieves its worst performance in northbound predictions with large values of τ. This can be associated to the increased network complexity south of the siding of interest (as seen in Figure 3). Namely, the baseline has trouble identifying the correct siding to be used. On the other hand, relevant information is available to the SVCs through the network state encoding, allowing it to determine at which siding the conflict is resolved and whether delays occur.



(a) Linear kernel.

(b) RBF kernel.

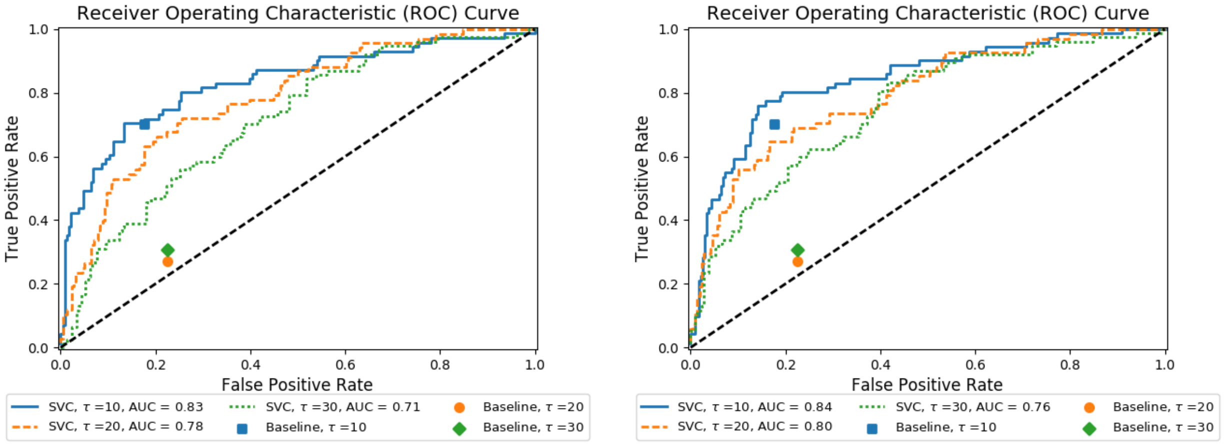
FIGURE 4 : ROC curves for northbound predictions at different values of τ and using the baseline predictor and SVC with linear kernel (a) and RBF kernel (b).

3.3 Model interpretation

One advantage of using a linear SVC is that its weight vector ω (Equation 3) can be interpreted in the context of this case study and the choice of τ. Namely, features with a large absolute weight are viewed as influential on the prediction output. Since the feature vectors in this research are high dimensional, we interpret the model by aggregating the features by *i*) feature type and *ii*) track segment. We follow the segment naming convention used in Figure 3, where siding #6-S is the siding of interest, and normalize the mean absolute feature weights to sum to one. The aggregated feature weights of the northbound and southbound linear SVC predictors with $\tau = 30$ minutes are found in Figure 7.

From Figure 7a we see that the train lengths (L) are the most influential for both the northbound and southbound predictions, followed by the train priorities (P), and track occupancy (O). This is intuitive, as a train being too long to fit in a siding poses additional difficulty to the dispatching decision. Further, the idea that dispatchers consider priority at the time of sending a train to the siding is reinforced. The remaining features, crew time remaining (R) and relative direction (D), although less influential, show non-negligible mean absolute feature weights. This suggests that dispatching decisions account for these attributes simultaneously.

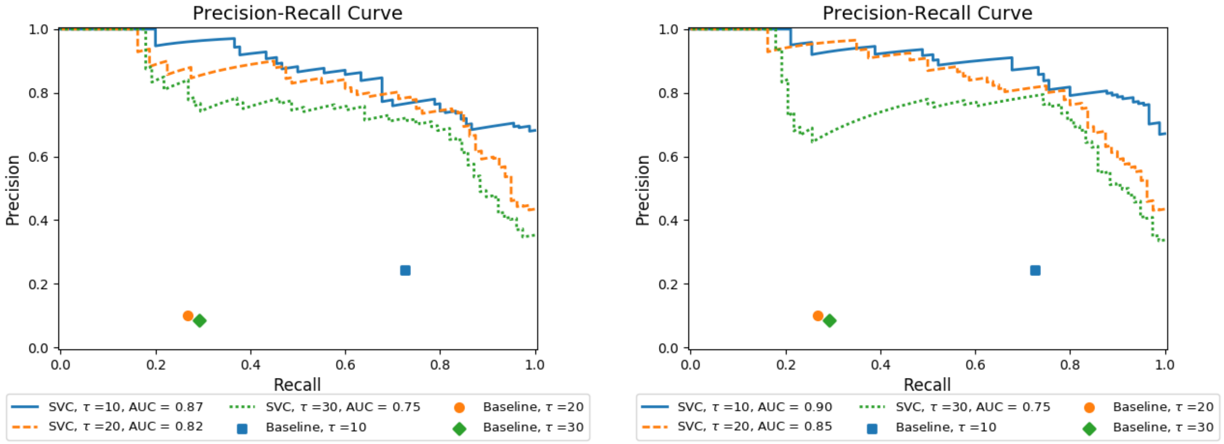
From Figure 7b we see that siding 10-S carries the highest weight, especially for northbound traffic. Further, we see that the track segments south of the siding of interest (segments 7 through 12, three of which are sidings) are in general more influential than the segments north of the siding of interest (segments 1 through 5, one of which is a siding). The least influential track segments in both directions are 4 and 5, which are long segments of main track north of the siding of interest. These results suggest that existing track occupancy is influential in the siding dispatch decision-making process.



(a) Linear kernel.

(b) RBF kernel.

FIGURE 5 : ROC curves for southbound predictions at different values of τ and using the baseline predictor and SVC with linear kernel (a) and RBF kernel (b).



(a) Linear kernel.

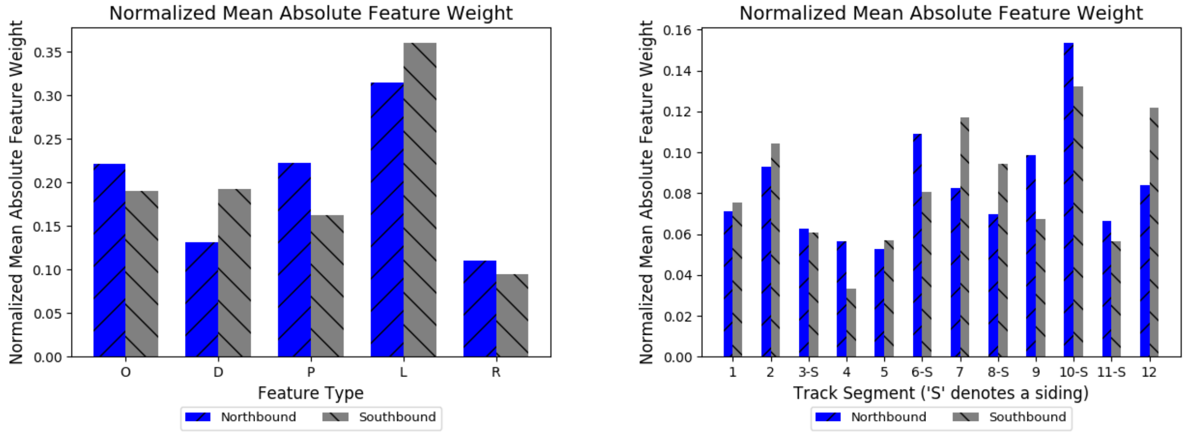
(b) RBF kernel.

FIGURE 6 : Precision-recall curves for northbound predictions at different values of τ and using *i*) baseline, and *ii*) SVC with linear kernel (a) and RBF kernel (b).

4 Conclusions and future work

This research presents a machine learning framework to predict the occurrence of delays at freight rail sidings. Our experimental results indicate that the SVC, trained with our encoding of network state and operating factors, is superior to the heuristic baseline. The outcomes suggest that our encoding of network state, beyond just train locations present in the baseline technique, contains information that is indicative of delay occurrence at sidings.

Ultimately, the selection of a discrimination threshold h for an SVC depends on the valuations of the performance metrics and their relationship to the particular application. Regardless, the RBF SVC achieves the best performance out of the predictors evaluated. However, the linear



(a) Aggregated by feature type across segments.

(b) Aggregated by segment across feature types.

FIGURE 7 : Mean absolute feature weights for linear SVC predictors with $\tau = 30$ minutes, aggregated by (a) feature type, and (b) track segment.

SVC has the advantage over the RBF SVC that it can be interpreted, as we show, with respect to feature weights. Analyzing the feature weights in the linear SVC can help develop a better sense of what factors matter in the context of siding dispatch decisions.

A limitation of this work is that the separation of predictions by direction may produce conflicting results. Future research may include quantifying the level of agreement between prediction directions, as well as proposing resolution schemes in the case of conflicting predictions. Other important research directions include more extensive feature engineering and expanding the problem scope to cases with more than one conflict. We are also interested in extending our findings to predict delay magnitude, as well as to assess the performance of other machine learning algorithms.

Acknowledgments

The authors acknowledge funding by the Roadway Safety Institute, the University Transportation Center for USDOT Region 5. Financial support was provided by the United States Department of Transportation's Office of the Assistant Secretary for Research and Technology (OST-R). The authors also acknowledge CSX Transportation for sharing their expertise and providing the historical data used in this research.

References

[1] AASHTO. Transportation – Invest in our future: America's freight challenge. Technical Report TIF3-1, American Association of State Highway and Transportation Officials, 2007.

[2] FHWA. Freight Story 2008. Technical Report FHWA-HOP-08-051, Federal Highway Administration, 2008.

[3] Monserrat Abril, Federico Barber, Laura Ingolotti, Miguel A. Salido, Pilar Tormos, and Antonio Lova. An assessment of railway capacity. *Transportation Research Part E: Logistics and Transportation Review*, 44(5):774–806, 2008.

[4] Mark H. Dingler, Yung-Cheng Lai, and Christopher P. L. Barkan. Impact of train type heterogeneity on single-track railway capacity. *Transportation Research Record: Journal of the Transportation Research Board*, 2177:41–49, 2009.

[5] Francesco Corman, Andrea D'Ariano, and Ingo A. Hansen. Evaluating disturbance robustness of railway schedules. *Journal of Intelligent Transportation Systems*, 18(1):106–120, 2014.

[6] Ivan Atanassov and C. Tyler Dick. Capacity of single-track railway lines with short sidings to support operation of long freight trains. *Transportation Research Record: Journal of the Transportation Research Board*, 2475:95–101, 2015.

[7] Lars-Göran Mattsson. Railway capacity and train delay relationships. In *Critical Infrastructure: Reliability and Vulnerability*, pages 129–150. Springer Berlin Heidelberg, 2007.

[8] Andrea D'Ariano and Marco Pranzo. An advanced real-time train dispatching system for minimizing the propagation of delays in a dispatching area under severe disturbances. *Networks and Spatial Economics*, 9(1):63–84, 2009.

[9] Rob M.P. Goverde. A delay propagation algorithm for large-scale railway traffic networks. *Transportation Research Part C: Emerging Technologies*, 18(3):269–287, 2010.

[10] Mark Dingler, Amanda Koenig, Sam Sogin, and Christopher P. L. Barkan. Determining the causes of train delay. In *Proceedings of the American Railway Engineering and Maintenance-of-Way Association Annual Conference*, Orlando, 2010.

[11] David R. Kraay and Patrick T. Harker. Real-time scheduling of freight railroads. *Transportation Research Part B: Methodological*, 29(3):213–229, 1995.

[12] Andrew Higgins, Erhan Kozan, and Luis Ferreira. Optimal scheduling of trains on a single line track. *Transportation Research Part B: Methodological*, 30(2):147–161, 1996.

[13] Alexandre Tazoniero, Rodrigo Gonçalves, and Fernando Gomide. Decision making strategies for real-time train dispatch and control. In *Analysis and Design of Intelligent Systems using Soft Computing Techniques*, volume 41, pages 195–204. Springer Berlin Heidelberg, 2007.

[14] Andrea D'Ariano, Marco Pranzo, and Ingo A. Hansen. Conflict resolution and train speed coordination for solving real-time timetable perturbations. *IEEE Transactions on Intelligent Transportation Systems*, 8(2):208–222, 2007.

[15] Andrea D'Ariano, Dario Pacciarelli, and Marco Pranzo. A branch and bound algorithm for scheduling trains in a rail network. *European Journal of Operational Research*, 183:643–657, 2007.

[16] Steven Harrod. A method for robust strategic railway dispatch applied to a single track line. *Transportation Journal*, 52(1):26–51, 2013.

[17] Selim Dündar and İsmail Şahin. Train re-scheduling with genetic algroithms and artificial neural networks for single-track railways. *Transportation Research Part C: Emerging Technologies*, 27:1–15, 2013.

[18] Pavle Kecman and Rob M. P. Goverde. Online data-driven adaptive prediction of train event times. *IEEE Transactions on Intelligent Transportation Systems*, 16(1):465–474, 2015.

[19] Ingo A. Hansen, Rob M.P. Goverde, and Dirk J. van der Meer. Online train delay recognition and running time prediction. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pages 1783–1788, Funchal, Portugal, 2010. IEEE.

[20] Ren Wang and Daniel Work. Data driven approaches for passenger train delay estimation. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pages 535–540, Las Palmas, Spain, 2015. IEEE.

[21] William Barbour, Juan Carlos Martinez Mori, Shankara Kuppa, and Daniel Work. Estimating arrival times for US freight rail traffic. *submitted to Transportation Research Part C: Emerging Technologies*, 2017.

[22] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[23] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[24] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop of Computational Learning Theory*, pages 144–152, Pittsburgh, 1992. ACM.