Remote Sensing for Forest Recovery: Final Report

Benjamin Frizzell Zanan Pech Mavis Wong Hui Tang
Piotr Tompalski Alexi Rodríguez-Arelis

2025-06-24

Table of contents

Executive Summary

This report details the technical framework and principal findings of the MDS Afforestation Monitoring project. The project's central goal was to create and validate a data-driven system for monitoring the progress of afforestation initiatives across Canada.

Afforestation plays a vital role in carbon capture, biodiversity enhancement, and climate resilience, while also benefiting communities by providing green spaces and reducing wildfire and flood risks. To advance these goals, the Canadian government launched the 2 Billion Trees program, aiming to plant two billion trees nationwide over 10 years. However, monitoring the success of such large-scale efforts is challenging, especially in the early stages when young trees are hard to detect with traditional remote sensing due to their sparse canopies. This makes tracking survival rates across Canada's diverse and remote planting sites a complex task for Natural Resources Canada. In this research study, we want to discover whether satellite-derived vegetation indices and site-lvel data will be useful for training predictive models.

Our methodology integrates satellite-derived vegetation indices with site-level data to train predictive models. This curated dataset was used to train and evaluate a suite of machine learning models and a deep learning model aimed at classifying the survival outcomes of newly planted forests. The report provides a comprehensive overview of the entire data science pipeline, from data acquisition and preprocessing to feature engineering and model evaluation.

A comparative analysis of classical machine learning algorithms and a deep learning architecture identified the Random Forest model as the most effective. It achieved an F1-score of approximately 0.52. While this performance does not align with expectations, it establishes a crucial baseline for further scientific research.

The report concludes with actionable recommendations for future work, including the integration of higher-resolution data and more advanced model architectures.

1 Introduction

Afforestation is essential for capturing carbon, enhancing biodiversity, and improving forest resilience to climate change. It also supports human well-being by providing green spaces for nature-based activities, which can improve mental health and reduce the risks of wildfires and floods in communities(S. Canada 2023). Recognizing its importance, the Canadian government launched the 2 Billion Trees program to provide financial support for organizations to plant trees over ten years across Canadian provinces (Natural Resources Canada 2021). However, monitoring the success of large-scale afforestation initiatives remains a critical and complex challenge, particularly during the early stages of growth. Young trees often produce weak spectral signals due to their sparse canopies, making them difficult to detect using traditional remote sensing methods (University of British Columbia Master of Data Science Program 2025). As part of the 2 Billion Trees program—which aims to plant two billion trees across Canada by 2031—Natural Resources Canada must track survival rates across hundreds of ecologically diverse and often remote planting sites (Natural Resources Canada 2021).

In this study, we aim to investigate two main research questions:

- Can satellite-derived vegetation indices and site-level data be used to accurately predict tree survival over time in large-scale afforestation programs?
- Which modelling approach is most effective, and how long after planting is needed before accurate survival predictions can be made?

To address these questions, this study leverages satellite-derived vegetation indices and site-level data to train machine learning models. By evaluating multiple modelling approaches—including logistic regression, random forests, and deep learning models, namely recurrent neural network architectures—we aim to determine which techniques provide the most accurate predictions of tree survival rates to support the mission of sustainable forest management and addressing climate change.

2 Data Science Methods

In this section, we discussed the data science methods used in this project, including data cleaning techniques, classical models, recurrent neural network (RNN) models and evaluation metrics.

2.1 Data Cleaning

Before proceeding with modeling, we performed extensive data cleaning to address some quality issues in the dataset. The preprocessing steps were outlined below:

1. **Records Removal**

   In order to preserve data integrity, we removed the following records from the dataset:

   - **Replanted Sites**: To avoid introducing complex survival dynamics, we removed all records from replanted afforested sites.
   - **Out-of-Range Values**: All records that are outside of the expected range for the spectral indices and survival rates were considered invalid and removed from the dataset.
   - **Missing Spectral Data**: All rows with missing spectral data were removed.
   - **Pre-Plantation Satellite Data**: To avoid introducing noise, satellite records captured before planting were removed, as pre-plantation site conditions are not relevant when modeling afforestation survival rates.

   Since the removed records only accounts for a small proportion of the total records, the impact of this removal on data size and distribution is negligible.

2. **Data Engineering**

   By normalizing tree counts (`Planted`) across site sizes (`Area_ha`), we created a new feature `Density`, which provides a more informative representation of underlying site conditions.

3. **Imputing Species Type**

   The missing values from the `Type` column was imputed from the `SpcsCmp` column. Using the threshold defined in the Forestry Glossary from Natural Resources Canada (2025), sites were labeled as `Conifer` if the proportion of softwood species exceeds 80%, `Deciduous` if hardwood species exceeds 80% and `Mixed` otherwise.

4. **Column Removal**

   To reduce redundancy and ineffectual data, the following columns were dropped:

   - `PlantDt` : This column was dropped since the majority of values in the column were missing (see Figure 1).
   - `NmblR`, `NmblT`, `NmblO`: As we excluded all replanted site records, these columns were no longer useful.
   - `prevUse`: Due to severe class imbalance, this column has limited predictive power.
   - `SpcsCmp` : As the majority of the data does not have any detailed species composition, this column was only used for imputing the species type.
   - `Year` : Both `Year` and `DOY` can be derived from `ImgDate`. The `Year` column was dropped to avoid redundancy with `ImgDate`. `DOY` was retained for seasonality tracking in RNN modeling.
   - `Area_ha`, `Planted` : These two columns were dropped after deriving the new feature `Density` to avoid multicollinearity.

5. **Train Test Split**

   We performed a 70:30 train test split on the processed data, splitting the data by site. This ensures that each site appears only in either the training set or the test set, avoiding data leakage and preserves the temporal structure of the satellite data. While this splitting method can lead to imbalanced splits—especially given the skew toward higher survival rates—it is a necessary trade-off to ensure valid model evaluation and reduce the risk of overfitting.

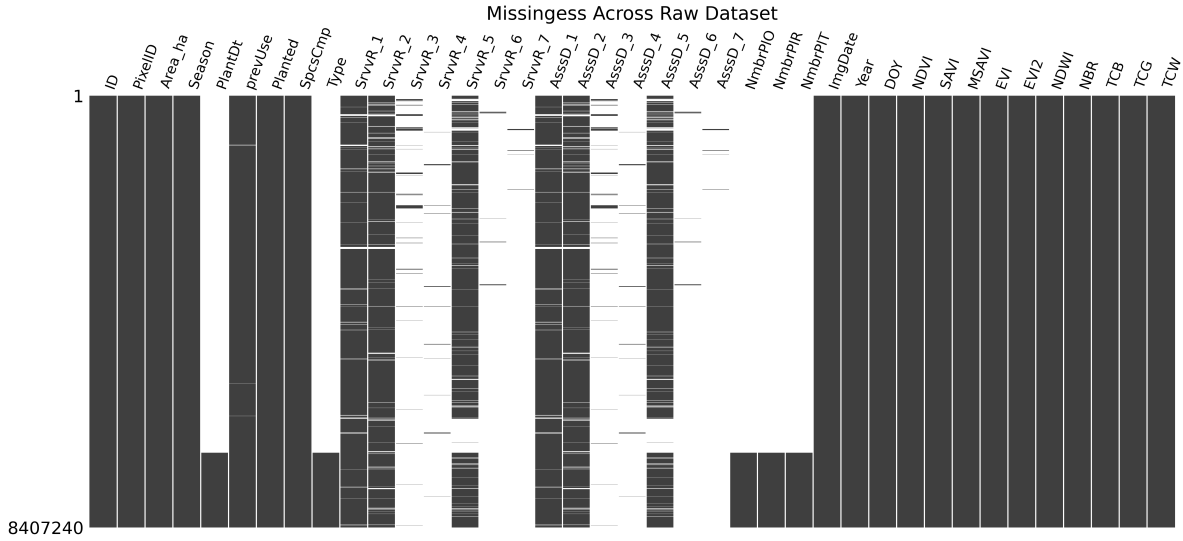

Figure 1: Missingness plot of the raw dataset, where black indicates data presence along rows and white indicates data absent along rows.

2.2 Phase 1: Classical Modelling

In this phase of modelling, we start with three classical machine learning models: Logistic Regression, Random Forest and Gradient Boosting. This section outlines the techniques used in data preparation and classical modelling.

2.2.1 Data Preparation for Classical Models

After data cleaning, we performed data transformation to prepare the cleaned data for classical model training. The preprocessing steps were listed below:

1. **Pivoting survival records**

   We pivot the data to combine the survival rates columns (`SrvvR_1` to `SrvvR_7`) into a single column (`target`), and the survey dates columns (`AsssD_1` to `AsssD_7`) into a

survey date column (`SrvvR_Date`). We added an `Age` column to keep track of the tree's age at the time of the survey.

2. **Satellite-Survey record matching**

   The survival rates data and satellite data were recorded at irregular time intervals. A ± 16 days average spectral signal of the survey date was computed to match the satellite data with the survival rate data. This time window was chosen specifically to match the repeat cycle of the Landsat satellite.

3. **Binary Target Mapping**

   We approached the problem as a classification problem and mapped the `target` (survival rates) into binary classes `Low(0)`/ `High(1)` survival rates based on a given classification threshold.

4. **OneHotEncoding of `Type`**

   While Random Forest and Gradient Boosting models have native support for handling categorical features (Sruthi 2025; Chen and Guestrin 2016), logistic regression models can only handle numeric features (Filho 2023). To maintain consistency, OneHotEncoding (Pedregosa et al. 2011) was applied to the `Type` column for all classical models.

5. **Standard Scaling**

   Since the logistic regression model is also sensitive to the scale of the data (Filho 2023), we applied StandardScaler (Pedregosa et al. 2011) to the numeric features before fitting the logistic regression model.

2.2.2 Logistic Regression

Logistic regression is a generalized linear model widely used for binary classification tasks, valued for its simplicity and interpretability. Recall that the continuous target is converted to a binary classification problem for simplicity; this means that the probability of a particular record belonging to the `High (1)` survival rate class is modelled here. Logistic Regression provides a statistically grounded baseline and serves as a proxy for the classical statistical modeling used prior to this analysis. To demonstrate the value of more sophisticated machine learning models in predicting survival rates, subsequent models were expected to achieve performance exceeding that of logistic regression.

2.2.3 Random Forest

The Random Forest model is an aggregate model composed of many decision trees, each trained on a bootstrapped subset of the training data and a randomly selected subset of the features. Although training Random Forests can be computationally intensive, each tree is trained independently, enabling efficient parallelization and scalability. Previous studies from Bergmüller and Vanderwel (2022) have demonstrated that Random Forests perform well when using vegetation indices to predict canopy tree mortality. Because of this, this model was selected as a candidate for the present analysis, however it was expected that this model may suffer drawbacks as it fails to explicitly capture the temporal sequencing of the data. Additionally, long training times -even with parallelization- made it difficult to finely tune hyperparameters with cross-validation.

2.2.4 Gradient Boosting

The Gradient Boosting model is a popular model that exists in a collection of 'boosting' models, which -unlike Random Forests- consists of a sequence of underfit and biased 'weak learner' models which converge to a high-performing 'strong learner' model when combined (Zhou 2025) by training on the errors of previous iterations. This model was selected as a candidate model due to strong performance across a wide variety of machine learning tasks; in particular, the implementation offered by the XGBoost library offers optimized training and additional regularization methods (Chen and Guestrin 2016). Similar to the Random Forest, this model treats remote sensing records as independent and does not consider temporal ordering. Therefore, it was expected to suffer similar drawbacks.

2.2.5 Feature Selection

To address collinearity among vegetation indices and evaluate the importance of both site-based and remote sensing features, three feature selection methods were applied prior to tuning. Each of the methods vary in interpretabilty and handling of collinearity, and were chosen to compensate for eachother's disadvantages in this regard.

2.2.5.1 Permutation Importance

We estimate each feature's importance by randomly shuffling its values across samples before training, then measuring the resulting change in the model's performance. This yields an interpretable, global importance metric. However, when predictors are highly correlated, it can misattribute importance—because different features may serve as proxies for one another—leading to misleading rankings (Pedregosa et al. 2011).

2.2.5.2 SHAP Values

SHAP (SHapley Additive exPlanations) return per-prediction feature contributions based on Shapley values from cooperative game theory (Lundberg and Lee 2017). This method provides both local (per-prediction) and global interpretability. However, SHAP may tacitly distribute credit among highly correlated features, depending on whether the model uses marginal or conditional expectations when computing the baseline.

2.2.5.3 Recursive Feature Elimination with Cross-Validation (RFECV)

Finally, RFECV is used to iteratively train the model and remove the least important features based on model-derived importance metrics (e.g., coefficients or feature gains). Each reduced feature subset was evaluated by its F_1 performance using cross-validation. This method directly handles correlated features by eliminating them if they do not contribute to the model performance, however it can be quite computationally exhaustive. Feature rankings based on how early features were removed are used as importance metrics.

2.3 Phase 2: Temporal Models

While previous models provide strong benchmarks for supervised learning, their assumption of independent input instances fails to capture the sequential and spatial dependencies inherent in the vegetation index data. To address this, the final phase of analysis employed Recurrent Neural Networks (RNNs), specifically Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures, which are well-suited for modeling temporal dynamics. These models, though more computationally intensive, are efficiently implemented using modern libraries such as PyTorch (Paszke et al. 2019). However, spatial correlations were not captured by this modeling paradigm; pixels within sites are still regarded as independent, which still could potentially limit model performance.

2.3.1 Processing and Sequence Generation

To prepare our data for RNN modeling, we performed a series of data preprocessing.

1. **Validation Split**

   We performed a 50:50 split on the test data to obtain a validation set for model evaluation when training the RNN models.

2. **Data Engineering**

   - `Log Transformed time_delta`: This feature records the difference between the image date and survey date. It is used to capture the irregularities in the time steps of the satellite records.

- `Negative Cosine Transformed DOY`: We perform a negative cosine transformation on `DOY` to capture the seasonality of the spectral indices.

3. **Data Normalisation**

   Since RNN models are sensitive to the scale of the data, we normalise the data to avoid vanishing or exploding gradient. To avoid data leakage, the summary statistics (mean and standard deviation) used for normalization was computed using only the training data.

4. **OneHotEncoding of `Type`**

   Since RNN models can only handle numeric data, OneHotEncoding (Pedregosa et al. 2011) was applied to the species `Type` column. `Type_Mixed` was dropped to remove linear dependencies between type columns and reduce redundancy.

5. **Sequence Generation**

   We split the site survey records and satellite records into separate data frames. For each row in the site lookup table, we searched the image table for all records with match `ID`, and `PixelID` and selected all satellite records up until the survey date. This would be the sequence data we use for training our RNN model. All survey records with no sequences available were removed from the dataset.

6. **RNN Dataset and Dataloader**

   To feed the sequence data into the RNN model, the sequence within the same batch needs to have the same sequence length. Depending on the age of the site, the sequence length for each survival record varies. To optimize memory usage while still introducing randomness to the data, we created a custom Pytorch dataset with an associated method that shuffles the dataset within their Age group to minimize the padding lengths required for each batch.

7. **Target mapping**

   We had trained regression RNN models instead of classification models, as training RNN models is time-consuming, and we want to avoid training separate RNN models for each threshold value. As such, the target values mapping to binary classes was done after model training. Further details on the RNN model design and training are presented in the next section (Section 2.3.2).

2.3.2 Modelling Pipeline

Following preprocessing, the deep learning pipeline proceeds as follows:

1. The sequence of vegetation indices and engineered features is passed through a bidirectional GRU or LSTM, producing a hidden state.

2. The static site features are concatenated onto the hidden state vector and passed to a multilayer FCNN.
3. The final layer of the FCNN output is a scalar, which is passed through a sigmoid activation and multiplied by 100 to produce an estimate of the survival rate of the site pixel.

In addition to these steps, **layer normalization** was experimented with, although no improvement to predictions was observed. **dropout** was also added within the FCNN layers, decreasing overfitting. Bidirectionality was added later in he analysis, as doing so seemed to generally increase performance by decreasing overfitting. Modeling was attemped with and without site features to assess their usage in predicting survival rate. Initally, prediction output was constrained to [0,100] using a simple 'clamp' function, but it was found that a smoother, scaled sigmoid actvation produced more consistent predictions.

Hidden state and hidden layer sizes, and the number of hidden layers are all variable hyperparameters than may effect model performance (Greff et al. 2017), however model performance seemed to 'plateau' after a certain degree of model complexity. For example, increasing the hidden state size beyond 32 did not increase prediction accuracy, nor did increasing hidden state layers beyond 3.

2.4 Evaluation Metrics

Model performance was primarily evaluated using F_1 score, **precision**, and **recall**, with classification accuracy treated as a secondary metric due to class imbalance favoring high-survival sites. To enable comparison between classical classification models and deep learning regression models, continuous survival rate predictions were thresholded to produce binary labels. In the context of these metrics, **low-survival sites are treated as the positive class**, reflecting the goal of identifying potentially failing afforestation sites for targeted intervention. To evaluate classical model performance across a range of decision thresholds, **Precision-Recall (PR) Curves** and **Receiver Operating Characteristic (ROC) Curves** were produced. As is standard for machine learning analysis, the area under each curve (**AUC, AUROC** respectively) were reported as a summary of performance across all thresholds.

3 Data Product & Results

Our project delivers a comprehensive, modular machine learning pipeline for predicting low survival rates in afforestation projects. This tool is designed for land managers, researchers, and anyone interested in using data to support better decisions in tree planting. The pipeline is also a practical example of how data science can be applied to real-world environmental challenges, and is intended to be accessible to users with a range of technical backgrounds.

3.1 What Does the Data Product Do?

Our delivered data product is a comprehensive machine learning pipeline, composed of a series of interconnected Python scripts. This pipeline automates the end-to-end process of data preparation, model development, and evaluation. Key functionalities include robust data processing steps such as cleaning, transformation (pivoting), and splitting into training and testing sets. Once the data is prepared, users can leverage dedicated scripts to train various machine learning models, perform hyperparameter tuning for optimization, and rigorously evaluate model performance. The primary objective of this data product is to provide an effective ML-driven solution for identifying and predicting instances of low survival rates.

3.2 Classical Models Evaluation

We trained several classical machine learning models, including Logistic Regression, Random Forest, and Gradient Boosting. These models were selected for their proven effectiveness in similar time series research. Logistic Regression provides interpretability, while the ensemble models capture more complex, non-linear relationships in the data. Each model was evaluated across multiple thresholds (50%, 60%, 70%, 80%) for defining low survival, providing a comprehensive view of model robustness under different definitions of the target variable.

3.2.1 Permutation Feature Importance

Permutation feature importance analysis, as illustrated in Figure 2, showed that at lower thresholds (50% and 60%), Logistic Regression placed greater importance on remote sensing vegetation indices (NDVI, EVI1, EVI2, NDWI). These indices are derived from satellite imagery and reflect the health and vigor of vegetation. As the threshold increased to 70% and 80%, tree-based models, especially Gradient Boosting, assigned higher importance to structural features like Density and Age, which are direct measures of stand structure and development. Random Forest maintained moderate importance across both types of features, indicating a balanced approach.

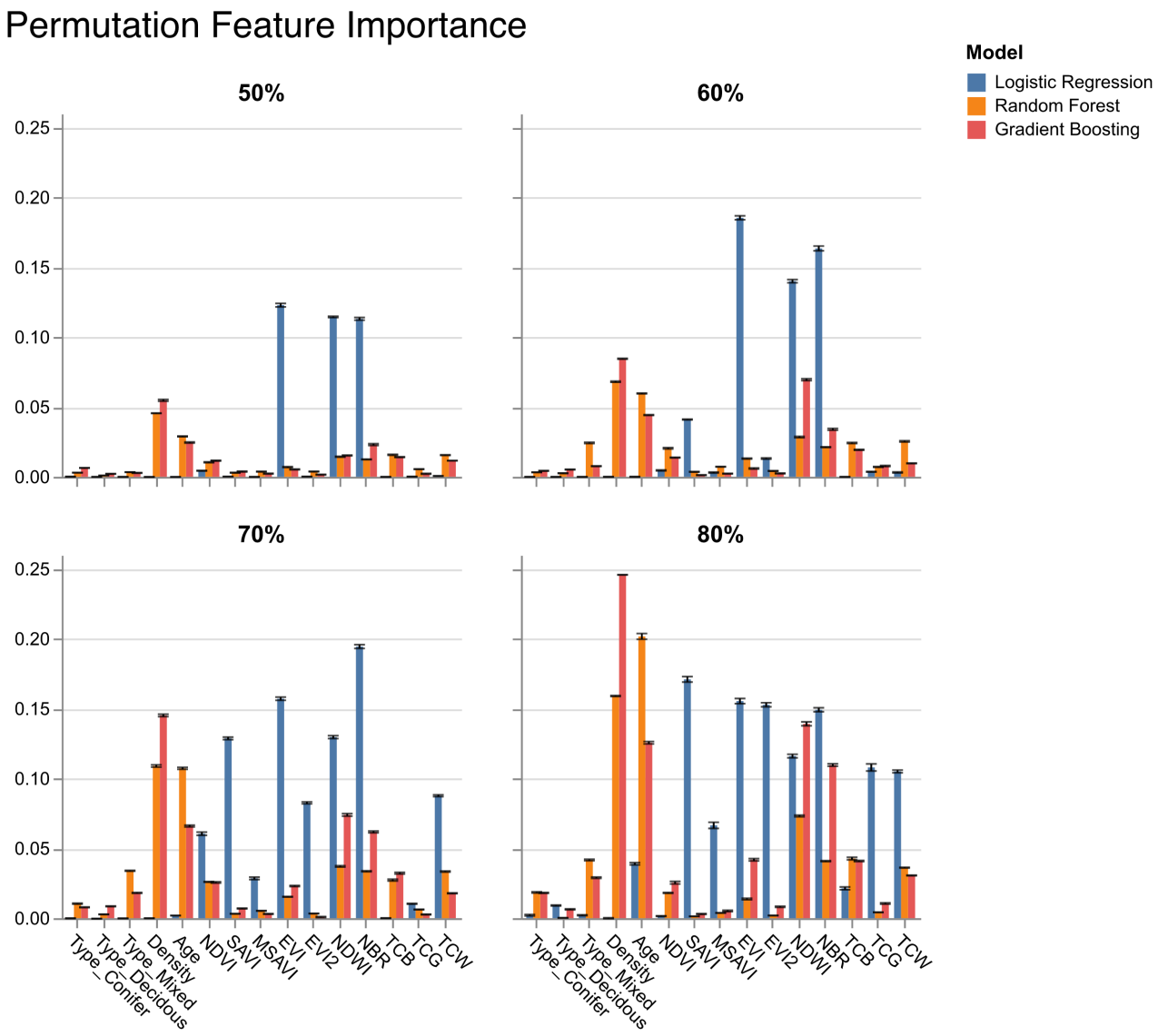

Figure 2: Permutation importance bar plot

Overall, Logistic Regression favors spectral vegetation indices, especially when the threshold for low survival is set lower. In contrast, Gradient Boosting and, to a lesser extent, Random Forest prioritize structural stand features like Density and Age more strongly as the survival threshold increases. This suggests that tree-based models may better capture non-linear relationships between structural features and survival, particularly when distinguishing very low from very high survival outcomes.

3.2.2 SHAP Feature Importance

SHAP analysis, as shown in Figure 3, provides a more nuanced view of feature contributions. Across all thresholds, Logistic Regression (blue) consistently assigns high SHAP values to vegetation indices such as NDVI, MSAVI, EVI1, EVI2, NDWI, and NBR, indicating a strong reliance on spectral information from remote sensing data for predictions. The influence of these indices becomes more pronounced as the survival rate threshold increases, with NDWI and MSAVI emerging as particularly dominant at the 70% and 80% thresholds.

In contrast, Random Forest (orange) contributes minimal feature importance across all thresholds, as indicated by its near-zero SHAP values. This suggests either weak feature attribution under SHAP for this model or that Random Forest relies more on complex interactions not easily captured by additive SHAP values.

Gradient Boosting (red) demonstrates moderate and evolving feature importance. At the 50% threshold, it shows high importance for Density and some attention to NDVI. As the threshold increases, Age becomes increasingly important for Gradient Boosting, especially at 70% and 80%, though the magnitude of SHAP values remains lower than those seen in Logistic Regression.

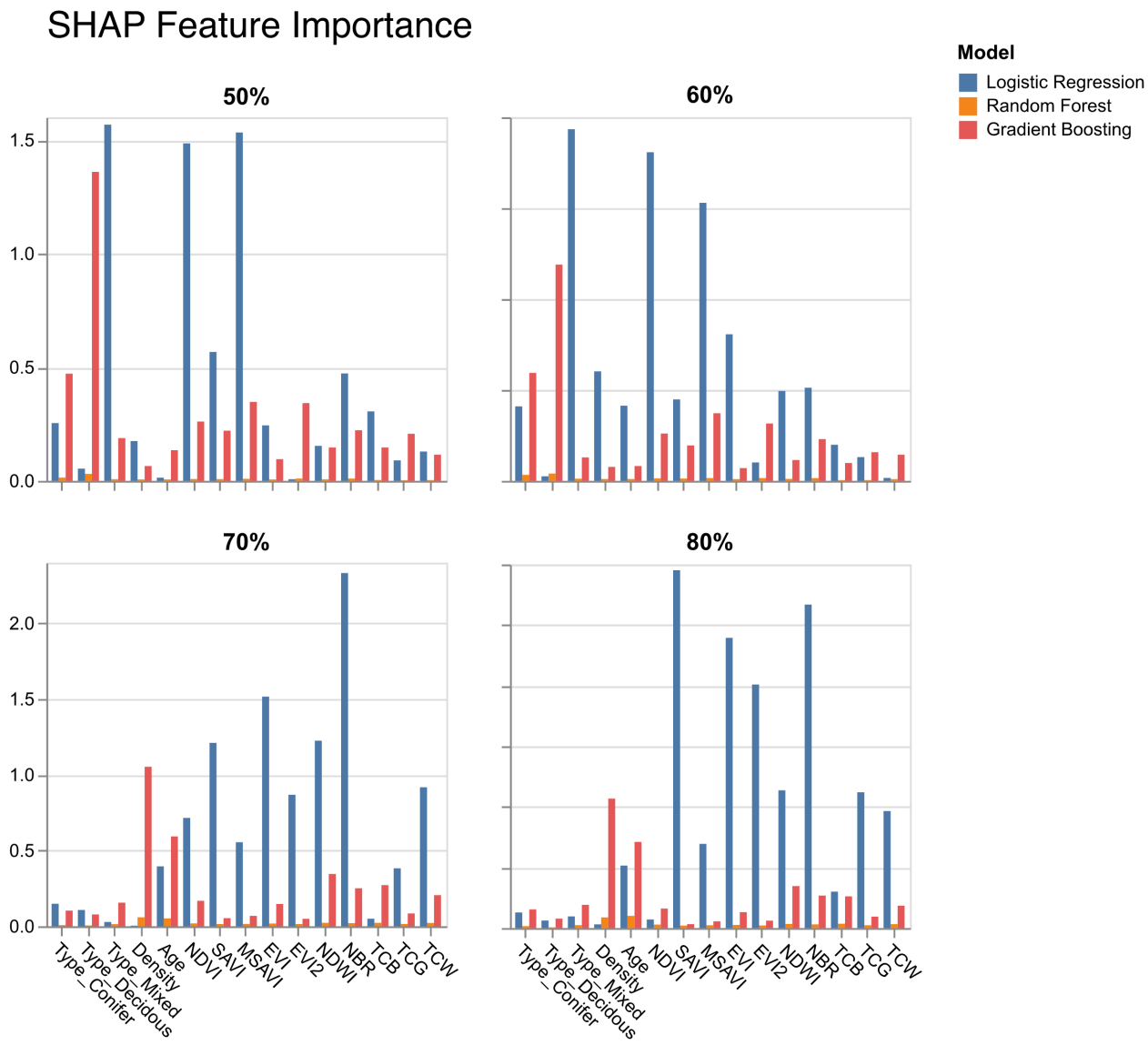

Figure 3: Shap feature importance bar plot

3.2.3 Recursive Feature Elimination (RFE)

RFE, as shown in Figure 4, demonstrated that as less informative features were removed, all models converged on a core set of impactful variables. At the highest threshold, every feature becomes highly important, highlighting the value of both site and spectral indices. This process helps streamline the model, making it more efficient and potentially more generalizable to new data. At the 50% threshold, Logistic Regression, Random Forest, and Gradient Boosting show

a varied distribution of feature importance, with some features like Age and Density appearing more significant across models. As the threshold increases to 60% and 70%, the concentration of important features becomes more pronounced, with Random Forest and Gradient Boosting consistently highlighting certain vegetation indices as highly relevant.

At the 80% threshold, the feature importance becomes more focused, with a significant portion of the heatmap dominated by darker green shades, indicating that every feature are highly relavant. The progression from 50% to 80% demonstrates how RFE expand its feature importance, potentially improving model performance by having more features.

In summary, Density seems to be the most important feature across all thresholds for every model, while the rest start to becomre more relevant as the threshold increases.

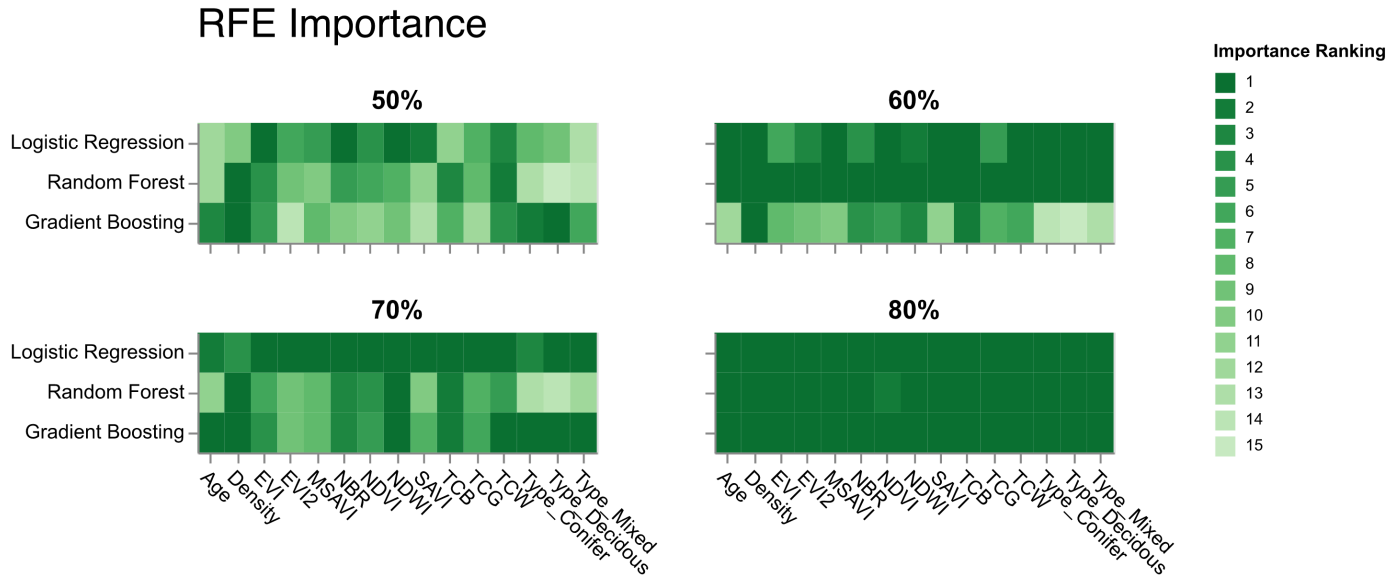

Figure 4: RFE heatmap

3.2.4 Precision-Recall Curves

The PR curves as showin in Figure 5 for Gradient Boosting, the curve starts high at low recall values but declines steadily, indicating a strong initial precision that decreases as recall increases. Logistic Regression shows a similar trend, with a sharp drop in precision after a moderate recall level, suggesting it maintains decent performance only at lower recall thresholds. In contrast, Random Forest exhibits a more stable curve, particularly at the 80% threshold, where it sustains higher precision across a broader recall range, reflecting better balance and robustness in classification performance. Overall, Random Forest appears to outperform the other models at higher thresholds, while Gradient Boosting and Logistic Regression show limitations as recall increases.

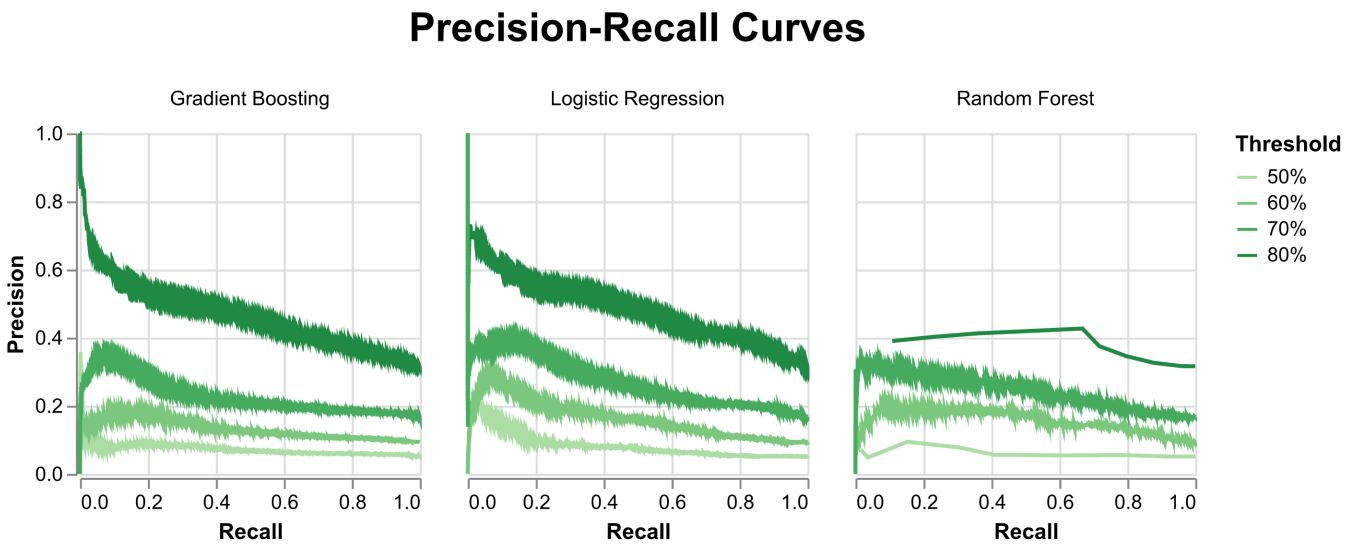

Figure 5: Precision Recall curves for each model across different thresholds

3.2.5 ROC Curves

Our ROC curves as shown in Figure 6 appear more linear rather than hugging the top-left border, suggesting that the models are not performing very well. The linear shape indicates that as we increase the number of true positives, the number of false positives also increases. In a good model, we would expect to find a point with a high true positive rate and a low false positive rate.

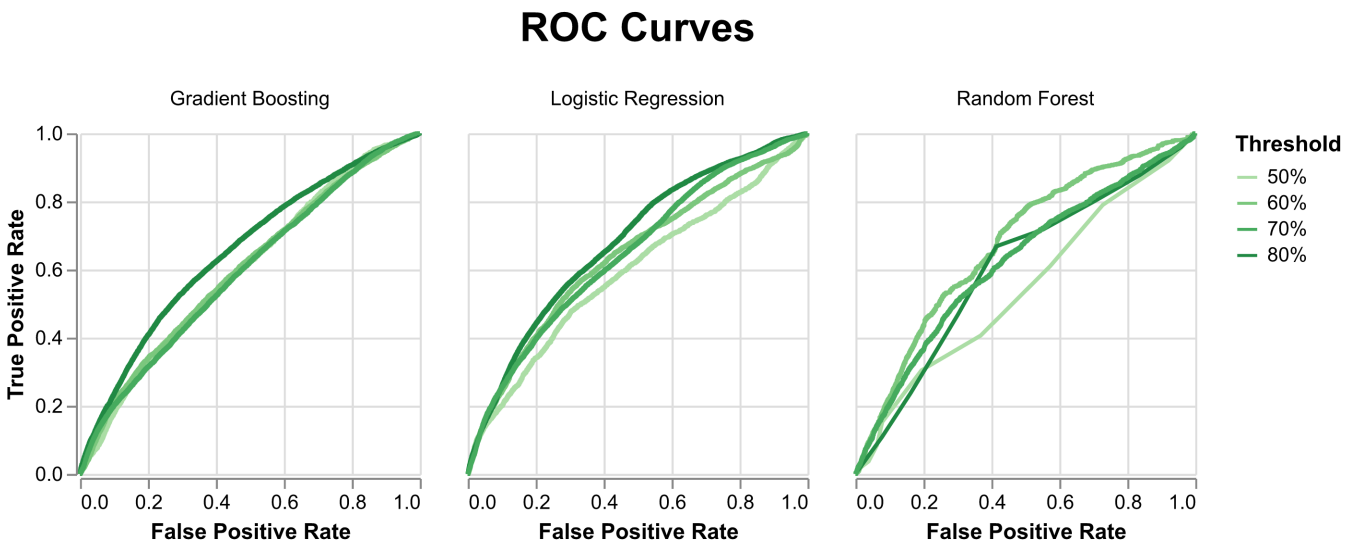

Figure 6: ROC curves for each of model across different thresholds

3.2.6 Confusion Matrices

Confusion matrices as illustrated in Figure 7 reaveals how well our classical machine learning models correctly predict true positive values which is low survival rate. Similar to what ROC curves suggest, we observe more true positives and a relatively fewer false positives at higher thresholds, but false negatives begin to increase rapidly which suggests that the issue is beyond class imbalance.

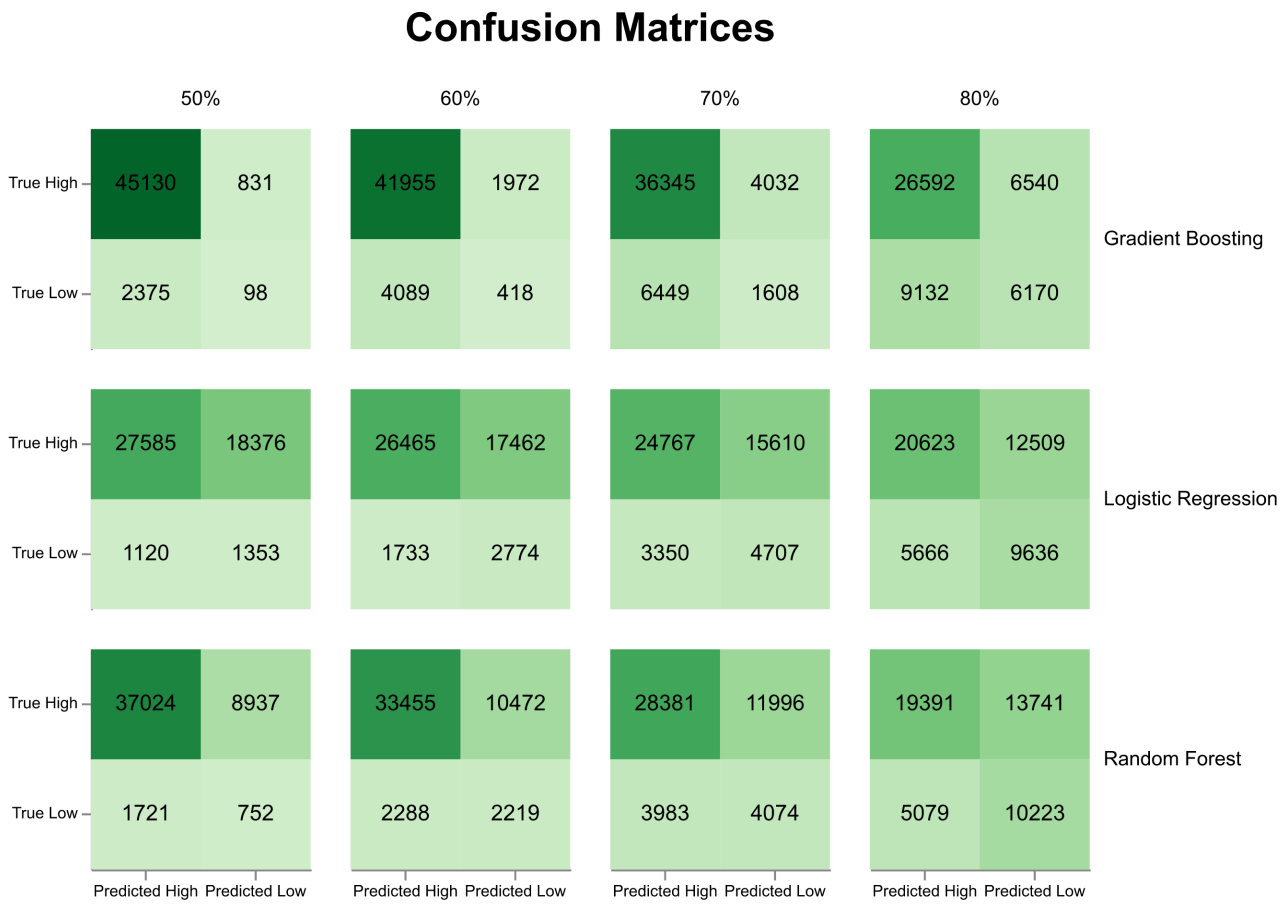

Figure 7: Confusion matrices for each model across different thresholds

3.2.7 Evaluation Metrics

Given the pronounced class imbalance in our dataset, we prioritized the F1 score as our main evaluation metric, as it balances both precision and recall. The F1 scores as shown in Table 1 across different models and thresholds reveal important trends. At the 50% threshold, all models perform poorly: Gradient Boosting achieves an F1 score of just 0.058, Logistic Regression reaches 0.122, and Random Forest attains 0.124. As the threshold increases to 80%, as shown in Table 2, performance improves markedly—Gradient Boosting rises to 0.441, Logistic Regression to 0.515, and Random Forest achieves the highest F1 score at 0.521. These results suggest that both Random Forest and Logistic Regression benefit from higher thresholds, with Random Forest consistently outperforming the others at the upper end. Gradient Boosting also improves but remains slightly behind.

Table 1: Scores at 50% threshold

	Model	Accuracy	F1 Score	F2 Score	AUC	AP
0	Gradient Boosting	0.934	0.058	0.045	0.599	0.072
8	Logistic Regression	0.597	0.122	0.228	0.590	0.083
4	Random Forest	0.780	0.124	0.192	0.545	0.061

Table 2: Scores at 80% threshold

	Model	Accuracy	F1 Score	F2 Score	AUC	AP
2	Gradient Boosting	0.676	0.441	0.417	0.653	0.465
10	Logistic Regression	0.625	0.515	0.578	0.680	0.483
6	Random Forest	0.611	0.521	0.600	0.606	0.386

3.3 Sequence Model Evaluation

We also implemented deep learning approaches, specifically Recurrent Neural Network (RNN) architectures including LSTM and GRU models, to address temporal dependencies in the data. These models are well-suited for sequential data, as they can capture patterns across time steps that classical models may overlook. The goal was to see if leveraging the time series nature of the data could improve predictive performance.

3.3.1 Residual Plots

The residual plots, as shown in Figure 8, for the Satellite and Site-Satellite datasets, featuring the GRU and LSTM models, reveal a distinctive pattern resembling a convex function, with the residuals predominantly centered around the 80% mark on the true value axis. This clustering suggests that both models tend to predict values close to 80% with high frequency across the range of true values, from approximately 30 to 100. Such a concentration indicates a potential bias in the models, where they consistently favor this particular value regardless of the actual data distribution. This behavior is highly undesirable, as it implies the models lack the flexibility to accurately capture the full spectrum of true values, rendering their predictions less useful for practical applications. The tight grouping of residuals around 80% for both GRU and LSTM, with some spread at the extremes, further highlights a limitation in their ability to adapt to diverse data points, undermining their overall predictive reliability.

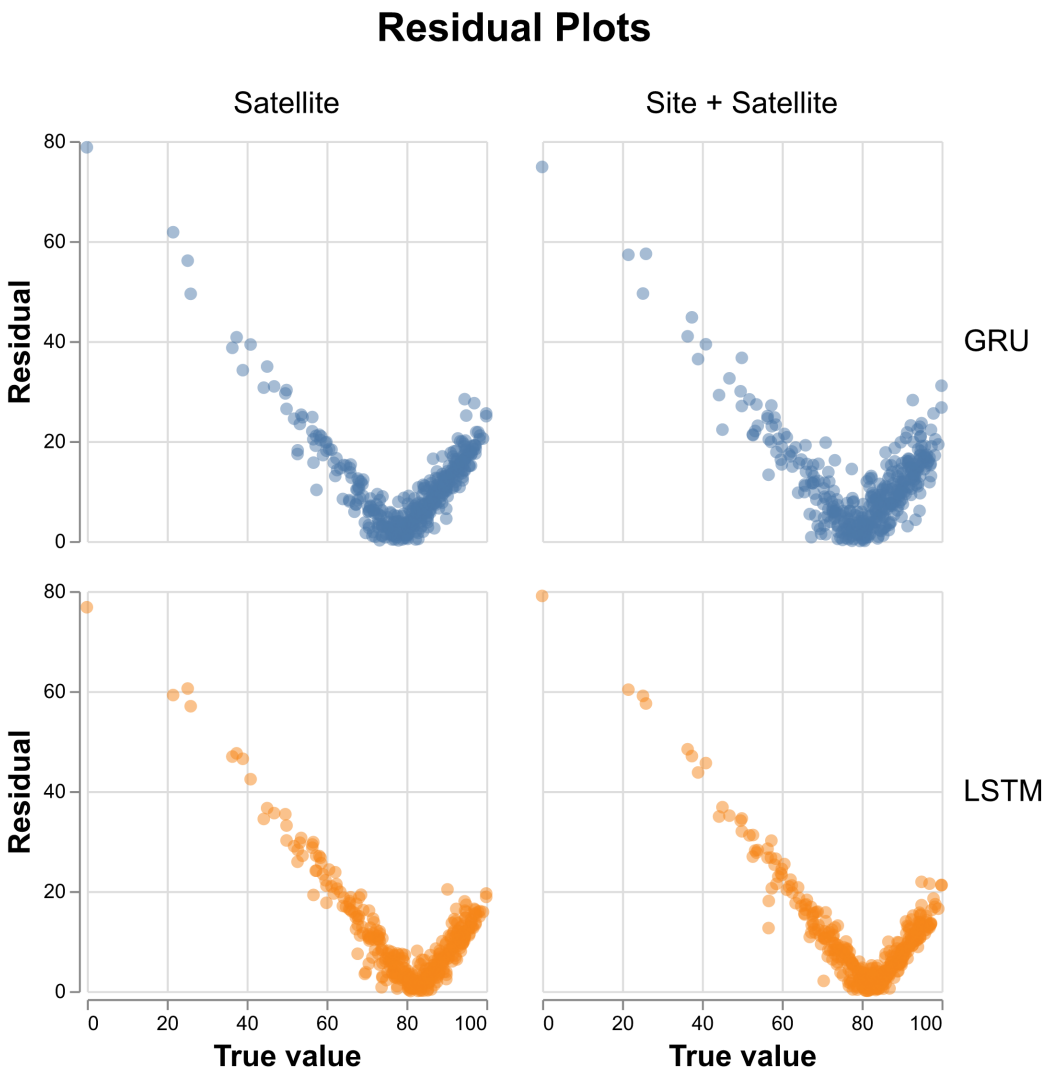

Figure 8: Residual plots

3.3.2 Confusion Matrices (RNNs)

The confusion matrices, as shown in Figure 9, at lower thresholds reveal an intriguing outcome: the model fails to make any correct predictions for lower survival rates, which is unexpected and suggests a significant limitation in its ability to identify these cases. This is likely due to the data being heavily skewed toward higher survival rates, with most data points concentrated around 100%, allowing the model to correctly predict only the high survival rates. At the higher threshold, as shown in Figure 10, of 80%, the model begins to show improvement by making some correct predictions, indicating that it is starting to learn the underlying patterns

in the data. However, the number of true positives remains lower compared to classical models, suggesting that while the model is adapting, it has not yet achieved the same level of accuracy or robustness in identifying positive cases across the full range of survival rates.

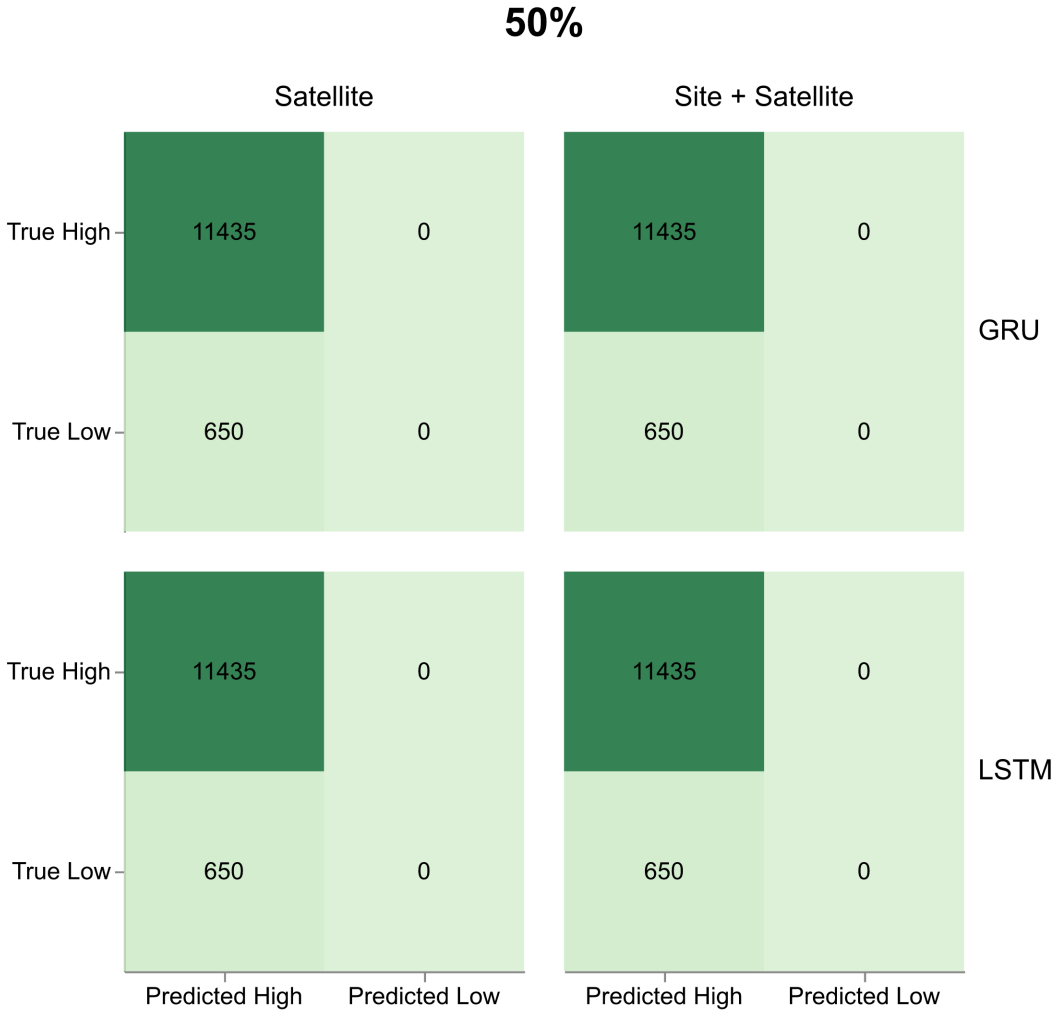

Figure 9: Confusion matrices for 50% threshold

Figure 10: Confusion matrices for 80% threshold

3.3.3 Evaluation Metrics (RNNs)

The tables highlight the F1 Scores for LSTM and GRU models using Site + Satellite and Satellite features at 50% and 80% thresholds. At the 50% threshold, all models and feature combinations show an F1 Score of 0, indicating no predictive capability. At the 80% threshold, performance improves: LSTM with Site + Satellite features achieves an F1 Score of 0.368, while LSTM with Satellite features reaches 0.393. GRU with Satellite features records a higher F1 Score of 0.434, and GRU with Site + Satellite features attains 0.44, demonstrating the best performance at this threshold. While GRU consistently outperforms LSTM, with notable improvements at the 80% threshold, it fails to outperform our classical machine learning models.

Table 3: Scores for LSTM model without site features

	F1 Score	F2 Score	Precision	Recall	Accuracy
50%	0.000	0.000	0.000	0.000	0.946
60%	0.026	0.017	0.176	0.014	0.907
70%	0.126	0.093	0.295	0.080	0.801
80%	0.391	0.392	0.390	0.392	0.635

Table 4: Scores for LSTM model with site features

	F1 Score	F2 Score	Precision	Recall	Accuracy
50%	0.000	0.000	0.000	0.000	0.946
60%	0.019	0.013	0.129	0.010	0.906
70%	0.128	0.095	0.300	0.081	0.801
80%	0.371	0.371	0.370	0.371	0.622

Table 5: Scores for GRU model without site features

	F1 Score	F2 Score	Precision	Recall	Accuracy
50%	0.000	0.000	0.000	0.000	0.946
60%	0.093	0.082	0.121	0.076	0.869
70%	0.213	0.212	0.214	0.212	0.719
80%	0.426	0.510	0.334	0.588	0.525

Table 6: Scores for GRU model with site features

	F1 Score	F2 Score	Precision	Recall	Accuracy
50%	0.000	0.000	0.000	0.000	0.946
60%	0.112	0.112	0.113	0.112	0.843
70%	0.256	0.266	0.240	0.274	0.714
80%	0.441	0.510	0.359	0.570	0.567

4 Conclusions & Recommendations

In this section, we discuss the limitations of the dataset and our modelling approaches, and provide actionable recommendations to address these issues.

4.1 Limitations

Despite exploring both classical modelling and RNN modelling approaches, all our models failed to deliver satisfactory results for predicting tree survival rates. Below, we identify the key factors limiting our model performance.

1. **Data Imbalance**

   Our data distribution was highly imbalanced, with the target values skewed heavily towards high survival rates. We believe this is the leading cause for the biased predictions across all of our models.

2. **Loss of Temporal Information in Classical Models**

   Our classical models fail to capture complex temporal structures in the satellite data. By averaging the satellite data over time, we were losing a lot of vital information, including seasonal variations and short-term vegetation responses.

3. **Lack of Spatial Information in RNN Models**

   Our RNN model lacks the ability to model spatial relationships between pixels. Each pixel is processed independently, ignoring spatial context within the same site. Neighboring pixels often share similar micro-climate and environmental conditions, without spatial data, our model may overlook key spatial dependencies that influence vegetation response.

4. **Misleading Target Labels**

   While our models were predicting at pixel level, survival records were recorded at site level and assigned uniformly to each pixel within a site. As a result, 'healthy' pixels and 'unhealthy' pixels are assigned identical targets labels, potentially misleading the model during training.

4.2 Recommendations

Here, we propose the following recommendations to mitigate current limitations and improve model robustness and predictive power in future work.

1. **Using Higher Resolution Satellite Data**

   Using higher-resolution satellite and field-survey data may help site, improving model performance.

2. **Obtaining Higher Resolution Field Survey Data** spatial resolution to improve the precision of our training targets. When combined with high resolution satellite data, this would allow models to learn localized vegetation dynamics more efficiently.

3. **Obtaining Annual Survival Records**

   Acquiring additional training data with complete annual survival rate records would substantially enhance the dataset's temporal resolution and modeling potential.

4. **Modeling at Site Level**

   Since survival records are measured at site level, we recommend that future models should aggregate satellite information across all pixels within a site, making predictions per site rather than per pixel.

5. **Incorporating Spatial Data**

   We suggest incorporating spatial data such as GPS coordinates to the current dataset, allowing the model to capture spatial correlations across sites and pixels.

6. **CNN-LSTM model**

   Alternatively, we suggest using raw satellite imagery instead of pre-extracted spectral indices. Using satellite image directly allow us to utilize convolutional architectures to learn spatial patterns, potentially improving model performance.

   We propose exploring a CNN-LSTM (Varalakshmi et al. 2024) architecture as the next step (see Figure 11). In this hybrid approach, the satellite image for each site will first passed through a CNN to extract spatial features. The CNN outputs at each time step is then fed into an LSTM or GRU to capture temporal patterns. The final hidden state can be passed through fully connected layers to predict the survival rate for the entire site. This architecture naturally accommodates both spatial and temporal dependencies, addressing key shortcomings of our current models.

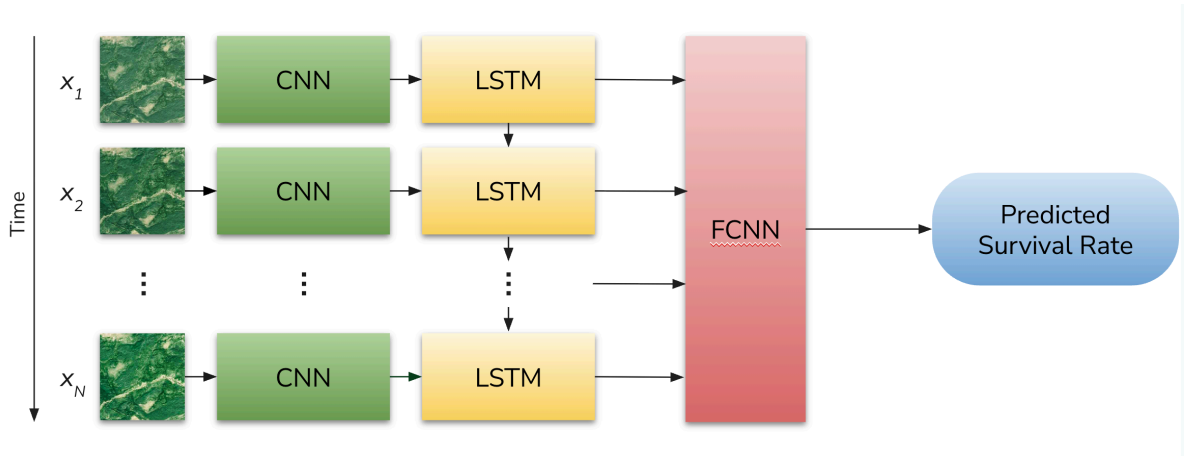

Figure 11: Basic architecture of a CNN-LSTM model, where inputs from a sequence of satellite images $(X_1, X_2, ..., X_n)$ passes through a convolution neural network (CNN) layer. The output sequence of the CNN layer is then taken one-by-one into the LSTM layer. The final hidden state of the LSTM model can then be passed through fully connected linear layers to predict the survival rate for the entire site.

References

Bergmüller, Kai O, and Mark C Vanderwel. 2022. "Predicting Tree Mortality Using Spectral Indices Derived from Multispectral UAV Imagery." *Remote Sensing* 14 (9): 2195.

Canada, Natural Resources. 2025. "Forestry Glossary | Natural Resources Canada." *Nrcan.gc.ca*. https://cfs.nrcan.gc.ca/terms/read/782.

Canada, Service. 2023. "Government of Canada." *Canada.ca*. / Gouvernement du Canada. https://www.canada.ca/en/campaign/2-billion-trees.html.

Chen, Tianqi, and Carlos Guestrin. 2016. "XGBoost: A Scalable Tree Boosting System." In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–94. KDD '16. ACM. https://doi.org/10.1145/2939672.2939785.

Filho, Mario. 2023. "How to Train a Logistic Regression Using Scikitlearn (Python)." https://forecastegy.com/posts/train-logistic-regression-scikit-learn-python/.

Greff, Klaus, Rupesh K. Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. 2017. "LSTM: A Search Space Odyssey." *IEEE Transactions on Neural Networks and Learning Systems* 28 (10): 2222–32. https://doi.org/10.1109/TNNLS.2016.2582924.

Lundberg, Scott M, and Su-In Lee. 2017. "A Unified Approach to Interpreting Model Predictions." In *Advances in Neural Information Processing Systems 30*, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, 4765–74. Curran Associates, Inc. http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

Natural Resources Canada. 2021. "2 Billion Trees Program." https://www.canada.ca/en/campaign/2-billion-trees/2-billion-trees-program.html.

Paszke, Adam, Sam Gross, Francesco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, et al. 2019. "PyTorch: An Imperative Style, High-Performance Deep Learning Library." In *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. 2011. "Scikit-Learn: Machine Learning in Python." *Journal of Machine Learning Research* 12: 2825–30.

Sruthi. 2025. "Random Forest Algorithm in Machine Learning." *Analytics Vidhya*, May. https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/.

University of British Columbia Master of Data Science Program. 2025. "Remote Sensing for Forest Recovery." https://pages.github.ubc.ca/mds-2024-25/DSCI_591_capstone-proj_students/proposals/Remote_Sensing_for_Forest_Recovery.html.

Varalakshmi, P et al. 2024. "Agroforestry Mapping Using Multi Temporal Hybrid CNN+ LSTM Framework with Landsat 8 Satellite Imagery and Google Earth Engine." *Environmental Research Communications* 6 (6): 065009.

Zhou, Zhi-Hua. 2025. *Ensemble Methods: Foundations and Algorithms*. CRC press.