

Article

Supervised Machine Learning Approach for Detecting Missing Clamps in Rail Fastening System from Differential Eddy Current Measurements

Praneeth Chandran ^{1,*} , Florian Thierry ¹, Johan Odelius ¹ , Stephen M. Famurewa ¹, Håkan Lind ² and Matti Rantatalo ¹

¹ Division of Operation and Maintenance Engineering, Lulea University of Technology, 97187 Lulea, Sweden; florian.thierry@ltu.se (F.T.); johan.odelius@ltu.se (J.O.); stephen.famurewa@ltu.se (S.M.F.); Matti.Rantatalo@ltu.se (M.R.)

² Bombardier Transport, 11743 Stockholm, Sweden; hakan.lind@rail.bombardier.com

* Correspondence: praneeth.chandran@ltu.se

Abstract: The rail fastening system forms an integral part of rail tracks, as it maintains the rail in a fixed position, upholding the track stability and track gauge. Hence, it becomes necessary to monitor their conditions periodically to ensure safe and reliable operation of the railway. Inspection is normally carried out manually by trained operators or by employing 2-D visual inspection methods. However, these methods have drawbacks when visibility is minimal and are found to be expensive and time consuming. In the previous study, the authors proposed a train-based differential eddy current sensor system that uses the principle of electromagnetic induction for inspecting the railway fastening system that can overcome the above-mentioned challenges. The sensor system includes two individual differential eddy current sensors with a driving field frequency of 18 kHz and 27 kHz respectively. This study analyses the performance of a machine learning algorithm for detecting and analysing missing clamps within the fastening system, measured using a train-based differential eddy current sensor. The data required for the study was collected from field measurements carried out along a heavy haul railway line in the north of Sweden, using the train-based differential eddy current sensor system. Six classification algorithms are tested in this study and the best performing model achieved a precision and recall of 96.64% and 95.52% respectively. The results from the study shows that the performance of the machine learning algorithms improved when features from both the driving channels were used simultaneously to represent the fasteners. The best performing algorithm also maintained a good balance between the precision and recall scores during the test stage.



Citation: Chandran, P.; Thierry, F.; Odelius, J.; Famurewa, S.M.; Lind, H.; Rantatalo, M. Supervised Machine Learning Approach for Detecting Missing Clamps in Rail Fastening System from Differential Eddy Current Measurements. *Appl. Sci.* **2021**, *11*, 4018. <https://doi.org/10.3390/app11094018>

Academic Editor: Cecilia Surace

Received: 18 March 2021

Accepted: 24 April 2021

Published: 28 April 2021

Keywords: clamps; rail fastening system; differential eddy current sensor; machine learning; classification

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Rail transport has emerged as a significant mode of transportation to overcome heavy congestions of road and sky, increasing energy costs and carbon emissions. It is an effective mode of transportation that supports the economic and industrial expansion of a nation, through mobilization and transportation of people and commodity [1]. Rail freight transport and passenger traffic has increased rapidly in Europe between 1990 and 2007. There has been a 15% increase in rail freight ton-kilometres and a 28% increase in passenger-kilometres between 1990 and 2007 in EU15 countries [2]. In Sweden, there has been an average annual growth of 1.1% traffic on the railway network between 1960 and 2010 and a minimum annual increase of 1% in traffic tonnage is anticipated up to 2050 [3]. The need to shift huge volumes of passenger and freight traffic to railways and the current state of the existing railway infrastructure are issues that require significant attention in the field of transportation [4]. One possible solution to meet the growing demand and

improve the rail performance could be capital expansion of the infrastructure, but this is a time consuming and cost-intensive approach. Hence, in order to improve the capacity, availability and the service quality of the existing infrastructure, an ideal solution would be to improve the maintenance and renewal (M&R) process.

The quality of the infrastructure and the utilization methods play a crucial role in determining the operational capacity of a given railway infrastructure [5]. The dependence between operational capacity and the condition of the infrastructure is a crucial aspect in railway infrastructure maintenance. When the railway infrastructure is in a good state with high quality, a higher operational capacity with higher quality of service is achieved. As the operational capacity increases, the infrastructure is subjected to more traffic and load, which leads to deterioration of the infrastructure and deformation of its components. Therefore, higher-frequency maintenance and renewal are required. These demand track possession, which in turn reduces the operational capacity. The down time arising from maintenance and renewal of networks is responsible for nearly half of all the delays to passengers. To investigate the actual delay within Sweden, the records of the infrastructure manager was analysed. Tables 1 and 2 show the delay in hours for 10 years, due to failure of components in track and switches and crossings (S&C) respectively. The delay time used for this study only includes the downtime of service arising due to the corrective maintenance employed to fix the fault or failure. The downtime arising due to corrective maintenance is unplanned and can occur at any given point of time, having consequential effect on traffic. On the other hand preventive maintenance actions are planned, i.e., the track occupation is guaranteed without traffic interference. However, in a few instances, preventive maintenance can occupy the track longer than the allocated time frame, thereby leading to traffic interference or delay [6], which is not included in the present study. The components considered for this study include only those that exhibit magnetic properties. Averages of 572.7 h and 670.3 h of delays are incurred in Sweden yearly due to failure of components in track and S&C respectively. To avoid such delays and to ensure safe and reliable operation, tracks and components need to be inspected periodically.

Track inspection is a crucial task that has to be carried out periodically to prevent catastrophic accidents and control the condition of the railway infrastructure [7]. Traditionally, rail inspections are carried out by trained inspectors who walk along the track length to search for visible defects and technical deviations. Such manual inspections are labour-intensive, slow and may include human errors, especially in tough winter condition. This method is time-consuming and expensive for railroad companies, especially for long-term and large-scale development projects. Recent technological developments have seen automated inspections systems based on machine vision being utilized for track inspections. Automated rail inspection systems are composed of various functions including, gauge measurement, rail-surface defect detection, rail profile measurement and fastener defect detection [8]. Rail fastening systems are a crucial component in the rail infrastructure as they clamp the rail to the sleepers, maintaining the gauge, preventing the longitudinal and transverse deviation of rails from the sleepers, and preserving the designed geometry of the track. Failures of fasteners reduce the safety of train operations, increase wheel flange wear and may lead to catastrophic accidents [9]. In the past two decades, the application of automated machine vision for fastener inspection has grown significantly; however, the detection method from these rail images have varied over time.

Table 1. Delay in hours due to failure of track components.

Year	Fasteners	Rail Corrosion	Rail Defects	Rail Wear	Insulated Joints	Track Foreign Object	Total
2010	18	6	792	1	10	11	838
2011	70	2	313	1	6	96	488
2012	17	0	397	6	22	10	452
2013	10	0	372	10	12	6	410
2014	13	1	479	37	21	18	569
2015	13	14	476	4	9	30	546
2016	50	7	686	1	17	18	779
2017	1	0	540	1	118	13	673
2018	7	0	393	0	49	6	455
2019	4	0	456	30	4	2	496

Table 2. Delay in hours due to failure of components in switches and crossings.

Year	Fastener	Crossings	Middle Part	Check Rail	Rail Corrosion	Rail Defect	Rail Wear	Insulated Joint	Turnout Foreign Object	Total
2010	8	93	1	1	6	260	3	0	22	509
2011	13	21	0	0	9	200	5	0	33	344
2012	1	38	0	1	6	282	105	0	51	592
2013	2	83	0	0	10	499	14	0	55	760
2014	29	37	86	0	31	274	36	0	40	643
2015	8	72	0	14	10	521	19	0	50	811
2016	18	94	3	1	12	602	33	2	39	948
2017	8	40	3	4	13	456	72	0	33	747
2018	9	43	0	1	18	491	20	1	49	780
2019	12	84	0	3	4	332	19	1	50	568

In 2007, Marino et al. [10] detected missing hexagonal-headed bolts from rail images using a multilayer perceptron neuron classifier. For hook-shaped fasteners Stella et al. [11] used a neural classifier to detect missing fasteners, employing wavelet transform and principal component analysis to preprocess the railway images. Yang et al. [12] used direction field as a template of the fastener in the rail images and used linear discriminant analysis (LDA) for matching, to obtain the weight coefficient matrix. To model two types of fasteners, Ruvo et al. [13] used error back propagation algorithm on the rail images and implemented the same on graphical processing unit to achieve real time performance. Ruvo et al. [14] also introduced a FPGA-based architecture employing the same algorithm on rail images. Xia et al. [15] and Rubinsztein [16] used AdaBoost algorithm for detecting fasteners from rail images. Li et al. [17] adopted image-processing techniques to detect fasteners and its components from the images obtained from visual inspections. H.Feng et al. [8] adopted structure topic model (STM) to model fasteners and learn from the probabilistic representation of different components within rail images. H.Fan et al. [18] used line local binary pattern (LLBP) on the rail images to distinguish between normal fasteners and failed fasteners. Support vector machines (SVM) [19], Gabor filters [20] and edge detection [21] methods are other commonly employed techniques to detect fasteners from rail images. Recent advancements in image processing have enabled the use of deep learning to [22,23] R-CNN [24] detect fasteners from rail images collected during automated visual inspection.

The present mode of fastener detection is carried out from images acquired during automated visual inspection of track and its components. They become a complicated task when the rail and its components are concealed due to the presence of rust and dust. Presence of snow, stones and other debris or heavy rain can also cause hindrance in visual inspection and minimise the efficiency in detecting the rail and its components. A reliable and high-quality automated visual inspection is a relatively expensive technique to carry out and are difficult to mount and maintain on an in-service train as they are integrated in the operation and are subjected to brightness fluctuations and motion blurring during high-speed travel and can reduce the accuracy of detection. In Sweden around 298,080 euros

alone was spent in 2014 to inspect two lines with a total track length of ca. 300 km, of which more than 75% was utilised to inspect track components that exhibit magnetic characteristics (rail fastening, weld joints, rail surface etc.) [25]. With the increasing demand for safety and cost-effectiveness, maintenance managers are striving to cut these operations and maintenance cost through effective condition-based maintenance while assuring the quality and capacity of the rail services.

In earlier research, the authors proposed a train-based differential eddy current sensor [25] for fastener inspection that can overcome the major challenges mentioned above. Eddy current sensors are not affected by the presence of nonconductive materials in the sensor-to-target gap and can be used in dirty environments, water, oil etc., where other inspection systems fail. The proposed inspection technique using a differential eddy current sensor was able to detect fastener signature from a distance of 65 mm above the railhead. The individual fastener signatures were easily distinguishable from the 1-D time signal plots. The sensor uses two driving fields with frequencies 18 kHz and 27 kHz respectively and the fastener signatures obtained from both the channels exhibited very good correlation. The sensor concept and working principle of the sensor is explained in Section 2 of the previous publication [25]. The results presented in the previous literature was based on study carried out for a short track section. However, when considering other track sections with an increased likelihood of disturbances, it was found that relying on one feature could generate a decrease in accuracy for detection. Hence, the purpose of this study was to develop a measurement system using the previously presented hardware solution in combination with a machine learning algorithm for detecting missing clamps within a rail fastening system. This study also compares different algorithms and evaluates their performance. The remainder of the paper is structured as follows. Section 2 elaborates the research methodology used for the study. The results and analysis are explained in Section 3 and the conclusions are discussed in Section 4.

2. Research Methodology

One of the most commonly observed faults in a rail fastening system is missing clamps. A missing clamp reduces the clamping force holding the rail on the sleeper. The track integrity is called in to question as soon as clamps are missing from the fastening system in consecutive sleepers as it may lead to slipping, excessive gage widening and low lateral resistance, which can further lead to risk of derailment. As stated in Section 1, the goal of this study is to develop an automated system for detecting missing clamps within a rail fastening system, using signals generated and measured by a differential eddy current sensor. An outline of the methodology used to develop an automated system is depicted in Figure 1. The data required for this study was collected using the differential eddy current sensor system. Features from the signal recorded using the two channels were extracted with the aid of signal processing techniques. The feature sets from individual channels and combined features from both channel were used as inputs for the classification algorithms. The idea of making use of three sets of input was to compare whether a single channel or both channels when combined performs better in missing clamp detection. The data were further processed before feeding as an input to the six classification algorithms used in this study. The algorithms were optimised using cross validation technique by splitting the input data in to training, validation and test set. The goal was to identify which algorithm performed the best for the missing clamp detection purpose. The steps involved in this study are elaborated further below. The system will make use of features extracted from the differential eddy current signals as an input for multiclass classification algorithm to differentiate intact clamps and one or two missing clamps from a fastening system. Only a missing clamp is considered as a failure in the fastening system as this is the most common failure mode of the fastener system found along the section of track examined for this study. This study makes use of standard laptop (Dell Ultrabook) with python 3.6 (with necessary packages such as Numpy [26], Pandas [27], and scikit-learn [28]) and Spyder.

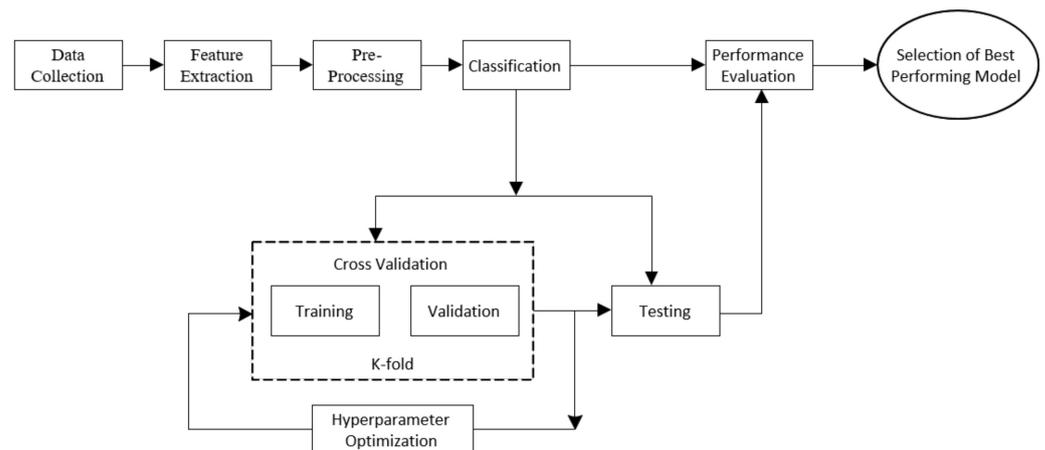


Figure 1. Generic fastener detection framework using machine learning algorithm.

2.1. Data Collection

The data for this study were collected along the northern loop of the heavy haul line at Katterjåkk and Stordalen, close to the Sweden–Norway border. The differential eddy current sensor was mounted 65 mm above the railhead on a trolley system and was made to run along the track, using a motor (refer Figure 2a). The speed of the trolley system varied between 1.3 m/s to 2 m/s. The sensor system, which is used to measure one side of the track, consists of two differential eddy current sensors with a driving field of 18 kHz and 27 kHz respectively placed at a distance of 20 cm apart in the travel direction. The above-mentioned carrier frequencies are selected as they fall under the rail norms. Each sensor consists of two differentially coupled pickup coils (P1 and P2) that are enclosed by the driving coil. The direct cross talk between the driver and the pickup coils within an individual sensor is cancelled, though not completely, by the differentially coupled pick-up coil. The inbuilt cross-talk cancellation (CTC) function within the sensor system cancels out the small cross talk between the two-carrier frequencies as they have a common factor of 9 KHz. The entire unit is vacuum potted with epoxy resin to stabilize the sensor, both against vibrations and to reduce temperature drift. The output voltage from an individual sensor is the result of the cross-talk residue between the driving coil and the pickup coils, and the induction of eddy currents along the rail, which are linearly superimposed. The differentially coupled pickup coils (P1–P2) are sensitive only to the changes in the eddy current in the rail and its vicinity. The resulting voltage will be zero if there is no change in either conductivity (σ), magnetic permeability (μ) or geometric form of the measured material, e.g., an ideal rail with no clamps or any surface defects.

The sensor was powered using a 12V62AH battery and the measured raw data were recorded using a laptop. The considered track sections consisted of concrete sleepers with E-clip fasteners. Tracks which were relatively healthy and track sections with damages were all considered for this study. Some of the visible damages on the rail head included squats, rail corrugation, crack and head checks. Figure 2b depicts a fastening system with intact clamps, a fastening system with one missing clamp and a fastening system with two missing clamps. A controlled measurement sequence was carried out along the track section to obtain the dataset for the e-clip fastening system. A pattern of missing clamps was created, along a measurement sequence, where clamps were removed from the outer and the inner side as well as from both sides at the 20th, 25th and 30th sleeper respectively, from the starting position of the measurement (refer Figure 3). This measurement sequence was carried out along various sections of the track. A fastening system with intact clamps was considered as a healthy system and fastening systems with one clamp missing (stage 1 fault) and two clamps missing (stage 2 fault) were considered as faulty system for this study.

A total of 2967 fastener signatures were used for this study, of which 2700 samples (91%) of the instances correspond to a healthy state (stage 0) with intact clamps, 168 samples

(5.67%) correspond to faulty fasteners with one clamp missing (stage 1) and the remaining 99 samples (3.33%) represent a faulty state with both clamps missing (stage 2).

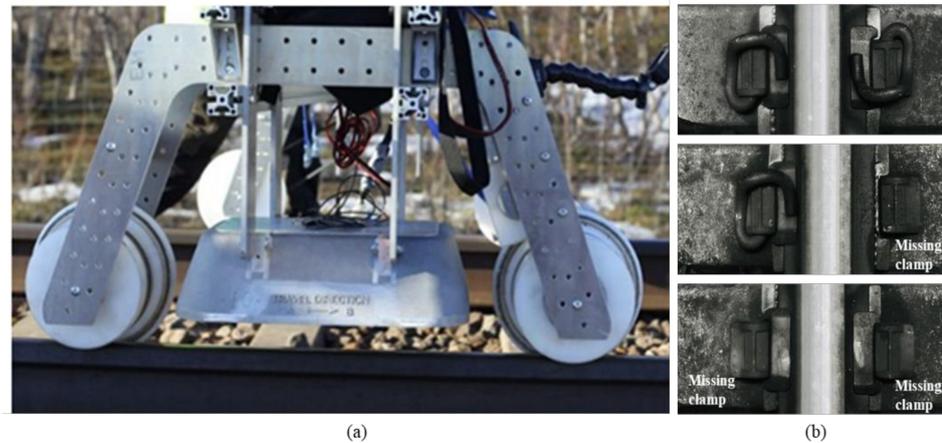


Figure 2. (a) Lindometer measurement system setup, (b) e-clip fastening system: intact, one clamp missing and both clamp missing.

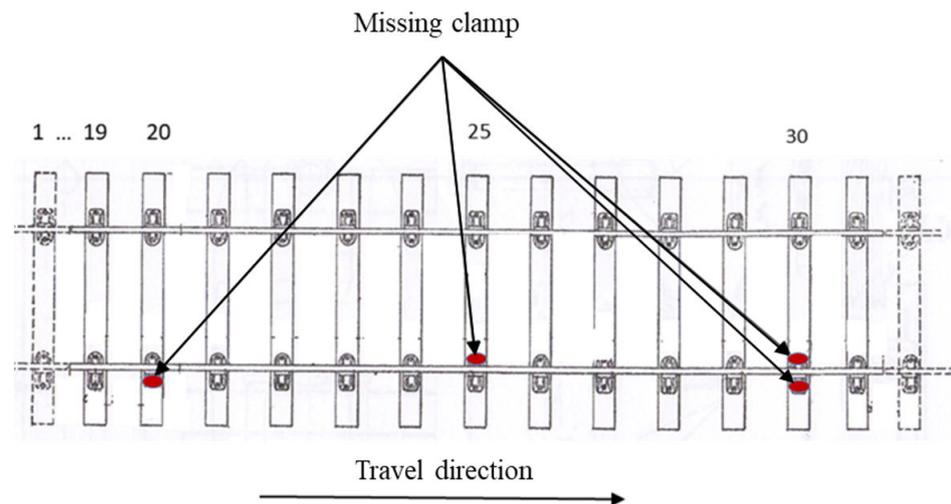


Figure 3. Pattern of missing clamps. Red ellipse marks the position of the missing clamp.

2.2. Feature Extraction

A number of signal processing methods were implemented before features were extracted from the raw signal pertaining to the individual fastener signatures. The eddy current (EC) signal had to be demodulated, filtered and rotated in order to extract information corresponding to the fastening system [25].

2.2.1. Demodulation and Resampling

The sensor signal was multiplied by its carrier frequency (for both channels respectively) and low-pass filtered (2 kHz) to demodulate the signal and extract the base band. The signal was further resampled from 215.52 kHz to 35.92 kHz.

2.2.2. Filtering

The EC signal was filtered further with a low pass filter of 3 Hz as the periodicity of the fasteners in the signal was found to be lower than 3 Hz, for both the channels. This was carried out to retrieve maximum information pertaining to the fastener system and attenuate other frequency components corresponding to noise and other ferromagnetic components.

2.2.3. Rotation of EC Signal

The demodulated and filtered fastener signatures were found to be shifted from the in-phase direction (real part). In order to retrieve maximum information and have a better visualisation, the complex EC signal was rotated such that the fastener signatures were projected along the in-phase direction. This to an extent aids in suppressing other responses in other demodulation angles not prevailing to the fastening system. The EC signal was rotated by degree θ or Φ radian, such that the peak amplitude of the fastener signatures were maximised. The signals were rotated by an optimal angle (found from the previous study [25]) of 83° and 222° , respectively for the two carrier frequencies.

The EC signal does not get affected due to the presence of nonconductive or nonmagnetic materials in the sensor-to-target gap. The disturbances arising due to the presence of conductive and magnetic material in the sensor-to-target gap can be suppressed to an extent by the above-mentioned low pass filtering and rotation of EC signal techniques. The low pass filter was set to extract the fastener signatures and to remove other high frequency components which could add to the energy content of the fastener signatures. The cut off frequency of the filter is dependent on the speed of the sensor and must be adjusted accordingly. Different components will have different geometric shapes, different values of magnetic permeability and electrical conductivity. Hence they will occur at different angles from the in-phase direction compared to the fasteners. Rotation of the EC signal based on fastener signature will thus to a major extent suppress information pertaining to other disturbances.

Four features for both channels are extracted for individual fasteners namely peak-to-peak, RMS, and magnitude of the fastener signature at clamp frequency and the arc length of the complex signal. Three separate feature matrices will be used as an input for the classification purpose in this study. The first feature matrices will use the four features obtained from fastener signatures acquired from the 18 kHz channel. The four features obtained from the 27 kHz channel representing the fastener signature will be used for the second feature matrices. The first and second feature matrices will have a dimension of 2967×4 (2967 samples and four features). The third matrices contain the combination of 18 kHz and 27 kHz features, eight features in total representing a fastener signature. Hence the third feature matrices will have a dimension of 2967×8 (2967 samples and eight features respectively). The three feature matrices are tested to identify whether a single channel or both channels when combined perform better in missing clamp detection.

2.3. Preprocessing

From the feature extraction section, a feature matrix is obtained containing the various features obtained from the signal processing for each observation. To improve the classification algorithms, the feature matrix can be rescaled using either normalization or standardization, when the features have different scales. Normalization rescales the data by squeezing the values into the range $[0, 1]$. This technique might be useful in cases where all parameters need to have the same positive scale, but normalizing the data can be sensitive to outliers. Standardization (or z-score normalization) rescales the data to have a mean value (μ) of zero and a standard deviation (σ) of one. In this study, the choice of standardization is adopted since the data has a Gaussian distribution and there is no need to obtain a positive range for the feature matrix.

2.4. ML Algorithm

The fastener detection in this study is a multiclass classification problem with the objective of classifying fastener signatures from differential EC sensor into healthy fasteners with no clamps missing, fasteners with one and both clamps missing. There are a wide range of machine learning algorithms that can solve multiclass classification problems. The classification algorithms are usually evaluated based on the classification accuracy [29]. However, for practical implementations, other key factors, such as running time efficiency for training, ease of implementation, parameter tuning time, prediction time and the ease of

continuous updating the algorithm online, need to be considered. A set of widely popular algorithms was selected for this study and each of them were compared to determine the best-suited algorithm for this application. Six established classifiers were selected: Gaussian naive Bayes (GNB), support vector machines (SVM), k-nearest neighbours (k-NN), gradient boosting decision trees (GBDT), random forest (RF) and AdaBoost (AB). The naive Bayes classifier was used to understand the lower bound classification performance on the fastener dataset and as a means to compare the other algorithms. The choice of the remaining selected algorithms was based on the performance, ease of implementation and the speed exhibited by the classifiers during training, parameter tuning and predictions observed in previous literatures [29–32]. Brief explanations for each algorithm are presented in the following subsection. Features extracted from both channels were tested using these algorithms, both individually and combined.

2.4.1. Gaussian Naive Bayes (GNB)

Naive Bayes classifier is a statistical classification technique based on Bayes theorem. Each and every feature variable are considered as an independent variable in naive Bayes. This probabilistic-based classifier can be trained very efficiently for supervised classification purpose and can be used in complex real world situations. GNB is a type of naive Bayes method that assumes a Gaussian or normal distribution on the values of the given class. The probability density of an observation x_i given a class c_k is computed as follows:

$$P(x_i|c_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(x_i - \mu_k)^2}{2\sigma_k^2}} \quad (1)$$

where μ_k the mean of the values in x is associated with class c_k and σ_k^2 is Bessel corrected variance of the values in x associated with class c_k .

2.4.2. Support Vector Machines (SVM)

SVM was initially developed for binary classification problems and, due to various complexities, its extension to multiclass problems is not straightforward. The SVM algorithm creates a line or hyperplane that separates the data points to different classes. This line or hyperplane is called the decision boundary and the elements of the input data that defines the boundary are called support vectors. For SVM, the best hyperplane is the one that maximises the margins from both the classes. In general, the larger these margins are, the lower the generalisation error of the classifier. SVM algorithm uses a set of mathematical functions called kernels to take the data as input and transform it into the required form. The kernel functions return the inner product between two points in a suitable feature dimension. In general, it is used to transform a nonlinear decision surface to a linear equation, in a higher number of dimensions.

To solve a multiclass classification problem, several binary SVM classifiers are combined. The most common approaches for combining binary classifiers are one vs. one (OvO), one vs. rest (OvA), directed acyclic graph (DAG) and error-corrected output coding [32]. For this study the one vs. one method was selected as this method requires lesser training data vectors for each classifier and the memory required to create the kernel matrix is much smaller. For ' M ' classes, the OvO algorithm constructs one binary classifier for every pair of distinct classes and a total of $M \times (M - 1)/2$ binary classifiers are constructed. The training samples are made to be inputs to each classifier and the output from each classifier is in the form of a class label. The max wins algorithm is used to combine these classifiers and the class label that occurs the most is assigned to that point in the data vector. A comprehensive explanation of multiclass classification using SVM is provided in [33]

2.4.3. k-Nearest Neighbour (k-NN)

k-NN is a nonparametric method where the training data is stored to be compared with unclassified data points rather than constructing a generic function for classification.

For this reason, k-NN algorithms are often called instance-based learning algorithms or lazy-learning algorithms. k-NN predicts the class of a test data by the majority rule, i.e., prediction with majority class of its k most similar training data points in the feature space. Due to its simplicity, easy implementation and ability to handle complex problems, k-NN algorithms are widely used for various applications [34]. A comprehensive description of k-NN is given by Cover and Hart [35]

2.4.4. Random Forest (RF)

RF is a type of ensemble classifier that works, based on the philosophy that a multitude of classifiers perform better than a single classifier. Each classifier is generated using random vectors sampled independently from the input vector, and each classifier contributes with a single vote for assigning the frequent class to the input vector [36]. The class is assigned based on the majority vote received. The output from every classifiers are averaged to improve the predictive accuracy and reduces the risk of overfitting to the training dataset. Breiman, 2001 [36] provides a comprehensive explanation of RF.

2.4.5. AdaBoost (AB)

AdaBoost, like RF, is an ensemble classifier that combines weak classifiers to form a strong classifier. AB starts with finding a weak classifier and subsequently fits it to a subset of training data to generate a new classifier [37]. The AB algorithm retrains iteratively by choosing the training subset based on the performance of the previous training. AB assigns a training weight after each training by a classifier and a misclassified item is assigned higher weight so that it appears with a higher probability in the training subset of the next classifier. A weight is also assigned to the classifier. A better performing classifier is assigned a higher weight so that it will generate more impact in the final output. For an input vector x of n features and $h_t(x)$ is the output of the t^{th} weak classifier, the combined classifier is expressed as

$$H(x) = \text{sign} \left[\sum_{t=1}^T \alpha_t h_t(x) \right] \quad (2)$$

where

$$\alpha_t = \frac{1}{2} \ln \left[\frac{(1 - e_t)}{e_t} \right] \quad (3)$$

e_t is the error of the weak classifier. The importance of the weak classifier becomes greater as the error becomes smaller. A comprehensive description regarding AB can be found in [38].

2.4.6. Gradient Boosting Decision Tree (GBDT)

GBDT makes a prediction by training ensemble of weak decision trees in a gradual, additive and sequential manner [37]. GBDT identifies the shortcomings of the weak classifier by using gradients in the loss function unlike AB, which uses high weight data points. The loss function indicates how good the model's coefficient is in fitting the underlying data points. GBDT improves the prediction of the ensemble via incremental minimisation of errors in successive iterations of new decision tree construction [39]. Friedman [40] provides a generic form of GBDT as follows:

$$F(x) = \sum_{j=1}^M \beta_j h_j(x) \quad (4)$$

where β_j is the coefficient calculated by the gradient boosting algorithm and $h_j(x)$ is the individual decision tree generated in each sequence. Friedman [39] and Schapire [37] gives a comprehensive description of the GBDT algorithm.

2.5. Cross Validation

To investigate the problem of underfitting or overfitting resulting from a simple holdout validation, the dataset should be split into a training set, a validation set, and a testing set to judge how the algorithm will perform with new external data. In this case, the training set is used to train the model, the evaluation is performed on the validation set to minimize bias and variance and a final evaluation is performed on the test set to evaluate if the algorithm is either underfitting or overfitting. However, partitioning the available data into three distinct sets will reduce drastically the number of samples used for training the model. As a result, cross validation (CV) was used to evaluate our machine learning models with limited data samples. The test set had to be held out as before, but the validation test was not needed since the cross validation (CV) handled validation by itself within the training set. The main principle behind CV such as k -fold is to split the training set into k smaller sets and to train the model on $k - 1$ sets. Validation of the model is performed on the remaining part of the data set to compute the performance measure. This process is repeated k times so that each k -fold serves as a validating group once, while the performance measure will be an average of the measure for each k -fold. In our multi-class application, the f1-macro was adopted as the performance measure. It is defined as the average of the harmonic mean of precision and recall for all classes. The f1-macro was calculated for each k -fold and averaged to obtain the final performance measure for a fixed set before parameter tuning. The data set was split in to training set (70%) and test set (30%) and the cross validation was carried out within the training set.

The process of cross-validation can be looped over to optimize the parameters of each classification method. This strategy to optimize the parameters when training the model is called nested cross validation. In our case, the optimization was performed exhaustively by generating candidates from a grid of parameters relevant to each classifier. Hence, the goal of the grid search was to evaluate all the possible combinations of the parameters within suitable ranges to hold the best combination with regard to the scoring function.

SVM, k-NN, RF, AB and GBDT need parameter tuning to achieve their best performances. A limited number of values for each parameter was selected for optimising, as it was unfeasible to try the entire range of values possible. The parameters selected for optimisation were based on literature studies [29,40,41]. The parameter value selected for optimisation and the value range are described below:

- SVM. Three main parameters are selected for optimising the SVM classifier: (i) kernel, which maps the observation into a feature space; (ii) C , which is the regularization parameter that aids in controlling the punishment given to the model for each misclassified point for a given curve; (iii) gamma, which defines how far the influence of a single training example reaches. Three-kernel types were tested: sigmoid, polynomial and radial basis function (RBF). The range for C and gamma were set from 10^{-12} to 10^{12} using a logarithmic scale.
- k-NN. Two main parameters were optimised for the k-NN algorithm: (i) the number of neighbours k with a value range of 1 to 38 in a linear scale and (ii) weight function, namely 'uniform' and 'distance'
- RF. Three parameters were examined for random forest: (i) the number of trees in the forest with a value range 2^n [where $n = 1$ to 9]; (ii) the maximum depth of the tree with a value range of 1–32; and (iii) the minimum number of samples required to split an internal node with a value range of [2 32]. All the vector values were defined using a linear scale.
- AB. Two parameters were optimized for AB algorithm: (i) the number of classifiers, which limits the construction of classifiers during boosting and the range of linear values selected is [10, 50, 100, 150, 200, 250, 500, 1000] and (ii) the learning rate, which shrinks the correction contribution of each classifier. Linear value range of [0.001, 0.005, 0.01, 0.05, 0.1, 0.15, 0.25, 0.5, 0.75, 1] was used for learning rate and the values selected were based on prior knowledge and previous literature.

- GBDT. Three parameters were examined for the GBDT algorithm: (i) the number of boosting stages to be performed, with a linear value range 2^n [where $n = 1$ to 9]; (ii) learning rate, which shrinks the contribution of each tree. A linear value range of $[0.001, 0.005, 0.01, 0.05, 0.1, 0.15, 0.25, 0.5, 0.75, 1]$ was used for examining the learning rate. (iii) Maximum depth of individual regression estimation, which limits the number of nodes in the tree. A linear value range of $[2$ to $32]$ was used for examining maximum depth.

Since a grid search algorithm was used for hyperparameter optimization, the search will be performed for all points predefined in the grid.

2.6. Model Performance Evaluation

Evaluation metrics are used to understand and evaluate how effective the model is based on some score, performed on the test dataset. Different evaluation metrics emphasise different aspects of performance of the classifier algorithm. It is essential to define the objectives of evaluation in order to enable suitable selection of metrics. The classification approach for this study is a multiclass classification model and the evaluation of these models can be understood in the context of a binary classification model.

The major evaluation metrics for binary classification model are derived from four categories.

- True positives (TP) where the actual label is positive and correctly predicted class is positive;
- False positives (FP) where the actual label is negative but incorrectly predicted as positive;
- True negatives (TN) where the actual label is negative and correctly predicted as negative;
- False negatives (FN) where the actual label is positive but incorrectly predicted as negative.

A multiclass classification problem can be viewed as a set of many binary classification problem. The commonly used evaluation metrics for multiclass classification based on these elements are described in Table A1 (in Appendix A). Standard metrics such as average accuracy and error rate can provide significant insight regarding the overall performance of the model. However, the data set used for this study consists of instances with unbalanced dataset, with the major class being the healthy system. Accuracy and error rate can become unreliable measures of model performance when the skew in class distribution is severe. Accuracy and error rate also have limitations with respect to different misclassification costs [42]. For railway application, the cost of misclassifying faulty systems as healthy systems is higher than misclassifying healthy systems as faulty systems. The former instance may lead to less intervention that can lead to hazardous deterioration of track system integrity and increase the risk of derailment. The later scenario of misclassifying healthy systems as faulty systems will not pose a threat to the safety and technical integrity of the system. However, the number of site visits might increase due to false alarms leading to increase in cost of maintenance. For the above-mentioned reasons accuracy and error were not considered for evaluating the models in this study.

The limitations of accuracy and error can be tackled using precision and recall or a combination of both. Precision quantifies the number of true positives of all actual positive predictions, however precision alone cannot be used to evaluate the performance of the model as it does not provide insight on the missed positive predictions. Recall is used to complement precision as it quantifies the number of correct positive predictions made from all possible positive predictions that could have been made. Precision addresses the question “of all the positive predictions, what is the probability of them being correct?” and recall addresses the question “among all the items that belong to positive class, how many does the classifier actually detect as positive?” Precision and recall aim at minimising false positives and false negatives respectively. Often increase in precision leads to the

reduction of recall. F-score or F1-score, which is the harmonic mean of the fractions of precision and recall, can be used to combine the properties of both precision and recall in to one single metric. F1-score can be used as a strong metric for model evaluation when both precision and recall are important.

For multiclass classification precision, recall and F1-score are represented by microaveraging or macroaveraging. A microaveraging method aggregates the contribution of all classes to calculate the average performance, whereas macroaveraging will calculate the performance individually for each class and then performs an average on it. Macroaverage considers each class equally. Since the study involves using an imbalanced data set, macroaveraging was utilised as it is insensitive to the class imbalance and treats them equal. For this study, the models were evaluated based on macroprecision, macrorecall and F1-score macro during the testing phase. F1-score macro was used as a scoring metric during the training and validation phase for parameter tuning.

3. Results and Analysis

Figure 4a,b shows the time signal plot after demodulating, resampling, filtering and rotating the raw signal, for both the driving field for a single measurement sequence. The measurement was carried along a short section of the track having 47 sleepers. This section of the track was relatively healthy without any damages to the railhead. The zero crossing in the signal from the positive to the negative induction represents the centre positioning of the fastening system. The zero crossing was used as a way to segregate individual fastening systems. Individual fastening systems are easily distinguishable from the time signal plot for both driving channels. A drop in amplitude of the fastener signature is visible at those positions where the clamps were missing from the fasteners. Missing clamps cause reduction in the metallic material and changes the geometry of the fastening system, thus reducing the amplitude of the return field. The healthy fastening systems with intact clamps are depicted using black markers and fastening systems with one clamp missing and fasteners with both clamps missing are represented using blue and red markers respectively.

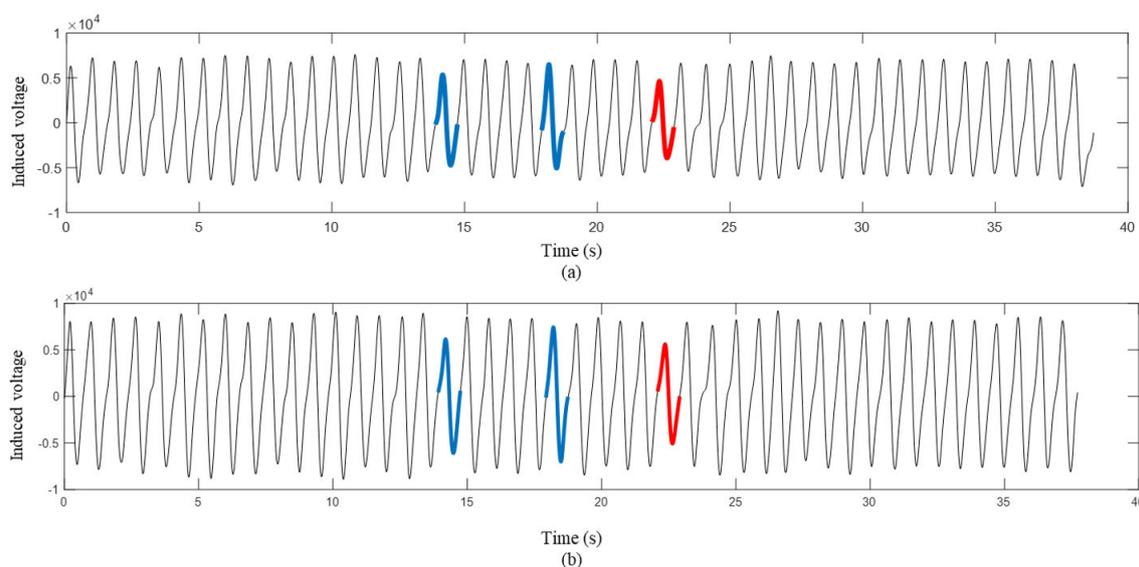


Figure 4. Time signal after demodulating, filtering and rotating the raw signal: (a) 18 kHz, (b) 27 kHz. Fastening systems with one clamp missing are depicted with blue markers and fasteners with both clamps missing are depicted using red markers. Healthy clamps are indicated with black markers.

Figure 5 depicts features pertaining to individual fasteners for both channels, obtained from the above time signal (refer Figure 4). All the four features, for both channels showed a drop in their value, when the clamps were missing from the fastening system. The

above plots are obtained for a short track section where the difference between feature levels are easily distinguishable. However, this clear difference between the three classes is not as evident for all track sections where additional disturbances can be present as observed in Figure 6. Figure 6 shows the histogram plot (in probability density function) for all the eight features for the entire data set of fasteners (from both 18 kHz and 27 kHz channel). The healthy fasteners are indicated within the black markers and fasteners with one or both clamps missing are marked in blue and red respectively. It is evident from the plot that there are significant overlaps between the three areas for all the features. The disturbances in track can affect the induced voltage in the eddy current sensor, thus causing fluctuations in the features. The overlap is seen in features extracted from both channels. It becomes a difficult task to mark a boundary or threshold to differentiate the state of fasteners and use individual features for classification purposes. This calls for the use of machine learning algorithms that can combine multiple features to create a boundary or threshold for classification purpose. Figure 7 depicts the 3-D representation of the overall data set used for classification using three random features for the 18 kHz channel.

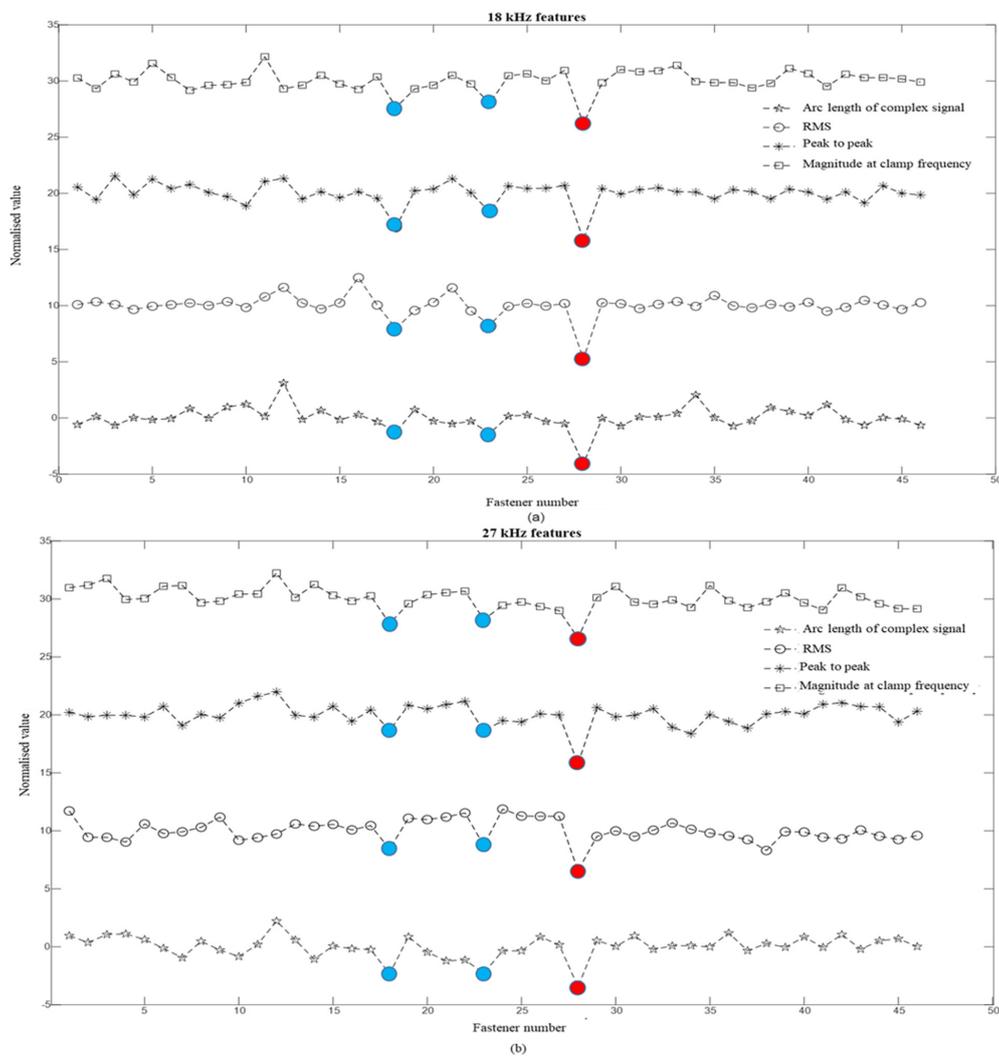


Figure 5. Standardised features for the measurement sequence (Figure 4) extracted from: (a) 18 kHz channel, (b) 27 kHz channel. Fastening systems with one clamp missing are depicted with blue markers and fasteners with both clamp missing are depicted using red markers. Healthy clamps are indicated with black markers.

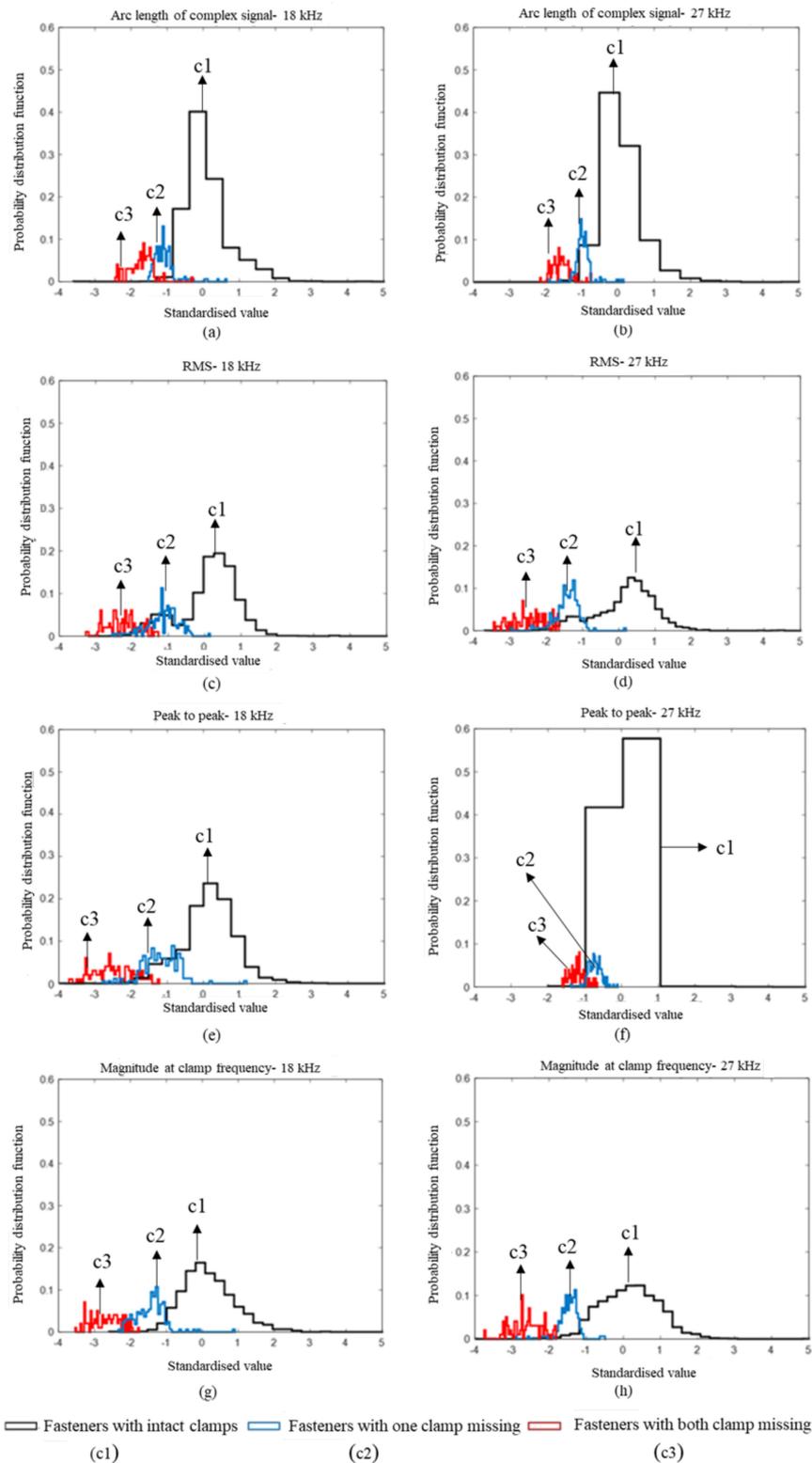


Figure 6. Histogram plots in probability distribution function with respect to standardized feature values for all features: (a) Arc length of complex signal-18 kHz, (b) Arc length of complex signal-27 kHz, (c) RMS-18 kHz, (d) RMS-27 kHz, (e) Peak to peak-18 kHz, (f) Peak to peak-27 kHz, (g) Magnitude at clamp frequency-18 kHz, (h) Magnitude at clamp frequency-27 kHz. Fastening systems with one clamp missing are depicted with blue markers and fasteners with both clamp missing are depicted using red markers. Healthy clamps are indicated with black markers.

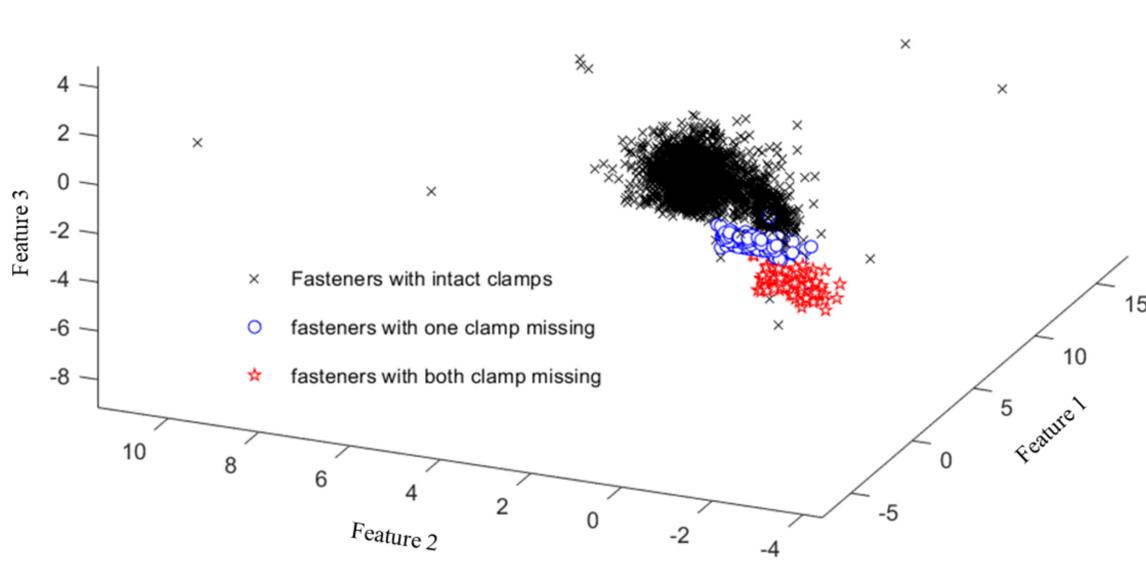


Figure 7. Overall data set used for classification represented in 3-D window (three random features selected for representation after standardization).

The training, validation and testing performance for all the six algorithms for 18 kHz channel, 27 kHz channel and both channels combined are presented in Table 3. The scores presented in the above table are obtained after cross validation where the best hyperparameter are extracted. The best hyperparameter combination for all algorithms are depicted in Figure 8. All the algorithms except Gaussian naïve Bayes exhibited an accuracy of above 97% for all scenarios, during training, validation and testing. Since accuracy is not considered as a strong metric for evaluating performance on imbalanced data, the model comparisons were carried out using F1-score macro. There were no huge variations in the score during training, validation and testing for all the cases, indicating that the algorithms did not overfit or underfit the data. The scores during training validation and testing did not vary significantly, indicating that the models did not have high bias and high variance. Figure 8 represents the learning curve for all the algorithms with their respective optimised hyperparameters. From the learning curve plot, it is evident that, for all the algorithms, the training samples were sufficient as the training score converged to a value. All the validation scores also tended to converge to a value close to the training score. Increasing the training sample may slightly increase the validation score. Since the training score was high (low error), the training data was well fitted by all the estimated models, indicating a low bias nature. The gap between the training and validation curve for all the algorithms was minimal, indicating low variance.

Table 3. Performance of the algorithms during training, validation and testing phase (all scores are in percentile).

Algorithm	Training Score (F1-Macro)			Validation Score (F1-Macro)			Testing Score (F1-Macro)		
	18 kHz	27 kHz	18 + 27 kHz	18 kHz	27 kHz	18 + 27 kHz	18 kHz	27 kHz	18 + 27 kHz
GNB	87.49	88.53	89.96	87.46	87.98	89.46	86.06	86.24	89.02
SVM	94.11	95.04	95.26	93.26	94.12	94.24	91.19	92.92	93.15
K-NN	94.62	95.53	96.14	93.86	94.76	95.36	92.29	93.97	94.56
RF	93.09	94.79	96.29	92.06	93.68	95.12	91.42	92.26	94.62
AB	93.17	96.01	96.50	92.44	95.63	96.28	89.33	95.15	96.02
GBDT	92.98	94.87	96.19	91.84	94.03	95.38	89.10	92.42	94.58

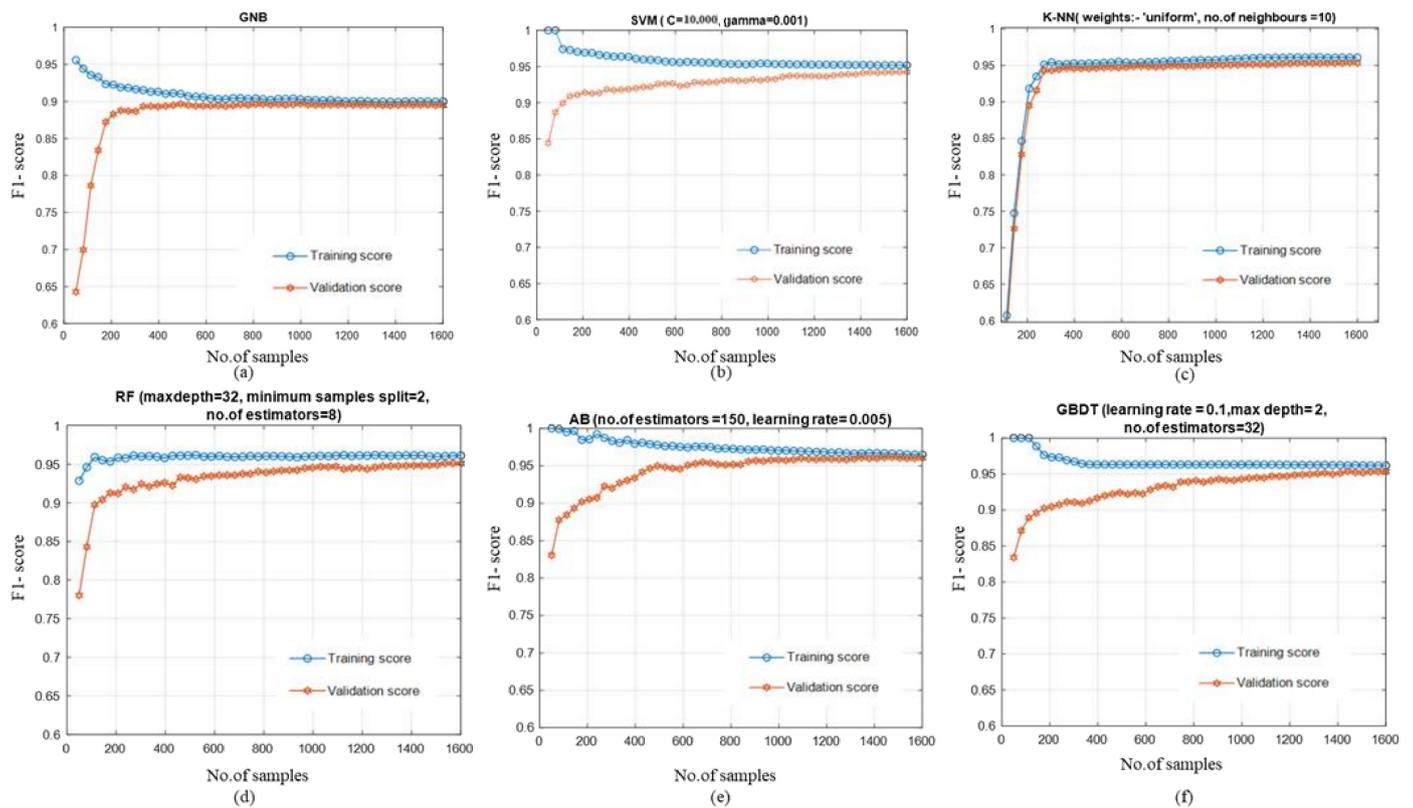


Figure 8. Learning curve with respect to number of training samples and F1-score for all algorithms: (a) Gaussian naive Bayes, (b) SVM, (c) K-NN, (d) RF, (e) AB, (f) GBDT. The best hyperparameter combinations/values after cross validation are marked above for each algorithm.

Gaussian naive Bayes exhibited the lowest score during training, validation and testing for all three feature sets with an F1-score below 90%. All the other five algorithms exhibited a F1-score macro better than the baseline Gaussian naive Bayes classifier during training, validation and testing, for the respective data set. Thus it was easier to rule out GNB classifier when considering the best performing algorithm for fastener detection. The performances of the remaining five algorithms were evaluated closely in order to select the best performing model. The F1-score was above 89% during training, validation and testing stage, for the features extracted from 18 kHz channel. SVM, K-NN and random forest (RF) had an F1-score above 90% for training, validation and testing, where K-NN registered the highest score. K-NN had a score (F1) of 93.86% during validation phase and 92.29% while testing. Ada-Boost and GBDT had a test score of less than 90% during testing phase.

The F1-score for all the algorithms during testing and training for the data set of 27 kHz features was better compared to the 18 kHz dataset. The score was above 92% for the entire algorithm tested on the 27 kHz data set during every stage. AdaBoost algorithm exhibited the best performance for this data set with a score of 96.01% during training, 95.63% during validation and a score of 95.15% during testing. When the data set comprised the features of both 18 kHz and 27 kHz representing the fastener, the F1-score for all the algorithms improved further during training, validation and testing, compared to when the data set contained features from individual channel. AdaBoost was the best performing algorithm for the combined data set, with an F1 score above 96% during all three stages. An F1-score of above 94% was achieved for k-NN, RF and GBDT during testing phase on the combined data set. It is evident from Table 3 that the detection of the clamp was higher when features from both channels were used simultaneously to represent a single clamp.

In railway applications, it is essential to have both high precision and recall. A higher precision of the algorithm will minimise the false positive rates. A higher precision will

thus contribute to better detection of the fastener state, thus ensuring safe and reliable operation of the railway. On the other hand, a higher recall of the algorithm will ensure the false negative rates are minimised. A higher recall will thus minimise cost incurred due to unwanted inspection. It is necessary to have a balance between reducing cost of inspection as well upholding the safety and reliability of the railway asset. Hence, the best algorithm was not selected based on the F1-score alone, but also on precision and recall score and the balance between the two, during the test stage.

Table 4 represents the results obtained during the testing phase, for all the six algorithms. For the data set obtained from the 18 kHz channel, RF registered the highest precision (93.43%); however, the recall for the same was below 90%. Recall was seen to be highest for the k-NN algorithm with 94.29%, while the precision for the same was around 90%. No algorithm exhibited a good balance between the precision and recall during the testing phase for the 18 kHz data set. For the data from 27 kHz channel, the highest precision (94.16%) was achieved using random forest, with a recall of 90.46%. The highest recall (96.45%) for the 27 kHz data set was achieved with AB, with a precision of 93.92%. The algorithms with higher performance on this data set, however, did not exhibit a good balance between precision and recall. Precision and recall value were well balanced for all the algorithms when the data set included features, from both channels were used simultaneously to represent the clamp. Highest precision and recall were achieved by the AdaBoost algorithm, at 96.64% and 95.52% respectively, with the highest F1-score of 96.02 during testing.

Table 4. Performance evaluation of the testing stage (all scores are in percentile).

Algorithm	F1-Macro			Macro Precision			Macro Recall		
	18 kHz	27 kHz	18 + 27 kHz	18 kHz	27 kHz	18 + 27 kHz	18 kHz	27 kHz	18 + 27 kHz
GNB	86.06	86.24	89.02	83.09	82.6	84.13	89.24	90.21	94.69
SVM	91.19	92.92	93.15	90.64	91.72	92.16	92.66	94.15	94.29
K-NN	92.29	93.97	94.56	90.47	91.89	94.72	94.29	96.28	94.41
RF	91.42	92.26	94.62	93.43	94.16	94.72	89.54	90.46	94.57
AB	89.33	95.15	96.02	90.07	93.92	96.64	88.79	96.45	95.52
GBDT	89.10	92.42	94.58	87.28	93.05	94.22	91.63	91.68	95.34

4. Conclusions and Future Work

At present, rails and fasteners are still inspected with the aid of automated visual inspection and manual inspection, despite the fact these methods require huge investment in both capital and time. Moreover, automated visual inspection becomes a challenge when the rail and its component are obscured due to adverse environmental conditions, such as snow, debris etc. The authors proposed a train-based differential eddy current sensor that can overcome such challenges in the previous study [24]. This paper attempts to develop an intelligent system with the aid of machine learning approach to facilitate, effective and reliable health monitoring of track system by reducing human biases and error for detection of the state of railway fastening system. The data set used for this study was obtained from the measurements carried out along a heavy haul line in the north of Sweden using the differential eddy current measurement system. Three sets of data were used for analysis in this study, two of which were obtained from the individual channel of the sensor system (18 kHz and 27 kHz), four features representing each. The third data set comprised of the combined features from both channels (eight features). Six machine-learning algorithms were selected and compared among themselves for all three scenarios of data, to determine the best-performing model.

The result of the study shows that all the algorithms perform better, when the data set includes features from both channels rather than when they are considered individually. Among the three data sources, all the algorithms performed weakly on the 18 kHz data set comparatively. The performance of the algorithms based on F1-score was higher for the 27 kHz data set and for data set with combined features. In railway industry, it is essential to balance the risk of failure and the cost of inspection. Therefore, it is necessary to have a good balance between the precision and recall of the detection algorithm. Both channels are preferred for detection of railway fastening systems as the algorithms performed better and exhibited a good balance between precision and recall, when the features of both channels were used for representing an individual clamp. Among the six algorithms tested, AdaBoost, which is a type of ensemble algorithm, slightly outperformed the other four algorithms in all evaluation matrices.

Further research will be carried out to incorporate other classification techniques (e.g., artificial neural network, XGBoost, etc.) that can have the capability of improving the predictive strength for fastener detection. The possibilities of unsupervised clustering for anomaly detection will also be investigated in the further studies. The current study incorporates only one type of fastener, namely Pandrol e-clip. The future of this study will also focus on different types of fasteners that can be distinguished by the rotation angle, as different fastening systems have different geometrical shapes. The future scope in this study also involves high-speed measurements, detection of other magnetic track components, quantification of rail defects and developing efficient condition monitoring techniques with the aid of artificial intelligence to detect and predict faults from big data. The work in this study was based on using features obtained from the measurement signals as an input to machine learning algorithms. These features are subject to change when the distance between the sensor and the object varies (i.e., lift-off effect). In this application, lift off can occur due to wheel wear. However, this is a slow-occurring process which can be handled by continuous automatic calibration of the system where healthy signatures are used as a reference. This study will be carried out in future work.

Author Contributions: Conceptualization, P.C., F.T., J.O., H.L. and M.R.; methodology, P.C., F.T. and M.R.; software, P.C., F.T. and J.O.; validation, P.C. and F.T.; formal analysis, P.C., F.T., J.O., S.M.F. and M.R.; investigation, P.C., F.T., resources, P.C., H.L., J.O., S.M.F. and M.R.; data curation, P.C., F.T., J.O. and S.M.F.; writing—original draft preparation, P.C.; writing—review and editing, P.C., F.T., J.O., S.M.F. and M.R.; supervision, M.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: This research was supported and funded by the Luleå Railway Research Centre (JVTC), Vinnova INFRASWEDEN 2030 and Trafikverket (Swedish Transport Administration), through the European Shift2Rail project IN2SMART. The authors would also like to thank Ulf Ranggard of EIOptic i Norden AB, Håkan Lind, Anders Thornemo and David Lindow of Bombardier Transportation Sweden and Olavi Kumpulainen of Consistent AB, for their guidance in this study.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Evaluation metrics for multiclass classification [40].

Evaluation Metric	Mathematical Formula	Description
Accuracy	$\sum_{i=1}^k \left(\frac{t_{pi} + t_{ni}}{t_{pi} + t_{ni} + f_{pi} + f_{ni}} \right)$	The overall accuracy/effectiveness of a classifier
Error rate	$\frac{\sum_{i=1}^k (f_{pi} + f_{ni})}{k}$	The average per classification error
Precision _μ	$\frac{\sum_{i=1}^k t_{pi}}{\sum_{i=1}^k (t_{pi} + f_{pi})}$	Agreement of data class labels with the ones from classifier, calculated from sums of individual decisions
Recall _μ	$\frac{\sum_{i=1}^k t_{pi}}{\sum_{i=1}^k (t_{pi} + f_{ni})}$	Effectiveness to identify class labels by the classifier, calculated from sums of individual decisions.
Fscore _μ	$\frac{2 * Precision_{\mu} * Recall_{\mu}}{Precision_{\mu} + Recall_{\mu}}$	Relations between data's positive labels and the ones given by the classifier based on sums of individual decisions
Precision _M	$\frac{\sum_{i=1}^k \left(\frac{t_{pi}}{t_{pi} + f_{pi}} \right)}{k}$	An average per-class agreement of data class labels with those of classifiers
Recall _M	$\frac{\sum_{i=1}^k \left(\frac{t_{pi}}{t_{pi} + f_{ni}} \right)}{k}$	An average per-class effectiveness of a classifier to classify class labels
Fscore _M	$\frac{2 * Precision_{M} * Recall_{M}}{Precision_{M} + Recall_{M}}$	Relation between data's positive labels and the ones from the classifier based on a per-class average

References

- Famurewa, S.M. Maintenance Analysis and Modelling for Enhanced Railway Infrastructure Capacity. Ph.D. Thesis, Lulea University of Technology, Lulea, Sweden, 2015. Available online: <http://urn.kb.se/resolve?urn=urn%3Anbn%3Ase%3Altu%3Adiva-17115> (accessed on 16 February 2021).
- Whiteing, T.; Menaz, B. Thematic Research Summary: Rail Transport. *Transp. Res. Knowl. Cent. Eur. Comm.* **2009**. Available online: http://www.eurosfair.pr.d.fr/7pc/doc/1257761907_trs_rail_transport_2009.pdf (accessed on 17 February 2021).
- Trafikverket. *Prognosis of Swedish Goods Flow for the Year 2050 (Prognos Över Svenska Godsströmmar År 2050: Underlagsrapport. Publikationsnummer: 2012:112)*; Technical Report; Trafikverket: Borlänge, Sweden, 2012; ISBN 978-91-7467-310-4.
- Commission, E.; Roadmap to a Single European Transport Area—Towards a competitive and resource efficient transport system. *Eur. Comm.* **2011**. Available online: https://ec.europa.eu/transport/themes/european-strategies/white-paper-2011_en (accessed on 17 February 2021).
- Patra, A.P.; Kumar, U.; Kráik, P.O.L. Availability target of the railway infrastructure: An analysis. In Proceedings of the 2010 Proceedings—Annual Reliability and Maintainability Symposium (RAMS), San Jose, CA, USA, 25 January 2010; pp. 1–6. [CrossRef]
- Filcek, G.; Gašior, D.; Hojda, M.; Józefczyk, J. An Algorithm for Rescheduling of Trains under Planned Track Closures. *Appl. Sci.* **2021**, *11*, 2334. [CrossRef]
- Aytekin, Ç.; Rezaeitabar, Y.; Dogru, S.; Ulusoy, I. Railway fastener inspection by real-time machine vision. *IEEE Trans. Syst. Man Cybern. Syst.* **2015**, *45*, 1101–1107. [CrossRef]
- Feng, H.; Jiang, Z.; Xie, F.; Yang, P.; Shi, J.; Chen, L. Automatic fastener classification and defect detection in vision-based railway inspection systems. *IEEE Trans. Instrum. Meas.* **2013**, *63*, 877–888. [CrossRef]
- Wang, S.; Dai, P.; Du, X.; Gu, Z.; Ma, Y. Rail fastener automatic recognition method in complex background. In Proceedings of the Tenth International Conference on Digital Image Processing (ICDIP 2018), Shanghai, China, 9 August 2018; Volume 10806, p. 1080625. [CrossRef]
- Marino, F.; Distanto, A.; Mazzeo, P.L.; Stella, E. A real-time visual inspection system for railway maintenance: Automatic hexagonal-headed bolts detection. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **2007**, *37*, 418–428. [CrossRef]
- Stella, E.; Mazzeo, P.; Nitti, M.; Cicirelli, G.; Distanto, A.; D'Orazio, T. Visual recognition of missing fastening elements for railroad maintenance. In Proceedings of the IEEE 5th International Conference on Intelligent Transportation Systems, Singapore, 9 July 2003; pp. 94–99. [CrossRef]
- Yang, J.; Tao, W.; Liu, M.; Zhang, Y.; Zhang, H.; Zhao, H. An efficient direction field-based method for the detection of fasteners on high-speed railways. *Sensors* **2011**, *11*, 7364–7381. [CrossRef]
- De Ruvo, P.; Distanto, A.; Stella, E.; Marino, F. A GPU-based vision system for real time detection of fastening elements in railway inspection. In Proceedings of the 16th IEEE International Conference on Image Processing (ICIP), Cairo, Egypt, 7–10 November 2009; pp. 2333–2336. [CrossRef]
- De Ruvo, G.; De Ruvo, P.; Marino, F.; Mastronardi, G.; Mazzeo, P.L.; Stella, E. A FPGA-based architecture for automatic hexagonal bolts detection in railway maintenance. In Proceedings of the Seventh International Workshop on Computer Architecture for Machine Perception (CAMP'05), Palermo, Italy, 4–6 July 2005; pp. 219–224. [CrossRef]

15. Xia, Y.; Xie, F.; Jiang, Z. Broken railway fastener detection based on AdaBoost algorithm. In Proceedings of the 2010 International Conference on Optoelectronics and Image Processing, Haikou, China, 11–12 November 2010; Volume 1, pp. 313–316. [CrossRef]
16. Rubinsztejn, Y. Automatic detection of objects of interest from rail track images. Master's Thesis, Faculty of Engineering and Physical Science, University of Manchester, Manchester, UK, 2011. Available online: http://studentnet.cs.manchester.ac.uk/resources/library/thesis_abstracts/BkgdReportsMSc11/Rubinsztejn-Yohann-bkgd-rept.pdf (accessed on 19 February 2021).
17. Li, Y.; Otto, C.; Haas, N.; Fujiki, Y.; Pankanti, S. Component-based track inspection using machine-vision technology. In Proceedings of the 1st ACM International Conference on Multimedia Retrieval, Trento, Italy, 17–20 April 2011; pp. 1–8. [CrossRef]
18. Fan, H.; Cosman, P.C.; Hou, Y.; Li, B. High-speed railway fastener detection based on a line local binary pattern. *IEEE Signal Process. Lett.* **2018**, *25*, 788–792. [CrossRef]
19. Mazzeo, P.L.; Ancona, N.; Stella, E.; Distanto, A. Visual recognition of hexagonal headed bolts by comparing ICA to wavelets. In Proceedings of the 2003 IEEE International Symposium on Intelligent Control, Houston, TX, USA, 8 October 2003; pp. 636–641. [CrossRef]
20. Mandriota, C.; Nitti, M.; Ancona, N.; Stella, E.; Distanto, A. Filter-based feature selection for rail defect detection. *Mach. Vis. Appl.* **2004**, *15*, 179–185. [CrossRef]
21. Singh, M.; Singh, S.; Jaiswal, J.; Hempshall, J. Autonomous rail track inspection using vision based system. In Proceedings of the IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety, Alexandria, VA, USA, 16–17 October 2006; pp. 56–59. [CrossRef]
22. Wei, X.; Yang, Z.; Liu, Y.; Wei, D.; Jia, L.; Li, Y. Railway track fastener defect detection based on image processing and deep learning techniques: A comparative study. *Eng. Appl. Artif. Intell.* **2019**, *80*, 66–81. [CrossRef]
23. Liu, J.; Huang, Y.; Zou, Q.; Tian, M.; Wang, S.; Zhao, X.; Dai, P.; Ren, S. Learning visual similarity for inspecting defective railway fasteners. *IEEE Sens. J.* **2019**, *19*, 6844–6857. [CrossRef]
24. Song, Q.; Guo, Y.; Yang, L.; Jiang, J.; Liu, C.; Hu, M. High-speed railway fastener detection and localization system. *arXiv* **2019**, arXiv:1907.01141.
25. Chandran, P.; Rantatalo, M.; Odelius, J.; Lind, H.; Famurewa, S.M. Train-based differential eddy current sensor system for rail fastener detection. *Meas. Sci. Technol.* **2019**, *30*, 125105. [CrossRef]
26. Harris, C.R.; Millman, K.J.; van der Walt, S.J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N.J.; et al. Array programming with NumPy. *Nature* **2010**, *585*, 357–362. [CrossRef]
27. McKinney, W. Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference, Austin, TX, USA, 28–30 January 2010; Volume 445, pp. 51–56.
28. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Vanderplas, J. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830. Available online: <https://jmlr.org/papers/v12/pedregosa11a.html> (accessed on 20 February 2021).
29. Zhang, C.; Liu, C.; Zhang, X.; Almpantidis, G. An up-to-date comparison of state-of-the-art classification algorithms. *Expert Syst. Appl.* **2017**, *82*, 128–150. [CrossRef]
30. Brown, I.; Mues, C. An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Syst. Appl.* **2012**, *39*, 3446–3453. [CrossRef]
31. Fernández-Delgado, M.; Cernadas, E.; Barro, S.; Amorim, D. Do we need hundreds of classifiers to solve real world classification problems? *J. Mach. Learn. Res.* **2014**, *15*, 3133–3181. Available online: <http://hdl.handle.net/10347/17792> (accessed on 21 February 2021).
32. Jones, S.; Johnstone, D.; Wilson, R. An empirical evaluation of the performance of binary classifiers in the prediction of credit ratings changes. *J. Bank. Financ.* **2015**, *56*, 72–85. [CrossRef]
33. Pal, M. Multiclass approaches for support vector machine based land cover classification. *arXiv* **2008**, arXiv:0802.2411.
34. Xu, H.; Zhou, J.; Asteris, P.G.; Jahed Armaghani, D.; Tahir, M.M. Supervised Machine Learning Techniques to the Prediction of Tunnel Boring Machine Penetration Rate. *Appl. Sci.* **2019**, *9*, 3715. [CrossRef]
35. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [CrossRef]
36. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
37. Schapire, R.E. The boosting approach to machine learning: An overview. *Nonlinear Estim. Classif.* **2003**, 149–171. [CrossRef]
38. Freund, Y.; Schapire, R.E. Experiments with a new boosting algorithm. In Proceedings of the 13th International Conference on Machine Learning, Bari, Italy, 3–6 July 1996; Volume 96, pp. 148–156.
39. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, 1189–1232. Available online: <https://www.jstor.org/stable/2699986> (accessed on 24 February 2021).
40. Japkowicz, N.; Shah, M. *Evaluating Learning Algorithms: A Classification Perspective*; Cambridge University Press: Cambridge, UK, 2011.
41. Sokolova, M.; Lapalme, G. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437. [CrossRef]
42. Rachman, A.; Ratnayake, R.C. Machine learning approach for risk-based inspection screening assessment. *Reliab. Eng. Syst. Saf.* **2019**, *185*, 518–532. [CrossRef]