

Feature Level Sensor Fusion for Improved Fault Detection in MCM Systems for Ocean Turbines

Janell Duhaney Taghi M. Khoshgoftaar John C. Sloan

Florida Atlantic University

777 Glades Rd., Boca Raton, FL 33431

jduhaney1@fau.edu; khoshgof@fau.edu; jsloan11@fau.edu

Abstract

This paper investigates feature level fusion for enhancing fault detection from vibration signals in an ocean turbine. Changes in vibration signatures from such rotating machinery typically indicate the presence of a problem such as a shift in its orientation or mechanical impact from its environment. We applied feature level fusion to vibration data acquired from two accelerometers attached to a box fan, and then assessed the abilities of twelve well known machine learners to detect changes in state from the raw accelerometer data and from the fused data. Analysis of the performance of these classifiers showed an overall performance improvement in all twelve classifiers in detecting the state of the fan from the fused data versus from the data from the two individual sensor channels.

Introduction

Ocean turbines harvest the flow of ocean currents to offer a promising alternative for clean and renewable energy. They operate autonomously in varying and sometimes harsh environmental conditions, and are required to produce a constant output while satisfying uptime requirements. Detecting problems as soon as they occur minimizes damage to the turbine, but is only possible through uninterrupted monitoring. Frequent manual inspections are infeasible due to high expeditionary costs to access the machines, and problems which occur between inspection intervals can go undetected until the next maintenance visit.

Machine condition monitoring (MCM) systems provide a means for continuous and intelligent problem detection within complex systems such as ocean turbines. Such systems allow for constant error checking and require minimal human intervention. They utilize a suite of heterogeneous sensors to monitor different physical phenomena such as oil quality, external and internal temperatures, and vibration. In this setting, data mining and machine learning techniques can help automate fault detection, identify failure states, and extract patterns in operational state and environment from the massive amount of data generated from the sensors. These techniques can also predict life and future health of the machine. Sensor fusion techniques are needed

in such systems to combine sensor data at different stages of the monitoring process, including state detection, health assessment, and advisory generation, to produce more accurate and complete results.

This paper focuses on employing feature-level sensor fusion to enhance the performance of machine learners when detecting operational state. This approach, known as feature level fusion, has been investigated in many other domains such as image fusion (Samadzadegan 2004) (Sharma & Davis 2006), protein classification (Zhang *et al.* 2006), land mine detection (Gunatilaka & Baertlein 2001), and biometrics (Kong, Zhang, & Kamel 2006). To the author's knowledge, this is the first study related to feature level fusion for vibration analysis of ocean turbines.

A case study discussed in this paper shows how feature level fusion can be used for more reliable state detection from vibration data gathered from rotating machinery, and demonstrates the use of data mining and machine learning techniques for classifying operational state. It includes results of analyses performed on experimental data gathered from a box fan, whose rotating blades produce vibration signatures which can be mined to determine its state as would be possible from an ocean turbine.

Related Work

The increase in world energy consumption over the past years, and the growing concerns for waning fossil fuel reserves and the environmental concerns associated with them, sparked a worldwide initiative for finding clean, renewable alternative energy sources. One such alternative involves using turbines to extract energy from the steady unidirectional flow of ocean currents such as the Gulf Stream. It is estimated, for example, that if only 0.1% of the potential energy in the Gulf Stream is captured, it would be able to satisfy 35% of the energy demand in Florida, U.S.A. (Minerals Management Service 2006).

Research into harvesting ocean current energy from the Gulf Stream is underway by the Southeast National Marine Renewable Energy Center at the Florida Atlantic University where a 20-kilowatt tidal turbine prototype is being developed for that purpose (Beaujean *et al.* 2009). In this prototype, the turbine is housed within a pressurized enclosure called a nacelle which is connected to a pressure buoy (to control its pitch, yaw and roll) and a three blade propeller.

This structure, shown in Figure 1 and depicted as component (e) in Figure 2, is connected via cabling to a monitoring and control buoy (b), and a barge for keeping the system upright (c), and is tethered to the ocean floor (d). The northbound flow of the Gulf Stream is shown as (a) in Figure 2.

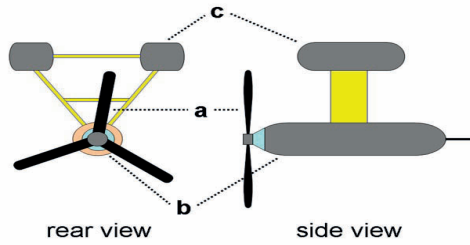


Figure 1: Closeup of nacelle

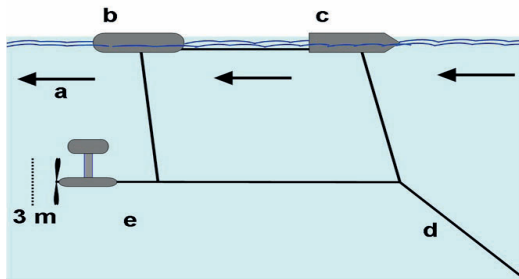


Figure 2: Turbine and its moorings

To harness underwater ocean currents, ocean turbines operate unattended below the ocean's surface, which in itself creates unique reliability concerns (Beaujean *et al.* 2009). Reliability is a particularly important issue due to the high costs associated with accessing, retrieving and maintaining the turbine. Some of the reliability issues related to development, maintenance and deployment of an ocean turbine are:

1. Bio-Fouling - While submerged, the ocean turbine is susceptible to biological fouling, or bio-fouling, which is the gradual but undesired accumulation of animals and plants on the turbine. Sensors on a buoyancy-driven underwater glider developed in 2003 as a part of the UCSD Spray Project to observe oceanographic features stopped functioning within a mere four weeks of deployment in the Monterey Canyon due to bio-fouling (Sherman *et al.* 2001).
2. Corrosion - The salinity of the ocean water behaves as a corrosive agent for parts of the turbine, with cabling being an easy target. Destruction of cabling leads to a loss of communication between the turbine and any components on the ocean surface, while corrosion of the nacelle could eventually cause a breach.
3. Turbidity - Oceanic wildlife or debris could impact or ob-

struct the turbine. Also, larger objects could tilt the turbine or cause it to lean on its mooring line.

To increase the reliability of ocean turbines and other complex machinery, machine condition monitoring (MCM) systems are used to perform continuous, automated self-checking. They record, manage, process and interpret readings from a suite of sensors to offer intelligent problem detection capabilities (i.e. automated fault localization, detection and classification). By providing information about the type and location of a fault, an operator can determine whether a maintenance expedition is necessary and exactly which tools or parts are required to correct the problem. Also, detecting faults as soon as they occur allows for quick remediation (such as adjustment or shutdown) which prevents damage to the turbine. Overall, MCM systems can minimize operation and repair-related costs, while maximizing the output and life of the turbine.

An MCM system for an ocean turbine could include sensors to measure oil level and quality, temperature, turbidity, electrical output, rotational velocity, and vibration. Some sensors such as oil and temperature sensors produce a single reading at predetermined intervals. Others, like the vibration sensor, are capable of continuously emitting waveform measurements in a data stream. All the data gathered from these sensors must be combined or fused and then interpreted to provide accurate information about the turbine's environment, current state and future health.

In the case study discussed in the next section, a combination of data mining and sensor fusion is used to identify problem states as they occur. Data mining and machine learning, which collectively refer to techniques for inferring knowledge from raw data by analyzing patterns, provides an avenue for automated interpretation of the sensor data and problem classification, while sensor fusion techniques are needed to combine data from multiple sources to get a complete, more accurate picture. These techniques would work collaboratively within the health assessment process of an MCM system to allow for complete, tested and automatic interpretation of the raw data.

Approaches to sensor fusion can be divided into categories based on the level at which they are performed. These levels are:

1. Data level fusion: Data level fusion (sometimes called pixel-level fusion in image fusion) involves combining raw sensor signals prior to performing any data transformations, feature extraction or data manipulation. In order to combine sensor signals at the data level, the signals must originate from sources which produce the same type of signal.
2. Feature level fusion: Feature level fusion, which is the focus of this paper, involves first extracting features or attributes which describe the data, and then combining the features from each signal to produce a fused signal. Unlike data level fusion, feature level fusion can be applied to data from both homogeneous and heterogeneous sensor types.
3. Decision level fusion: Decision level fusion (Chen, Wang, & Chi 1997) involves making a local decision from

each signal and then combining the decisions to get the final output. Another approach to decision level fusion (Veeramachaneni *et al.* 2008) is to apply a mining algorithm that determines the probability of each possible outcome from each signal, and then combine the probability of class membership for each class using summing or voting. The final decision is made by selecting the outcome with the highest probability.

A comparison of the different fusion levels on vibration data remains for future work. Previous studies performing this type of comparison were specific to different application domains (Samadzadegan 2004).

In some domains, the real challenge behind feature level fusion is the extraction or determination of appropriate features which best describe the object. For our application, features are derived from a wavelet transformation which, given a time series of amplitudes, produces output which tells the frequency at which an oscillation is detected. Wavelet analysis provides distinct advantages over other preprocessing methods such as Cepstrum analysis and Fourier transforms: it works over multiple scales, and is highly efficient on streaming data (Peng & Chu 2004). The wavelet transform used was a discrete Haar wavelet, which we implemented by applying the same methodology as (Wald *et al.* 2010). Due to space limitations, additional details about the transform have been omitted.

To fuse the sensor data at the feature level, a set union of the features produced by the wavelet transform from both channels was performed, which, intuitively, should improve a classifier's ability to perform state detection since all the available data is being taken into account during the data mining process.

Our case study investigates feature level fusion of wavelet transformed vibration data. It is meant to show how feature level fusion could improve the ability of machine learning algorithms to detect the orientation of a turbine based on data from its vibration sensors, known as accelerometers. As the researchers did not yet have access to vibration data from the ocean turbine at the time this study was conducted, two accelerometers were mounted on a box fan and readings were taken while the fan was running in three different orientations. Although a fan runs at a much higher speed than a turbine, the rotation of its blades produces different vibration signatures depending on varying operating conditions, as would a turbine.

Experimental Setup

Vibration readings of rotating machinery contain distinct signatures which can be used to determine the state of a machine. In this experiment, feature level fusion and data mining are applied to vibration data acquired from multiple sensors installed on a box fan. Like a turbine, the rotation of the blades in a box fan result in vibration patterns that are representative of the state of the fan. The motivation behind this experiment is to show, using experimental data, how feature level fusion enhances problem detection in rotating machinery by reducing misclassification rates, i.e. the amount of incorrect state predictions. The lower the mis-

classification rate, the more confidence one can have in the classification. Although this experiment does not represent a complete MCM system, it demonstrates how intelligent problem detection occurs within such a system using sensor fusion and data mining.

For this case study, measurements were recorded from the fan while it was operating at the same speed (1010 RPMs) in four different setups: standing upright (baseline), tilted on a soft surface like a hand (TOH), tilted on a hard surface such as a wall (TOW), and slowed with an object like a pencil (SWO). These four experiments correlate to four scenarios possible while an ocean turbine is submerged – running normally, tilted on its mooring line, tilted on a submerged object like a piling and obstructed by debris. The baseline state is considered the normal class, while TOH, TOW and SWO are the faults or problem scenarios that the data mining classifiers will try to detect.

The data used in this case study were acquired from two identical accelerometers (model AC136-1A) installed on the outer casing of a typical 50cm 120V AC box fan by an IO-Tech Wavebook/516-E Data Acquisition Unit (DAQ). Time synchronous averaging (Lebold *et al.* 2000) of the vibration signals is also performed by the DAQ. This process segments the data into equal length blocks related to the different rotational phases and averages the blocks to reduce noise.

Data from the accelerometers were sampled at 1000 Hz for 3 seconds for each experiment, producing a total of 3000 readings per burst. These accelerometers are denoted as channels 1 and 2 (CH1 and CH2) throughout this case study. Each experiment was repeated a total of six (6) times, resulting in 18,000 measurements per experiment. The raw data consisted of 48 files = 2 channels x 6 runs x 4 setups. The six runs of each experiment were combined to form 8 files = 2 channels x 4 setups.

The measurements recorded from both accelerometers during these experiments were already synchronized and complete with no data points missing. Also, the number of baseline data points was the same as the number of data points in each scenario of interest. For fusion of vibration data from a live turbine, sensor synchronization may be necessary to align measurements taken at the same instance in time since readings may arrive late or out of order due to problems during transmission. Data imputation techniques (Van Hulse & Khoshgoftaar 2008) may also be utilized to fill in missing data values, which is possible due to packet loss during transmission and/or sensor malfunction.

The data in each of the 8 files were passed through a discrete Haar wavelet transform process similar to the one used in (Wald *et al.* 2010) which converts the time series of raw amplitude readings to a time-frequency representation of the signal. The output of the wavelet transform are 10 nominal features, whose value can be either 0 or 1. A value of 0 indicates that no wave was detected at a given scale and time, and a 1 means that a wave was detected at the given scale and time.

In this study, feature level fusion is done by performing a union of the features across both sensor channels for each setup. The 10 wavelet features from channel 1 were com-

bined with the 10 wavelet features from channel 2 for each setup, producing 4 new files, each containing 20 features. Counting the 8 files from the individual channels and the 4 files with the combined features, there are now 12 data files = (2 channels x 4 setups) + (4 setups with 20 features).

A sliding window (Lee, Lin, & Chen 2001) of size 100 (i.e. there are 100 instances in each window) was then applied to the transformed data in each of the 12 data files by taking the arithmetic sum of the 1s for each feature over the length of the window. The window size was selected based on experimentation with other window sizes; due to space limitations, the results of the window size experiment were not included.

For this experiment, twelve (12) machine learning techniques were trained to detect the underlying patterns in the vibration signatures and to predict the state of the machine. This problem was reduced to a binary classification problem by combining the data from each of the three faults (TOH, TOW and SWO) with the data from the baseline setup for each channel. In other words, the decision a classifier needed to make in each case was between the normal class (baseline) and a single fault (either SWO, TOH or TOW). Three models were built for each machine learner (or classifier) and fault type by training the classifier on data from each accelerometer independently and from the fused data. The classifiers used in this case study are all available in WEKA¹ data mining software package (Witten & Frank 2005). These models were built in WEKA by performing five-fold cross-validation. The twelve classifiers are listed on page 6 (Table 1) and the results for all twelve learners are presented in the next section. Default parameter values were used unless otherwise noted. Non-default parameter values were used only where experimentation indicated an overall improvement in classification performance for all channels. Details and results of individual experiments were excluded due to space limitations.

In a binary classification problem, the overall performance of the classifier is summarized in terms of a confusion matrix – a 2 x 2 matrix showing the number of data points correctly labeled as either faulty or baseline, as well as the number of items misclassified. The number of faulty instances correctly identified is called the True Positive Rate (TPR), and the number of instances that were actually normal but were labeled as faulty by the classifier is the False Positive Rate (FPR).

The performance measure used for this study was the AUC, or area under the Receiver Operating Characteristic (ROC) curve. The ROC curve is the graph of the FPR on the x-axis versus the TPR on the y-axis. The AUC summarizes the information provided by the ROC curve as a single numeric value between 0 and 1, with larger values being better, allowing for quicker analysis and a concise representation of classification performance. In addition, the AUC correctly expresses classification performance regardless of the class distribution, and is independent of the decision threshold (Huang & Ling 2005).

¹ Available for download from <http://www.cs.waikato.ac.nz/ml/weka>

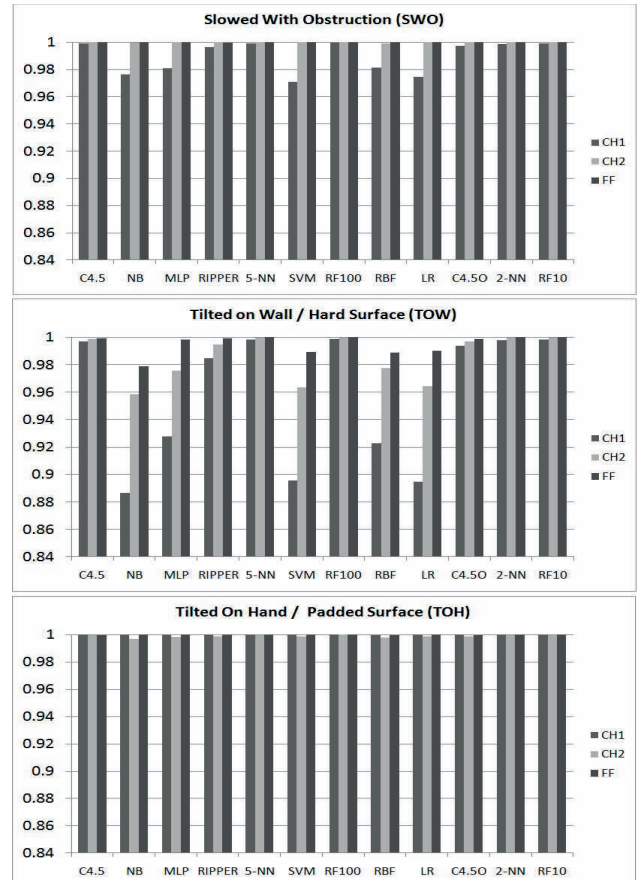


Figure 3: Results

Results

The results of the SWO, TOH and TOW experiments are shown in the three clustered bar graphs in that order in Figure 3. In the graphs for each fault, the performance of the twelve classifiers on distinguishing that fault from data from independent channels (CH1 and CH2) and the fused channel (denoted FF) is presented. Each classifier is represented as a separate cluster of three vertical bars on the x-axis where the height of the bar is the AUC value; so, the higher the bar, the better that classifier was at detecting that fault. The three vertical bars comprising the cluster for each classifier represent the values obtained from CH1, CH2 and FF data respectively.

For the SWO fault, classifier performance from CH2 and FF data was almost perfect but dipped as low as 0.97 AUC for some learners on CH1 data. So, for SWO, the results CH1 and FF data yielded similar classification performance for detecting the TOH fault, and results for CH2 data were a little worst for some learners. Classifier performance were lowest on TOW, where CH1 data produced the worst performances. Although classifiers performed better on CH2 for detecting the TOW fault, the best results were obtained from the FF data. The degradation of classifier performances from TOW data across all channels may be due to increased noise levels from wall resonance.

While its effect on classifier performance was less apparent for the SWO and TOH faults, the feature level fused channel FF showed the highest AUCs of the three channels (CH1, CH2, FF) across all setups. In addition, results for TOW were best on the FF channel than on any individual channel. So, by applying feature level fusion, classifiers were able to distinguish faults with greater confidence and lower misclassification rates.

Conclusion

Feature level fusion can be applied within an MCM system to improve problem detection from vibration data. This was confirmed by experimental results from a case study which showed that for each of three faults to be detected, feature level fusion provided either similar results as the better accelerometer channel or better results than both individual channels. Future work includes comparing feature level fusion against data level and decision level fusion approaches, and further application of feature level fusion to vibration data from other rotating machinery including an ocean turbine.

References

- Beaujean, P.-P.; Khoshgoftaar, T. M.; Sloan, J. C.; Xiros, N.; and Vendittis, D. 2009. Monitoring ocean turbines: a reliability assessment. In *Proceedings of the 15th ISSAT International Reliability and Quality in Design Conference*, 367–371.
- Breiman, L. 2001. Random Forests. In *Machine Learning*, 5–32.
- Buhmann, M. D., and Buhmann, M. D. 2003. *Radial Basis Functions*. New York, NY, USA: Cambridge University Press.
- Charalampidis, D., and Muldrey, B. 2009. Clustering using multilayer perceptrons. *Nonlinear Analysis: Theory, Methods & Applications* 71(12):e2807 – e2813.
- Chen, K.; Wang, L.; and Chi, H. 1997. Methods of combining multiple classifiers with different features and their applications to text-independent speaker identification. *International Journal of Pattern Recognition and Artificial Intelligence* 11:417–445.
- Cohen, W. W. 1995. Fast Effective Rule Induction. In *In Proceedings of the Twelfth International Conference on Machine Learning*, 115–123.
- Fraiman, R.; Justel, A.; and Svarc, M. 2010. Pattern recognition via projection-based kNN rules. *Comput. Stat. Data Anal.* 54(5):1390–1403.
- Frank, E.; Trigg, L.; Holmes, G.; and Witten, I. H. 2000. Technical note: Naive Bayes for Regression. *Machine Learning* 41(1):5–25.
- Gunatilaka, A. H., and Baertlein, B. A. 2001. Feature-level and decision-level fusion of noncoincidently sampled sensors for land mine detection. *IEEE transactions on Pattern Analysis and Machine Intelligence* 23:577–589.
- Huang, J., and Ling, C. 2005. Using AUC and accuracy in evaluating learning algorithms. *Knowledge and Data Engineering, IEEE Transactions on* 17(3):299 – 310.
- Kong, A.; Zhang, D.; and Kamel, M. 2006. Palmprint identification using feature-level fusion. *Pattern Recognition* 39(3):478 – 487.
- Lebold, M.; McClintic, K.; Campbell, R.; Byington, C.; and Maynard, K. 2000. Review of vibration analysis methods for gearbox diagnostics and prognostics. *Proceedings of the 54th Meeting of the Society for Machinery Failure Prevention Technology* 623–634.
- Lee, C.-H.; Lin, C.-R.; and Chen, M.-S. 2001. Sliding-window filtering: an efficient algorithm for incremental mining. In *Proceedings of the tenth international conference on Information and knowledge management, CIKM '01*, 263–270. New York, NY, USA: ACM.
- Minerals Management Service. 2006. Technology White Paper on Ocean Current Energy Potential on the U.S. Outer Continental Shelf.
- Peng, Z. K., and Chu, F. L. 2004. Application of the wavelet transform in machine condition monitoring and fault diagnostics: a review with bibliography. *Mechanical Systems and Signal Processing* 18(2):199 – 221.
- Platt, J. C. 1998. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines.
- Samadzadegan, F. 2004. Data integration related to sensors, data and models. In *ISPRS*, number 4 in 35, 569 – 574. Natural Resources Canada.
- Sharma, V., and Davis, J. W. 2006. Feature-level fusion for object segmentation using mutual information. In *In IEEE Int. Wkshp. on Object Tracking and Classification Beyond the Visible Spectrum*.
- Sherman, J.; Davis, R.; Owens, W.; and Valdes, J. 2001. The autonomous underwater glider "Spray". *Oceanic Engineering, IEEE Journal of* 26(4):437 – 446.
- Van Hulse, J., and Khoshgoftaar, T. M. 2008. A comprehensive empirical evaluation of missing value imputation in noisy software measurement data. *J. Syst. Softw.* 81(5):691–708.
- Veeramachaneni, K.; Osadciw, L.; Ross, A.; and Srinivas, N. 2008. Decision-level fusion strategies for correlated biometric classifiers. In *Proc. of IEEE Computer Society Workshop on Biometrics at the Computer Vision and Pattern Recognition (CVPR) conference*, 1–6.
- Wald, R.; Khoshgoftaar, T. M.; Beaujean, P.-P.; and Sloan, J. C. 2010. Combining wavelet and fourier transforms in reliability analysis of ocean systems. In *Proceedings of the 16th ISSAT International Reliability and Quality in Design Conference*, 303 – 307.
- Witten, I. H., and Frank, E. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. San Francisco: Morgan Kaufmann, 2 edition.
- Zhang, S.-W.; Pan, Q.; Zhang, H.-C.; Shao, Z.-C.; and Shi, J.-Y. 2006. Prediction of protein homo-oligomer types by pseudo amino acid composition: Approached with an improved feature extraction and naive bayes feature fusion. *Amino Acids* 30:461–468. 10.1007/s00726-006-0263-8.

Learner	Label	Description
Tuned C4.5 Decision Tree	C4.5	This algorithm (Witten & Frank 2005) derives a set of classification rules from the training data and uses these rules to classify or label new instances. J48, the WEKA implementation of the C4.5 decision tree algorithm, was constructed with pruning disabled, Laplace smoothing enabled, and default values for remaining parameters.
Naive Bayes	NB	A simplified form of a Bayesian network, the Naive Bayes learner (Frank <i>et al.</i> 2000) predicts the probability of each outcome based on the naive assumption of independence among the predictive features and selects the outcome with the highest probability as its prediction.
Multi-Layer Perceptron	MLP	The Multi-Layer Perceptron (MLP) Neural Network (Charalampidis & Muldrey 2009) is a form of feed-forward neural network which maps input values to an output. This learner was built in WEKA using default values for all parameters except for the 'hiddenLayers' and 'validationSetSize' parameters which were set to '3' and '10' respectively.
RIPPER	RIPPER	RIPPER, or Repeated Incremental Pruning to Produce Error Reduction, (Cohen 1995) is a rule based learner which generates a decision on new instances based on a set of rules built from a training dataset. For this case study, the default parameters for the JRip algorithm – the WEKA implementation of RIPPER – was used.
k-Nearest Neighbor with k=5	5-NN	The k-Nearest Neighbor algorithm (Fraiman, Justel, & Svarc 2010) classifies a new instance by doing a majority vote of the classes of k instances in the training dataset that are closest to the new instance within the feature space. Default values were selected for all parameters of the IBk algorithm, the WEKA implementation of the k-Nearest Neighbor algorithm, with the exception of the value of k which was set to 5.
Support Vector Machine	SVM	The simplest form of the Support Vector Machine (SVM) is a hyperplane which divides a set of instances into two classes with maximum margin. The SVM in WEKA is implemented as John Platt's SMO algorithm (Platt 1998). Default values were used for all parameters.
Random Forest with 100 trees	RF100	Random Forest (Breiman 2001) is an ensemble learner composed of multiple decision trees. This is the WEKA implementation of the Random Forest learner with the number of trees parameter value set to 100.
Radial Basis Function Neural Network	RBF	The WEKA implementation of the Radial Basis Function Network (Buhmann & Buhmann 2003) is a normalized Gaussian form of the neural network. No changes were made to the default WEKA parameter values for this learner.
Logistic Regression	LR	The Logistic Regression learner (Witten & Frank 2005) is implemented in WEKA with a multinomial regression model for minimizing error. No parameter values were changed for this case study.
C4.5 Decision Tree	C4.5O	This is the WEKA implementation of the J48 algorithm with default parameters.
k-Nearest Neighbor with k=2	2-NN	The WEKA implementation of the k-NN algorithm with k=2.
Random Forest with 10 trees	RF10	The WEKA implementation of the Random Forest learner with the number of trees parameter set to 10.

Table 1: Machine Learners