ITcon
www.itcon.org - Journal of Information Technology in Construction - ISSN 1874-4753

MACHINE LEARNING APPROACHES TO DETERMINING TRUCK TYPE FROM BRIDGE LOADING RESPONSE

SUBMITTED: January 2023
REVISED: June 2023
PUBLISHED: July 2023
EDITOR: Robert Amor
DOI: 10.36680/j.itcon.2023.018

Yueren Wang, Associate Professor
Guangzhou University,
wangyueren1@126.com

Ian Flood, Professor,
University of Florida,
flood@ufl.edu

SUMMARY: *The paper is concerned with the development and comparison of alternative machine learning methods of determining the type of truck crossing a bridge from the dynamic response it induces within the bridge structure, the so-called weigh-in-motion problem. Weigh-in-motion is a rich engineering problem presenting many challenges for current machine learning technologies, and for this reason is proposed as a benchmark for guiding and assessing advances in the application of this field of artificial intelligence. A review is first provided of existing methods of determining truck types and loading attributes using both machine learning and heuristic search techniques. The most promising approach to date, that of artificial neural networks, is then compared to support vector machines in a comprehensive study considering a range of configurations of both modeling techniques. A local scatter point smoothing schema is adopted as a means of selecting an optimal set of design parameters for each model type. Three main model formats are considered: (i) a monolithic model structure with a one-versus-all truck type classification strategy; (ii) an array of sub-models each dedicated to one truck type with a one-versus-all classification strategy; and (iii) an array of sub-models each dedicated to selecting between pairs of trucks in a one-versus-one classification strategy. Overall, the formats that used an array of sub-models performed best at truck classification, with the support vector machines having a slight edge over the artificial neural networks. The paper concludes with some suggestions for extending the work to a broader scope of problems.*

KEYWORDS: *Artificial Neural Network; Empirical Modeling; Machine Learning, Support Vector Machine; Truck Type Classification; Weigh-In-Motion.*

REFERENCE: *Yueren Wang, Ian Flood (2023). Machine learning approaches to determining truck type from bridge loading response. Journal of Information Technology in Construction (ITcon), Vol. 28, pg. 346-359, DOI: 10.36680/j.itcon.2023.018*

1 INTRODUCTION

Machine learning is a branch of artificial intelligence (AI) concerned with developing computational devices that learn to respond to a problem based on experience rather than through direct programming. It is a form of empirical modeling in that it develops a solution from observations of the response of the system being modeled or from observations of the behavior of an analog of that system. It is widely used in fields such as science and engineering when there is a lack of theory describing the relationship between the system variables, but it can also be used as a substitute for theoretically derived models when they are found to be too slow computationally (Flood & Issa, 2010). Among the most common examples of machine learning devices are Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs).

There are several outstanding concerns with machine learning solutions, as with all empirically derived models, that need to be resolved before the full potential of the approach can be realized. Most notably, these are:

- **lack of insight** - they are black box devices that provide little or no explanation of the rationale behind the solution provided;
- **limited ability to extrapolate** – generally, they do not extend to versions of a problem beyond the scope of the observations used to develop the model (the input variables are value-constrained);
- **lack of extensibility** - they cannot be adjusted easily to operate beyond the set of input variables built into the initial model (the model structure is variable-constrained) – any extension requires major redevelopment of the model;
- **rigid input format** - the presentation of input values must match the exact format used when the model was developed – this means, for example, that spatially and temporally distributed input values cannot be translated within that space; and
- **excessively large training datasets** - the number of observations required to develop a model tends to increase geometrically with the number of input variables required, the so called '*curse of dimensionality'* (Bellman, 2003) – this is perhaps the most restrictive factor since, in practical terms, this limits model complexity to 5 or 6 input variables if those variables are perfectly uncorrelated (Flood & Issa, 2010).

While these concerns are particularly significant in engineering applications they do not render machine learning tools unusable and there is no indication they cannot be resolved or circumvented. As such they should be seen as challenges to the application of machine learning rather than as fundamental limitations. An appropriate set of benchmark problems can provide an effective way of directing and measuring progress in the research and development of a technology. For machine learning, character recognition problems (such as classifying the examples in the MNIST handwritten digit database (LeCun et al., 2017)) have provided a popular benchmark, however, they do not directly address issues such as model extensibility or growth in the training dataset size with respect to problem complexity. A good benchmark engineering problem for machine learning is truck weigh-in-motion (WIM) where the goal is to estimate the attributes of a fast moving truck (such as, its axle configuration, axle spacings, and axle loads) from measurements of the strain response of the structure over which it is traveling. An accurate WIM system has many potential applications including allowing comprehensive statistics on truck-bridge loading to be obtained for use in highway bridge design or fatigue rating of existing bridges. WIM is a rich problem exhibiting many of the challenges that machine learning techniques have difficulty resolving, such as, input vector translation (resulting from, for example, event timing, velocity and acceleration of a truck), model extensibility (to take into account alternative bridge configurations, lane configurations, multiple/concurrent truck passage events, and truck axle configurations), and a geometric explosion in the number of training patterns relative to problem complexity (Flood & Issa, 2010).

This paper first reviews and compares the main modeling approaches that have been developed for solving the WIM problem, including a numeric simulation based heuristic search approach and ANN based empirical approaches, identifying the advantages and limitations of each. The paper then compares the most promising of these approaches (ANN-based empirical methods) to an alternative machine learning approach that has had a lot of success in competing with ANNs, namely support vector machines (SVMs) (Cortes & Vapnik, 1995). A range of alternative strategies are considered for applying ANNs and SVMs to the truck classification aspect of the WIM problem. The performances of these approaches are then compared for a simply supported bridge problem to define a benchmark against which future developments in machine learning based WIM solutions can be compared. This is followed by an identification of the outstanding issues with current WIM solution methods.

2 APPROACHES TO THE WEIGH-IN-MOTION PROBLEM

The concept of WIM is illustrated in FIG. 1. When a truck crosses a bridge it induces a strain response in the structure which changes as truck axles move onto and pass across the bridge, as illustrated by the actual strain mapping (the dashed red curve in the figure). Numeric modeling techniques (for example, the Finite Element Method (FEM)) can be used to predict the magnitude and form of this strain-time envelope for given truck loading scenarios (the blue area in the figure), and is a critical part of the loading analysis of a bridge design. WIM is the reverse of the loading analysis problem in that it attempts to estimate the truck loading scenario from the actual strain-time envelope, as illustrated by mapping (3) in FIG. 1. However, whereas bridge loading analysis can be readily solved using numeric modeling techniques (such as FEM) WIM has no theoretically based solution method.

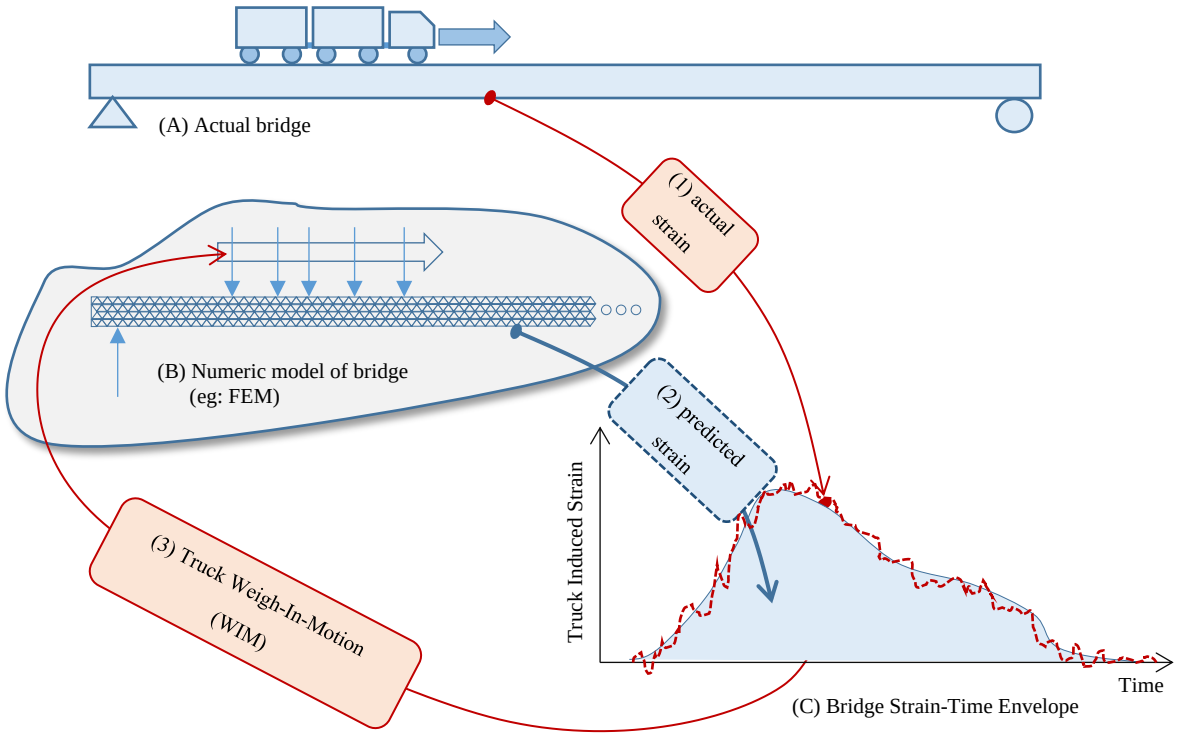


FIG. 1: The Truck-Bridge Weigh-In-Motion Problem.

2.1 Non-Machine Learning Solution Methods

Using non-machine learning based algorithms to solve WIM has attracted some attention, in particular with the advance of model regularization techniques. For example, Fitzgerald et al. (2017) used a novel moving force identification algorithm to predict axle weights. Instead of using machine learning based generalization technique such as hyper-parameter tuning and cross validation, this method uses traditional mathematical regularization techniques to prevent model overfitting when solving ill-posed problems. The approach can only solve for trucks with 2 axles and lacks the ability to extend to more complex axle configurations.

Vala et al. (2011) used a heuristic approach to solve the WIM problem based on the use of genetic algorithms (GA's). The GA would search for a truck loading scenario that could best explain the actual strain envelope. A numeric model was used to calculate the strain envelopes for the evolving truck loading scenarios. The fitness of a solution was measured by comparing the numerically derived strain envelope to the actual strain envelope. The GA approach was found to be reasonably accurate predicting axle loads to less than 15% error, depending on the axle position and the number of strain sampling points used. However, there was no indication that the approach could significantly outperform (in terms of accuracy of estimates) existing empirically derived solutions to the problem.

The main benefit of the GA approach is that it is not subject to the usual issues encountered by empirical modeling. For example, the approach could be extended to new versions of the problem (in this case to different bridge designs) simply by reconfiguring the numeric model. In addition, the approach does not require the use of training data to develop a solution and so the issue of a geometric explosion in the number of observations required as a function of problem complexity is not relevant.

The main problem with the GA approach is that it is computationally very slow. Each iteration in the GA process (moving from one generation of solutions to the next) requires all current solutions to be evaluated for fitness. The time required to determine fitness is dependent on the amount of processing required to execute the numeric simulation that maps a loading scenario to the strain envelope, which for two or even three dimensional models (necessary for modeling all but the simplest of bridge designs) can take many hours to complete (Farmaga et al. 2011). This duration must then be multiplied by the number of iterations in the search process, which typically is in the order of hundreds or thousands (McCall, 2005). Consequently, the GA approach does not provide a practical tool for solving non-trivial WIM problems given current computing technology, especially if the application requires the output of solutions in real-time.

2.2 Artificial Neural Networks Solution Methods

Early empirical modeling work by Gagarin et al. (1994) demonstrated the viability of using artificial neural networks (ANNs) to estimate truck attributes from bridge strain data. A two stage neural network system was considered, similar to that illustrated in FIG. 2. The first of the two stages (that shown to the left of the figure) was designed to classify a given truck loading condition and thus select an appropriate set of networks from the second level of the system. The laterally connected architecture shown at level 1 in the figure is typical of networks used for classification purposes, though conventional feedforward networks were used in the original study. The networks in the second level were designed to operate for a given truck loading class, providing estimates of velocity, axle spacings and axle loads.

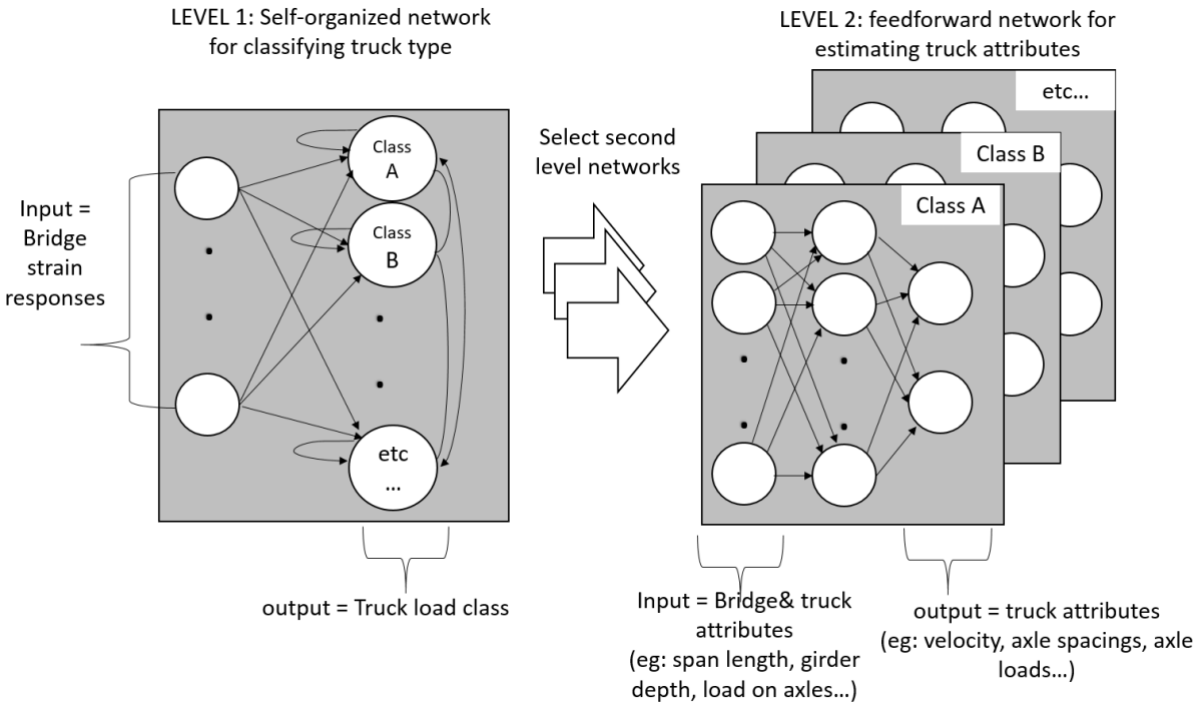


FIG. 2: Overall Structure of Modular ANN.

This modularized approach was adopted, in preference to a monolithic network, to facilitate the task of training the ANNs (Gagarin et al., 1994). In addition, it enabled individual modules to be retrained, and new modules added, as new training data became available, without the burdensome task of having to retrain the complete network system. However, this system is only as good as its weakest link. In particular, if the first level network misclassified the type of truck crossing the bridge, then the incorrect second level network would be selected and

the estimates would be completely invalid. The focus of much subsequent work, therefore, was to develop alternative more accurate modules for the classification stage of this system (level 1 in FIG. 2).

A variety of different types of supervised-training neural networks were considered for the first level in the network, the truck type classifier. The first of these was a radial-Gaussian feedforward networking system (RGIN) that uses an incremental supervised training algorithm (Flood & Kartam, 1998). The FHWA system of truck classification was used in the study, and is illustrated in FIG. 3 with a summary of the axle loading and spacing ranges summarized in Table 1 (see Gagarin et al, 1994). Input to the RGIN networks was an array of strain readings measured at the mid-span location on a girder of the bridge during the passage of a truck, as schematized in FIG. 4a. Each output from the RGIN network represented a different truck loading class. The class to which a given truck loading situation belonged was indicated by the output neuron that generates the value closest to 1.0 (all other outputs were trained to generate a value close to 0.0).

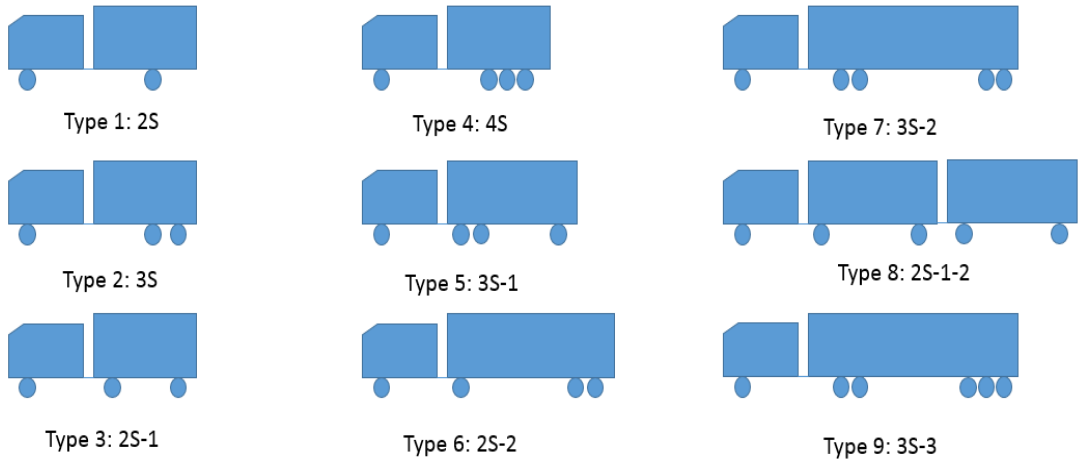


FIG. 3: Nine truck types used in this paper adapted from Gagarine et al. (1994)

Table 1: Axle Load and Spacing Range of Nine Truck Types adapted from Gagarin et al. (1994)

Truck Type	Axle Loads (KN)						Axle Spacings (m)				
	1	2	3	4	5	6	1 and 2	2 and 3	3 and 4	4 and 5	5 and 6
1	13.3-53.4	8.8-80.1					2.74-6.10				
2	13.3-53.4	8.8-80.1	8.8-80.1				2.74-6.10	1.22			
3	13.3-53.4	8.8-80.1	8.8-80.1				2.74-4.98	5.49-11.6			
4	13.3-53.4	8.8-80.1	8.8-80.1	8.8-80.1			2.74-5.49	1.22	1.22		
5	13.3-62.3	8.8-71.2	8.8-71.2	8.8-80.1			2.74-6.10	1.22	6.10-11.6		
6	13.3-53.4	8.8-80.1	8.8-80.1	8.8-80.1			2.74-5.49	6.10-11.6	1.22		
7	13.3-53.4	8.8-71.2	8.8-71.2	8.8-80.1	8.8-80.1		2.74-6.10	1.22	6.10-11.6	1.22	
8	13.3-53.4	8.8-80.1	8.8-80.1	8.8-80.1	8.8-80.1		2.74-5.49	5.49	3.05	5.49	
9	13.3-53.4	8.8-71.2	8.8-71.2	8.8-80.1	8.8-80.1	8.8-80.1	2.74-6.10	1.22	6.10-11.6	1.22	1.22

A second type of ANN used for the truck type classification was an extension of the simple Hamming network EHAM, a detailed description of which is provided by Flood & Kartam (1998). Again, this system used a supervised training algorithm and the FHWA classification system for trucks. Input to the EHAM networks was a matrix of binary values representing a projection of the strain readings measured at a fixed location on a girder of the bridge during the passage of a truck. The binary map was a 32 by 32 matrix, with one dimension representing sample strains taken at different points in time during a truck crossing event, and the other dimension indicating the magnitude of the strain readings (see, for example, FIG. 4b). Each output from the EHAM network represented a different truck loading class. The class to which a given truck loading situation belonged was indicated by the output neuron that generated the binary value 1 (all other outputs were trained to output a binary value of 0).

A detailed description of the training and relative performances of these classifiers is provided in Flood (2000). In summary, however, they each had an average success rate at truck classification in the 80-90% range, except for the truck type 2S-1-2 which was in the high 70% range. The poor performance of the classifiers for the truck type 2S-1-2 could be attributed to the significantly different axle configuration of this type of vehicle compared to the others in the FHWA classification system.

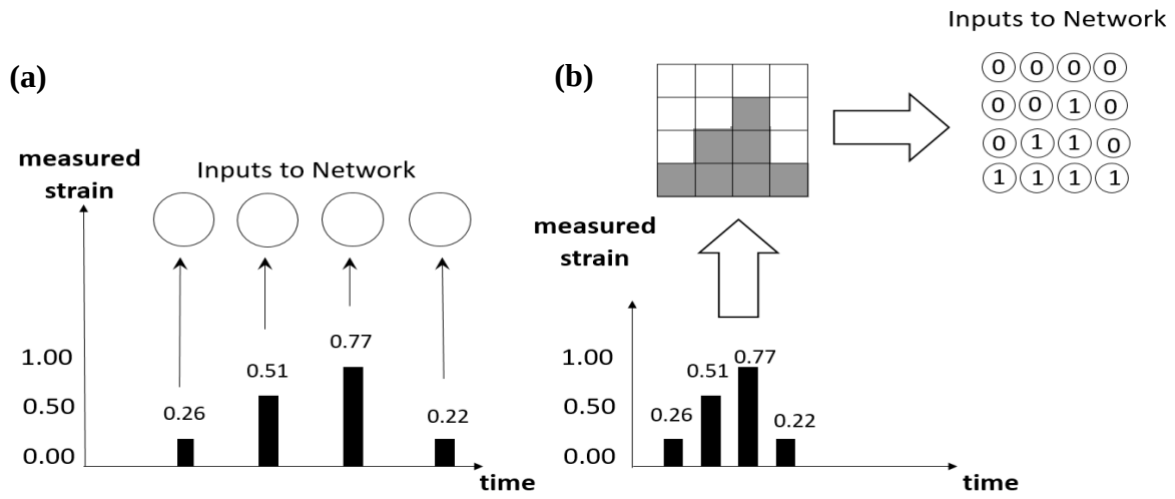


FIG. 4: Formatting Strain-Time Curves for Input to a Neural Network: (a) vector of real-values; (b) matrix of binary values (adapted from Flood (2000))

Since the 2000s, there has been a surge in interest in the application of machine learning to WIM and its related problems, corresponding to significant increases in the performance of artificial intelligence techniques. For example, machine learning methods have been used to predict potential vehicle overload from telematics such as speed, engine speed, and road gradient (Kirushnath & Kabaso, 2018). Other research extends bridge WIM as a sub-problem to help identify bridge damage using machine learning (Gonzalez & Karoumi, 2015), although the technique was only used in railway networks where vehicle axle spacing is known. Deep ANNs (comprising multiple hidden layers) have been applied to the WIM problem, in particular using a convolutional neural network (CNN) architecture. In the case of Ma et al. (2020), the CNN was trained using strain data gathered from multiple sensors, which gave it more scope for accuracy but will increases it cost of implementation and maintenance. Kawakatsu et al., (2019) used a CNN approach with a single sensor vehicle weighing system to predict lane speed, vehicle position and number of axles, but axle loads were not predicted. Deep ANN technologies are a burgeoning and rapidly developing field, with a lot of potential for future application to WIM.

Innovative WIM technology is not limited to modelling and prediction, but includes novel telemetric methods for measuring bridge response and traffic loading data. For example, instead of using structural deformation measurements (strain), bridge piezo-bearing sensors have been developed that measure induced load directly (Choo et al., 2018). This technique uses the bridge bearing as a weighing scale to measure the reaction forces at the supports resulting from passing traffic, and has the advantage of being relatively inexpensive to install. Fiber optic system have been developed for detect axles and measure their configuration for a vehicle, instead of using bridge deformation and reaction loads. The disadvantage of using fiber optics is its cost and installation effort compared to traditional strain-based methods (Lydon et al., 2015). These alternative methods of data collection offer potential for the development of future approaches to WIM modelling and prediction.

3 SUPPORT VECTOR MACHINE VERSUS ARTIFICIAL NEURAL NETWORK APPROACHES

A series of studies was undertaken by the authors to determine whether support vector machines (SVMs) could improve on the truck type classification performance of the ANN approach. New training and testing data were established for this purpose to facilititate a direct comparison between the techniques. The bridge type considered was 100 meters in length, single span, simply supported, with a single lane. The bridge was treated as a rigid beam and the study assumed no dynamic effects on the structure. Single truck crossing events only were considered.

3.1 Model Structure

A truck crossing event was represented as an array of bending moments induced at mid-span, while the type of truck inducing the bending moments was indicated across an array of outputs. The FHWA system of truck classification was adopted as used for the studies described above.

Three different model formats were considered as illustrated in FIG. 5 and described in detail by Wang (2015). The first model format comprised a monolithic model that mapped directly from the input array of bending moments to a set of 9 outputs representing the different truck types. Each output represented a different truck type and was capable of generating a value between 0.0 and 1.0. The output that generated the closest value to 1.0 in response to a set of bending moments was assumed to identify the truck type. This format was only adopted for the ANN model since SVMs cannot include more than one output.

The second model format shown in FIG. 5 comprised a set of 9 sub-models, each dedicated to a single truck type. A single array of bending moments was shared as input, and each sub-model had a single output capable of generating a value between 0.0 and 1.0. As for the first model format, the output that generated the closest value to 1.0 was assumed to identify the truck type. This format was adopted for both the ANN and SVM models.

The third model format shown in FIG. 5 comprised 36 sub-models, each dedicated to selecting between a pair of truck types (there being 36 permutations of truck pairs in total). A single array of bending moments was shared as input. Each output would select between a pair of trucks. For example, sub-model 1 was dedicated to comparing truck types 1 and 2; an output of 0 would indicate truck type 1 and an output of 1 would indicate truck type 2. Each output of 1 was regarded as a vote for that truck type. The truck type with the most votes across the output array was assumed to be the truck type crossing the bridge.

3.2 Truck Crossing Simulation

The data used for training and validation of the models was based on a random selection of truck configurations within the ranges provided in Table I. Data was generated by simulating the passage of a truck crossing the bridge. The bending moment induced at the mid-span of the bridge, m, was calculated during the truck crossing event using a 50 Hz sample rate. Each simulated truck crossing event was used to generate a single input to output pattern to be used for training or validation of the models. For model formats 2 and 3 (FIG. 5), each sub-model was trained independently. Each pattern comprised 626 inputs representing the bending moments induced by the truck crossing event, and an array of binary outputs used to indicate truck type. A total of 900 input patterns were generated (100 for each truck type) and the corresponding outputs were tailored to match the operation of each model/sub-model. For each pattern, the axle loads and spacings were selected using a uniformly distributed random variate with values ranged between the limits listed in Table 1.

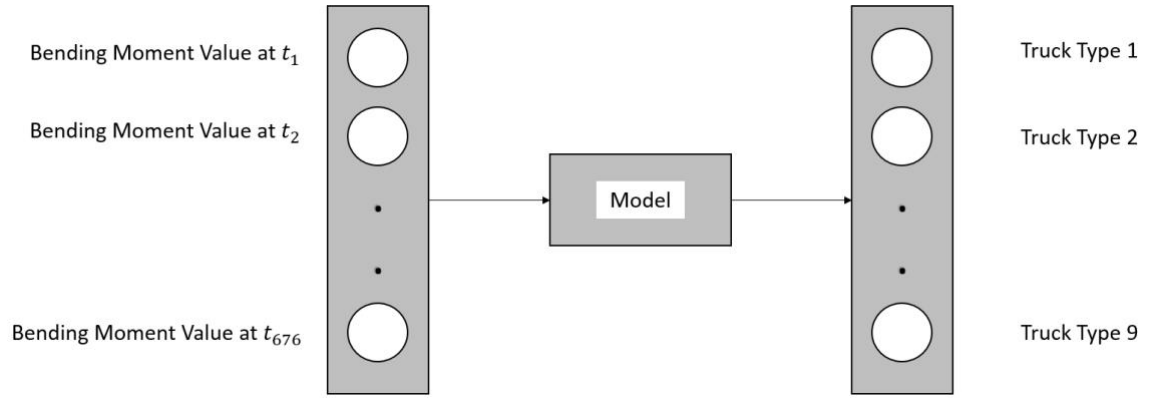
3.3 Model Development and Parameter tuning

Training of both the ANN and SVM models requires preselection of certain model parameters, the values of which can significantly affect the performance of the model. In addition, since the initial input arrays had a high dimension (626 values) principal component analysis (PCA) was used to prune this number down to something more manageable.

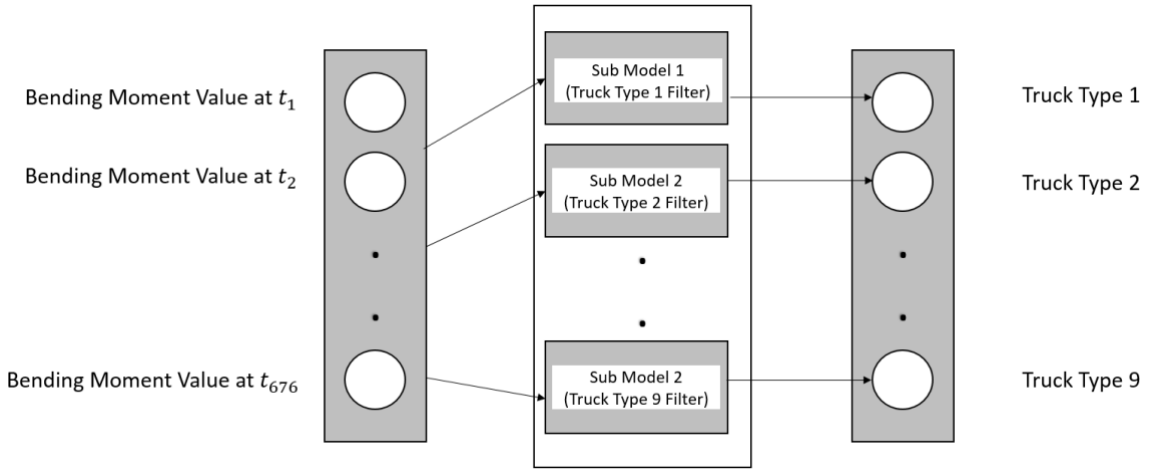
The architecture of the ANNs adopted for this study was the popular feedforward layout. Two ANN variants were considered, one with a single hidden layer of 600 neurons and a second with two hidden layers of 300 hidden neurons each. This provided a total of four ANNs, two using format 1 (FIG. 5) and two using format 2. All ANNs used the sigmoidal activation function. The training algorithm used was error backpropagation, a gradient descent technique, and was implemented within the MATLAB R2015a environment (Mathworks Inc., 2016).

Since the error backpropagation requires careful selection of the step size to ensure convergence and acceptable training quality, this study tested a range of learning rates from 0.01 to 0.1 in intervals of 0.01. For the SVM, the kernel function adopted was the Radial Basis Function due to its popularity. The kernel function also requires careful selection of its scaling value and so a range of values were tested from 3.60 to 3.70 in intervals of 0.01. For pruning the number of input variables, a range in the array size was considered from 10 to 55 in steps of 5, using principal component analysis (PCA) to select the most significant inputs in each case.

MODEL FORMAT 1: MONOLITHIC



MODEL FORMAT 2: ONE VS. ALL



MODEL FORMAT 3: ONE VS. ONE

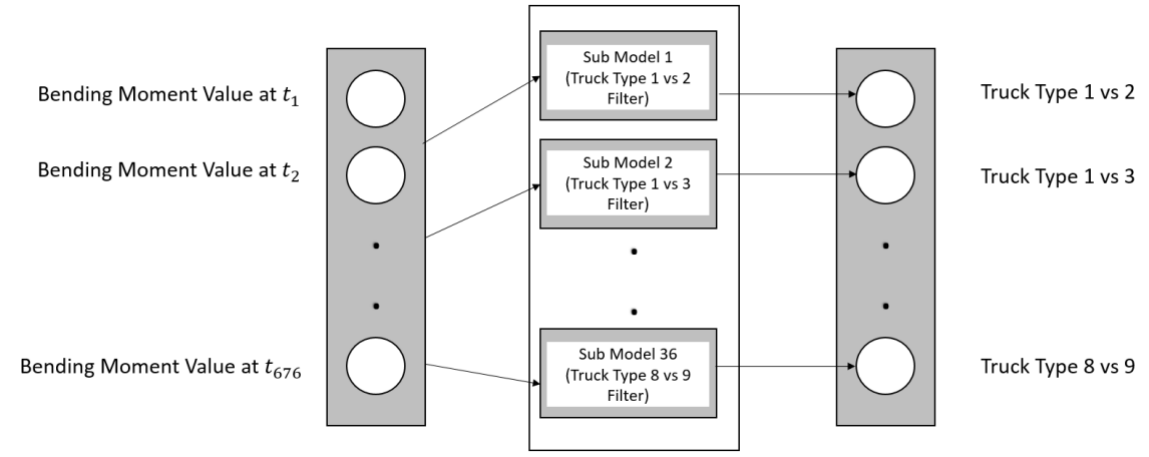


FIG. 5: Three model formats adopted for the study

3.3.1 ANN Development, Model Format 1

Training of each ANN was allowed to progress until 300 epochs had been completed, with the algorithm converging around 240 epochs. Training used a random selection of 80% of the 900 pattern data set. The remaining 20% of the patterns were used for validating the resultant ANN. Model development was repeated for the range of learning rates and input vector sizes outlined in the previous subsection (Model Development) providing 100 training trials. These experiments were repeated 10 times, each occasion using a different set of 900 patterns.

The performance of the ANNs was measured as the portion of the validation patterns correctly classified. This was averaged over the 10 repetitions of the experiment. Local scatterplot smoothing (LOESS) was used (with a span value of 0.15) to find the peak performance and thus the optimal values for the number of input variables and the learning rate. The backpropagation training algorithm was set to include adaptive learning rate changes, and all parameters (except for the learning rate which was ranged) were set to their default values for the implementation. FIG. 6 shows the results of these experiments for the ANN with one hidden layer of 600 nodes, plotting the proportion of correct classifications (from 0.0 to 1.0) against the number of inputs and the learning rate. The optimal values were found to be 33 for the number of inputs and 0.0605 for the learning rate.

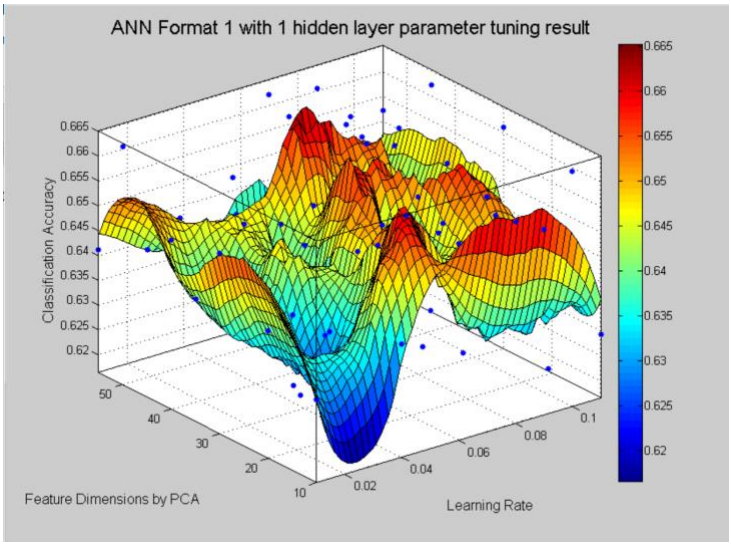


FIG. 6: Performance of One-Hidden Layer ANN using Format 1 (FIG. 5) with LOESS Smoothing

The experiment was repeated this time using the ANN with two hidden layers of 300 nodes each, the results for which are shown in FIG. 7. The optimum set-up was found to be 35 inputs with a learning rate of 0.0656.

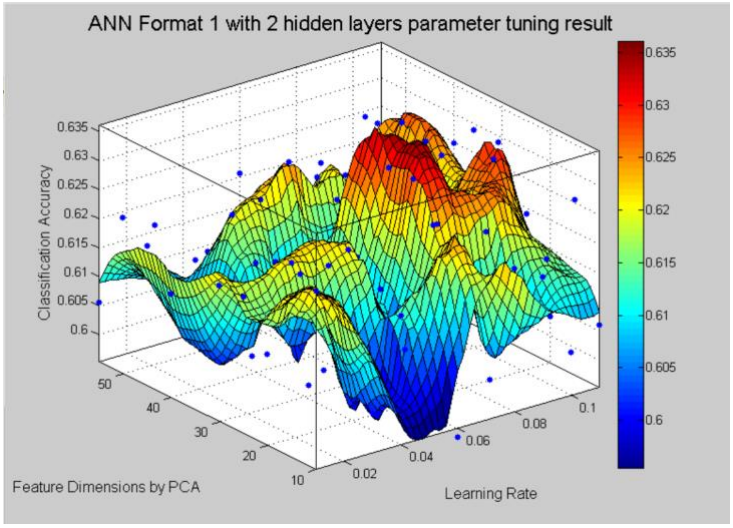


FIG. 7. Performance of Two-Hidden Layer ANN using Format 1 (FIG. 5) with LOESS Smoothing

The results of these two sets of experiments for ANN model Format 1 indicated that performance was not particularly sensitive to changing from one to two hidden layers. The peak performances on the surfaces were 0.6623 and 0.6309 for the proportion of correct classifications for one hidden layer and two hidden layers respectively. Thus, the best performing model Format 1 ANNs both misclassified about a third of the validation cases.

Both the learning rate and the PCA dimension affected the performance significantly, although the behavior of the LOESS smoothed surfaces (FIGs. 6 and 7) were quite erratic with no obvious trend. Thus, it seems unlikely that if a new independent set of trials were performed that the same basic surface form and peak locations would be achieved.

3.3.2 ANN Development, Model Format 2

The previous set of experiments were repeated but this time using the model Format 2 shown in FIG. 5, that is, the system comprising 9 sub-models with a one versus all output selection strategy. For the one hidden layer ANN, the number of hidden neurons in each sub-model was 67 giving 603 hidden neurons in total (close to the 600 hidden neuron total used for model Format 1). For the two hidden layer ANN, 33 hidden neurons were included in each layer of each sub-model providing a total of 594 hidden neurons. The training algorithm, training parameters and number of training epochs were the same as those used in the development of the Format 1 ANN models.

FIGs. 8 and 9 show the results of these experiments for the one-hidden layer and two-hidden layer ANNs respectively, as before plotting the proportion of correct classifications against the number of PCA selected inputs and the learning rate.

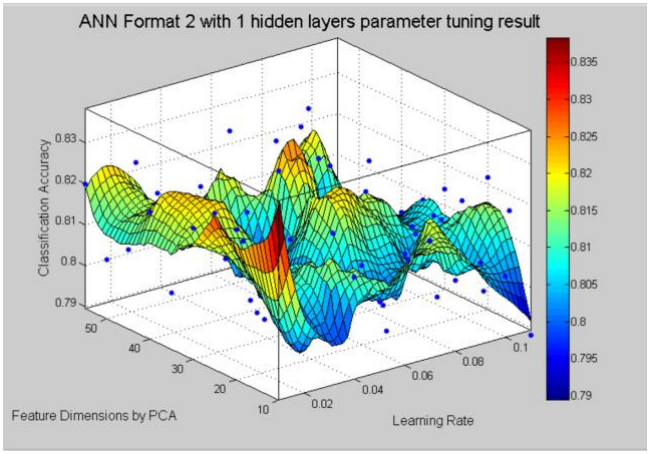


FIG. 8. Performance of One-Hidden Layer ANN using Format 2 (FIG. 5) with LOESS Smoothing

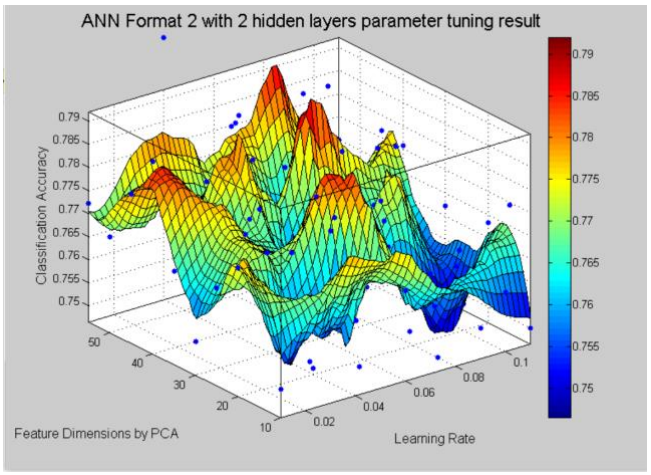


FIG. 9. Performance of Two-Hidden Layer ANN using Format 2 (FIG. 5) with LOESS Smoothing

The optimal values for the one-hidden layer ANN based on model Format 2 were found to be 10 for the number of PCA selected inputs and 0.01 for the learning rate. Similarly, for the two hidden layer ANN based on model Format 2, the optimal values were found to be 40 for the number of inputs and 0.0767 for the learning rate.

As with the Format 1 ANN experiments, the Format 2 ANNs were not particularly sensitive to changing from one to two hidden layers. The peak performances were 0.8384 and 0.7905 for the proportion of correct classifications of the validation cases for the one hidden layer and two hidden layer ANNs respectively, significantly better than the Format 1 ANNs at 0.6623 and 0.6309 respectively.

As for the model Format 1 ANNs, both the learning rate and the PCA dimension affected the performance significantly, although the behavior of the LOESS smoothed surfaces (FIGs. 8 and 9) were again quite erratic with no obvious trend.

3.3.3 SVM Development, Model Formats 2 and 3

The next two sets of experiments concerned development of the SVM models, the first using the one versus all strategy (model Format 2) and the second using the one versus one strategy (model Format 3). FIG. 10 shows the LOESS smoothed performance surface for the SVMs based on model Format 2. The surface plots the proportion of correctly classified trucks in the validation data set versus the number of PCA selected inputs and the Radial Basis Function kernel scaling value. The optimal values were found to be 20 for the number of inputs and 3.6312 for the kernel scaling value.

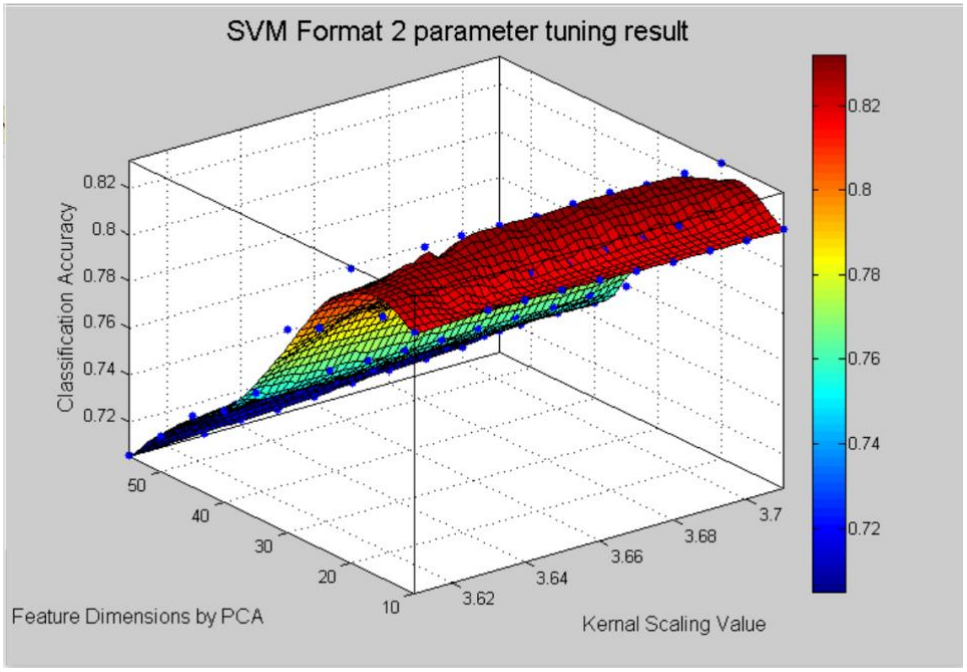


FIG. 10. Performance of SVM using Format 2 (FIG. 5) with LOESS Smoothing

Similarly, the SVM system based on model Format 3 is plotted in FIG. 11. The optimal values were 17 for the number of PCA selected inputs and 3.6262 for the kernel scaling value. It can be seen from FIGs. 10 and 11 that after narrowing down the scaling value to between 3.6 and 3.7 it did not play an important role in determining the performance of the resultant SVM. However, the number of PCA selected inputs was clearly very important for both SVM modeling approaches. Moreover, the surfaces are well-behaved indicating a clear trend in performance with respect to the PCA value.

The peak performances were 0.8280 and 0.8578 for the proportion of correct classifications of the validation cases for the SVM Format 2 models and SVM Format 3 models respectively. The one-vs-one format (Format 3) slightly outperformed the one-vs-all format (Format 2), with around 14-15% misclassification of the validation cases.

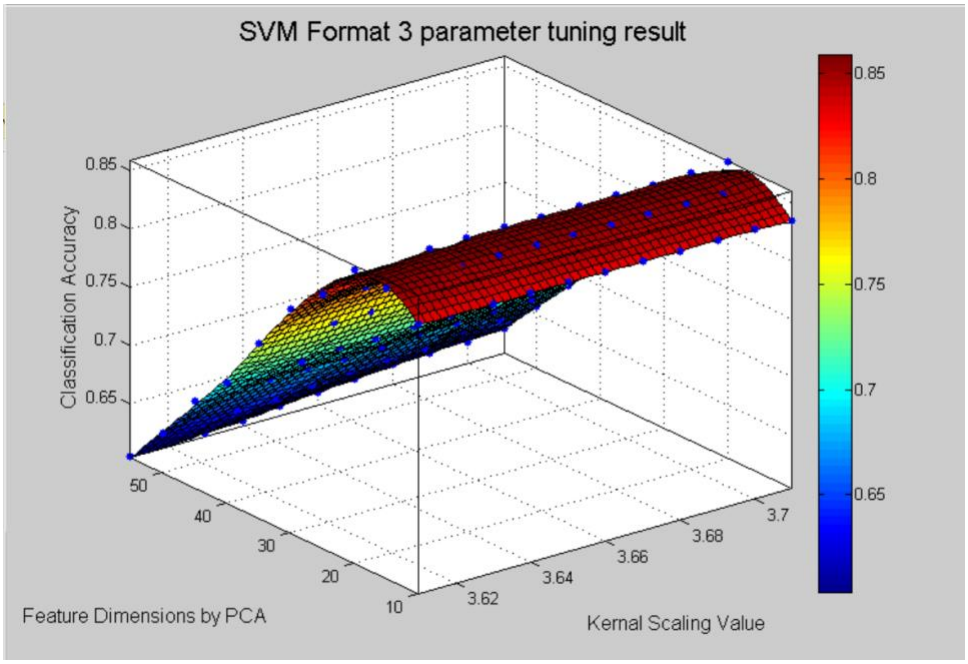


FIG. 11. Performance of SVM using Format 3 (FIG. 5) with LOESS Smoothing

4 MODEL EVALUATION

The optimal values determined for the number of inputs and the learning rate or kernel value were used to develop a set of final evaluation versions of each of the 6 model forms (the 6 forms being the 1 and 2 hidden layer ANNs for both model Formats 1 and 2, and the SVMs for model Formats 2 and 3). The performances of each optimal model configuration are compared in FIG. 12 in terms of their ability to correctly classify the validation patterns. Each bar summarizes the performances of 10 versions of the optimal configuration of a model form, showing the full range of the results (lowest and highest bars), the lower and upper quartiles (lower and upper bounds of the shaded box) and the median (thick bar). All 1,800 validation patterns generated across the 10 data sets were used for this purpose. The graph demonstrates that the SVM models outperform the ANN models, and that the SVM with the one-vs-all strategy was found to slightly outperform the one-vs-one strategy (median performance). However, this best model achieved a median performance of less than 80% correct classifications, leaving a need for further improvement.

For the ANN models, the structure comprising 9 sub-models significantly outperformed the monolithic ANN structure. Having individual sub-models may provide the system more flexibility in learning the pattern of a specific truck type and therefore improve the accuracy of the combined model.

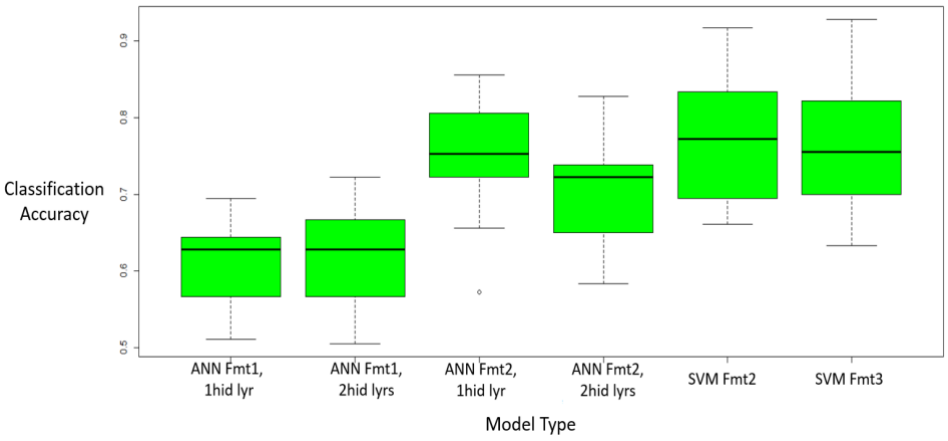


FIG. 12. Comparison of the Performance of the Optimal Configurations of Each Model Form

FIG. 13 provides an analysis of the misclassified truck cases for an example close to the median performance for the best performing model in this evaluation, the SVM using the one-vs-all format. The arrows indicate the number and direction of the misclassifications. It can be seen from this figure that the misclassifications fall into clusters between truck types 1 and 2, truck types 3, 5 and 6, and truck types 7, 8 and 9. These three clusters accounted for around 85% of all the misclassifications. The confusion appears to be between trucks that have the same number of axle groupings. For example, truck types 3, 5 and 6 comprise 3 groupings of axles. This suggests a better modeling approach may be to first classify trucks by the number of axle groupings, and then to use second level models to refine the classification to the specific FHWA type.

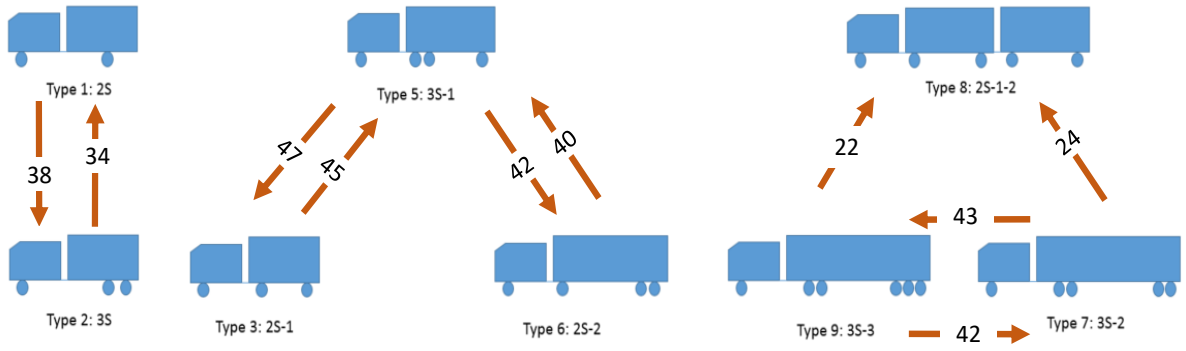


FIG. 13: Three Clusters Containing 85% of Truck Misclassification for the SVM Classifier using the One Versus All Format (total cases: 1800)

5 CONCLUSIONS AND FUTURE WORK

The study first reviewed existing approaches for solving the truck WIM problem identifying this as a good benchmark problem for developing more effective empirical modeling techniques. The study then developed and compared the performances of six ANN and SVM based truck classifiers using synthetically generated truck weigh-in-motion data. Three basic model formats were considered: (i) a monolithic model structure with a one-vs-all truck type classification strategy; (ii) an array of sub-models each dedicated to one truck type with a one-vs-all classification strategy; and (iii) an array of sub-models each dedicated to selecting between pairs of trucks in a one-versus-one classification strategy. The optimal versions of each model were determined using a LOESS based model development parameter selection schema. Overall, the formats that used an array of sub-models outperformed the monolithic models, with the support vector machines having a slight edge over the artificial neural networks.

Future work should be concerned with developing models that are extendable to a wider range of problems, including bridges of different lengths, span configurations, and numbers of lanes, as well as situations involving multiple truck crossing events and noise. Such models should also be able to estimate truck parameters such as axle loadings and spacing, the level 2 components of FIG. 2. A challenge is to achieve this while circumventing the problem of a geometric increase in the number of training patterns with respect to the number of variables required to describe the problem. Work should also consider a range of novel ANN architectures applicable to WIM, including deep systems such as ResNets (He et al., 2016), RNN (recurrent neural networks) and CNNs (Dupond, 2019). Optimization should include parametric studies ranging the number of hidden layers and other parameters in these devices.

ACKNOWLEDGEMENT

This work was funded by the University of Florida graduate student fellowships program.

REFERENCES

Bellman, R. E., (Republished, 2003). Dynamic Programming. Dover Publications, NY.

Choo, J. F., Ha, D. H., Chang, S. G., Lee, D. H., & Cho, C. B. (2018). New bridge weigh-in-motion system using Piezo-bearing. Shock and Vibration.

Cortes, C. and Vapnik, V., (1995). Support-Vector Networks, *Machine Learning*, Vol. 20, 273–297.

Dupond, S., (2019). "A Thorough Review on the Current Advance of Neural Network Structures". Annual Reviews in Control, 14, p. 200-230.

Farmaga, I., Shmigelskyi, P., Spiewak, P. and Ciupinski, L., (2011). Evaluation of computational complexity of finite element analysis, *11th International Conference on Experience of Designing and Application of CAD Systems in Microelectronics*. 213–214.

Fitzgerald, P. C., Sevillano, E., OBrien, E. J., & Malekjafarian, A. (2017). Bridge weigh-in-motion using a moving force identification algorithm. Procedia engineering, 199, 2955-2960.

Flood, I., (2000). Developments in Estimating Truck Attributes from Bridge Strain Data using Neural Networks, *6th Congress on Computing in Civil Engineering*, ASCE, Stanford, CA, 8 pp.

Flood, I. and Issa, R., (2010). Empirical modeling methodologies for construction, *Journal of Construction Engineering and Management*, Vol.136, 36–48.

Flood,I. and Kartam, N., (1998). A binary classifier with applications to poorly defined engineering problems, *Journal of Artificial Intelligence in Engineering, Design, Analysis and Manufacturing*, 259–272.

He, K., Zhang, X., Ren, S. and Sun, J., (2016). "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 770-778.

Gagarin, N., Flood, I. and Albrecht, P., (1994). Computing truck attributes with artificial neural networks, Journal of Computing in Civil Engineering, Vol. 8, 179–200.

Gonzalez, I., & Karoumi, R. (2015). BWIM aided damage detection in bridges using machine learning. J Civ Struct Heal Monit 5 (5): 715–725.

Kawakatsu, T., Aihara, K., Takasu, A., & Adachi, J. (2018). Deep sensing approach to single-sensor vehicle weighing system on bridges. IEEE Sensors Journal, 19(1), 243-256.

Kirushnath, S., & Kabaso, B. (2018, July). Weigh-in-motion using machine learning and telematics. In 2018 2nd International Conference on Telematics and Future Generation Networks (TAFGEN) (pp. 115-120). IEEE.

LeCun, Y., Cortes, C. and Burges, CJC, (accessed Nov, 2017). The MNIST Database of Handwritten Digits, http://yann.lecun.com/exdb/mnist/ .

Lydon, M., Taylor, S. E., Robinson, D., Mufti, A., & Brien, E. J. O. (2016). Recent developments in bridge weigh in motion (B-WIM). Journal of Civil Structural Health Monitoring, 6(1), 69-81.

Ma, R., Zhang, Z., Dong, Y., and Pan, Y., (2020). Deep Learning Based Vehicle Detection and Classification Methodology Using Strain Sensors under Bridge Deck, Sensors 2020, 20(18), 25 pp.

Mathworks Inc., (2016). MATLAB - MathWorks, www.mathworks.com/products/matlab.

McCall, J., (2005). Genetic algorithms for modelling and optimisation, Journal of Computational and Applied Mathematics, Vol. 184, 205–222.

Vala, G., Flood, I. and Obonyo, E., (2011). Truck Weigh-in-Motion using Reverse Modeling and Genetic Algorithms, International Workshop on Computing in Civil Engineering, 760–767.

Wang, Y., (2015). Structured Versus Direct-Mapping Approaches to Empirical Modelling of Civil Engineering Problems, PhD Dissertation, University of Florida.