Comparison of Artificial Neural Networks and Support Vector Machines for Weigh-In-Motion Based Truck Type Classification

Yueren Wang
Rinker School, College of Design, Construction and Planning,
University of Florida
Gainesville, United States
ywang016@ufl.edu

Ian Flood
Rinker School, College of Design, Construction and Planning,
University of Florida
Gainesville, United States
flood@ufl.edu

Raja R.A. Issa,
Rinker School, College of Design, Construction and Planning,
University of Florida,
Gainesville, United States
raymond-issa@ufl.edu

***Abstract* – the paper develops and compares a comprehensive range of configurations of artificial neural networks and support vector machines for solving the truck classification by weigh-in-motion problem. A local scatter point smoothing schema is also demonstrated as a means of selecting an optimal set of design parameters for each model type. Three main model formats are considered: (i) a monolithic structure with a one versus all strategy for selecting truck type; (ii) an array of sub-models each dedicated to one truck type with a one versus all truck type selection strategy; and (iii) an array of sub-models each dedicated to selecting between pairs of trucks. Overall, the SVM approach was found to outperform the ANN based models. The paper concludes with some suggestions for extending the work to a broader scope of problems.**

Keywords – artificial neural network; empirical modeling; support vector machine; truck weigh-in-motion.

I. INTRODUCTION

Empirical modeling is concerned with the development of a representation of some aspect of a system based on data observed from that system or from an analog of that system. While empirical modeling is widely used in fields such as business, engineering, and science, its poses many challenges that need to be overcome before its full potential can be realized [1]. The ability to classify moving trucks based on the strain envelopes they induce on bridge girders (termed weigh-in-motion) has been identified as an example problem rich in the issues challenging empirical modeling, and thus provides a good point of reference in developing and evaluating this modeling technique [1].

To date a variety of empirical modeling techniques have been applied to the problem of truck-type classification from bridge weigh-in-motion data. Supervised learning methods such as artificial neural networks (ANNs) have been studied extensively in this regard. In a comprehensive study by Gagarin et al. [2] an ANN was used to map directly from a stream of strain readings measured on a bridge girder resulting from a truck crossing event to an output array where each element represented a truck type; the output element with the strongest response represented the ANN's determination of the truck-type. This approach demonstrated reasonable accuracy in classifying trucks, around 857% correct classifications, but it did not perform well for truck types with similar wheel configurations. A later study [3] attempted to improve performance by using a type of Hamming Network (a binary classifier) with novel presentation formats. While both types of ANN showed some promise as classifiers, they suffered from a more fundamental problem associated with direct-mapping solutions, namely that each classifier can only work for a single bridge configuration. Each new bridge requires collection of a new set of training patterns followed by training of a new classifier. Moreover, it is not possible to extend the scope of application of these classifiers by including additional input variables to describe a bridge's configuration since this would lead to a geometric explosion in the number of training patterns required.

In an attempt to get around this problem, a radically different approach was considered by Vala et al. [4] based on genetic algorithms (GAs). In this study the GA was used to evolve a truck-type configuration that could best explain the strain envelope, and a numeric structural model of the bridge was used to evaluate the evolving solution. While the GA approach was, in principle, more flexible than the ANN classifier in terms of its scope of application, the study was only preliminary and its ability to estimate truck-type satisfactorily was not conclusive.

Work is on-going at the University of Florida developing more flexible methods of classifying truck-types from weigh-in-motion data as part of a broader line of study

 ISBN: 978-1-61208-416-9

developing empirical modeling methods. However, to conclude the work on direct-mapping it was decided to compare the use of ANNs to that of support vector machines (SVMs), an empirically based classification device that has developed a growing interest since the mid-1990's with success in a diverse range of applications including facial detection [5], CT image recognition [6], and market power estimation [7].

This paper is specifically concerned with comparing the performance of SVM and ANN based approaches to weigh-in-motion based truck-type classification.

II. Modeling Approach

A. Truck Types and Bridge Properties

A total of nine different truck types were considered in this study, representing those most frequently operating on US highways, as adopted in earlier research [8] and outlined in Figure 1. For each truck type there is a range of values defining its axle loadings and axle spacings as summarized in Table I.

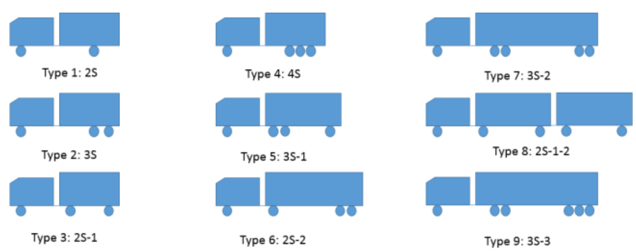

Figure 1. Nine truck types used in this paper (adapted from Gagarine et al. [1])

TABLE I: Axle Load and Spacing Range of Nine Truck Types (adapted from Gagarine et al. [1])

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	8.8-71.2	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
9	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	

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.

B. 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.

Three different model formats were considered as illustrated in Figure 2. 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.

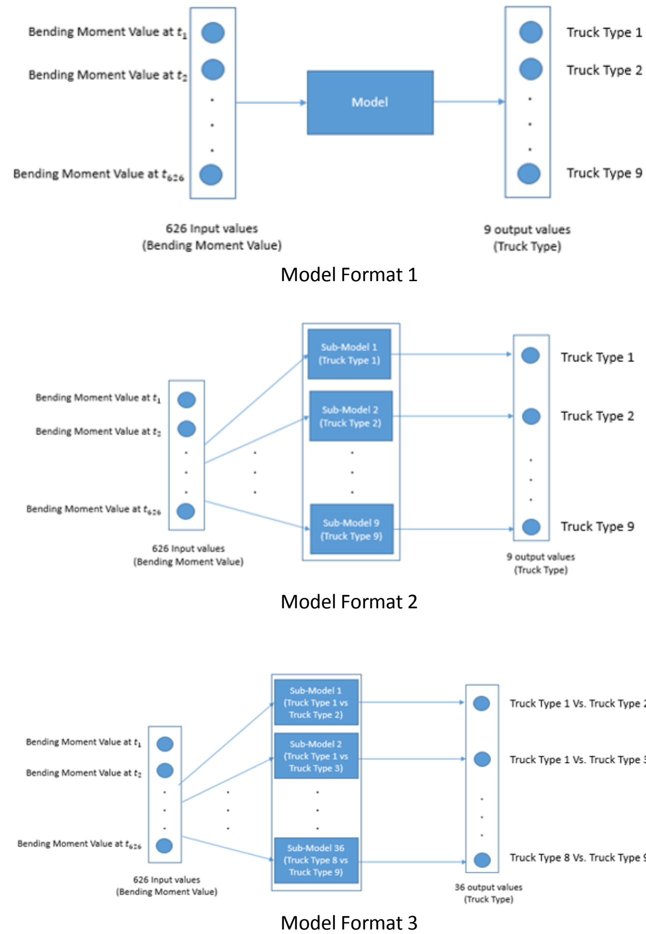

Figure 2. Three model formats adopted for the study

The second model format shown in Figure 2 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 Figure 2 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

 ISBN: 978-1-61208-416-9

types 1 and 2; an output of 0 would indicate truck type 1 and an output of 1 would indicate truck type 2. The truck type with the most selections across the output array was assumed to be the truck type crossing the bridge.

C. Truck Crossing Simulation

The data used for training and validation of the models was based on a random selection of truck configurations (based on Table I). Data was generated by simulating the passage of a truck over 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 as indicated in Figure 3.

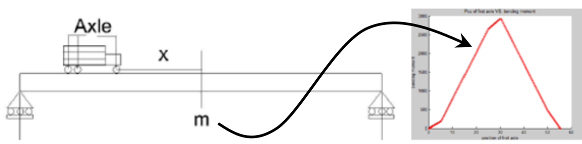

Figure 3. The simulation of a type 3S truck passing a bridge and its bending moment envelope

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/sub-models. For model formats 2 and 3, 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 I.

III. PRELIMINARY MODEL DEVELOPMENT

Training of both the ANN and SVM models requires preselection of certain training parameters, the values of which can significantly affect the performance of the model. In addition, since the initial input arrays have 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 (Figure 2) and two using format 2. All ANNs used the sigmoidal activation function. Future work may use a sensitivity analysis to assess the dependence of model performance on the number of hidden neurons. The training algorithm used for the ANN was error backpropagation, a gradient descent technique, and was implemented within the MATLAB R2012a environment.

Since the ANN backpropagation approach requires a 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 SVM's kernel function also requires a careful selection of its scaling value, and so a range of values were tested from 3.60 to 3.70 with 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.

A. ANN Development, Model Format 1

Training of an ANN was allowed to progress until 300 training epochs had been completed. 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 above, 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 adopted (with a span value of 0.15) to find the location of the peak performance and thus the optimal values for the number of input variables and the learning rate. Figure 4 shows the results of these experiments, plotting the proportion of correct classifications (from 0.0 to 1.0, color coded) against the number of inputs (PCA derived) and the learning rate. The optimal values were found to be 32 for the number of inputs and 0.088 for the learning rate. The corresponding R square value was 0.8249, indicating an acceptable description of parameter relationship using the smoothing method.

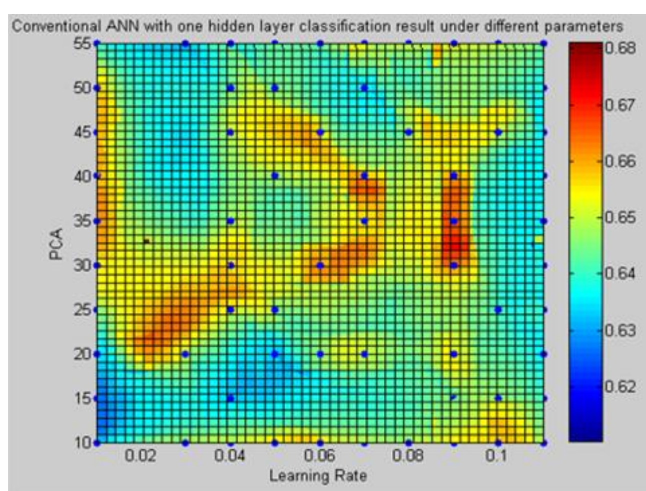

Figure 4. LOESS regression on the performance of ANN Format 1 with one hidden layer

The experiment was repeated this time using an ANN with two hidden layers, with the resultant performance surface shown in Figure 5. The optimum set-up was found to be 35 inputs and a learning rate of 0.0676. The R squared value was again acceptable at 0.8926.

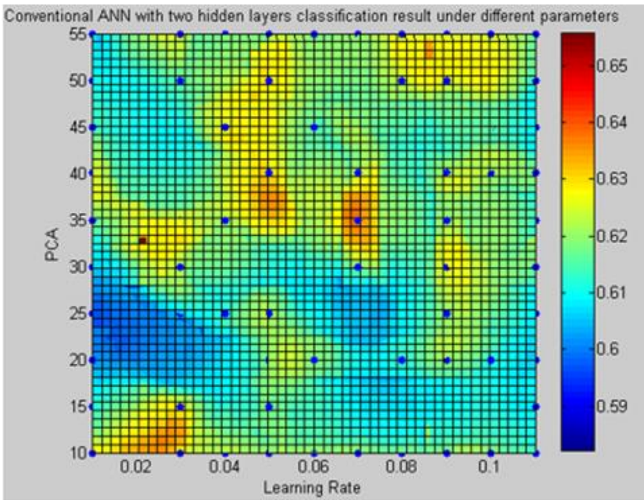

Figure 5. LOESS regression on the performance of ANN Format 1 with two hidden layers

B. ANN Development, Model Format 2

The set of experiments described in section A above were repeated but this time using the model format 2 shown in Figure 2, that is, the system comprising 9 sub-models.

For the one hidden layer ANN, the number of hidden neurons in each sub-model was 67 for the one hidden layer ANN, giving 603 hidden neurons in total. 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.

Figure 6 shows the results of these experiments, as before plotting the proportion of correct classifications against the number of PCA selected inputs and the learning rate.

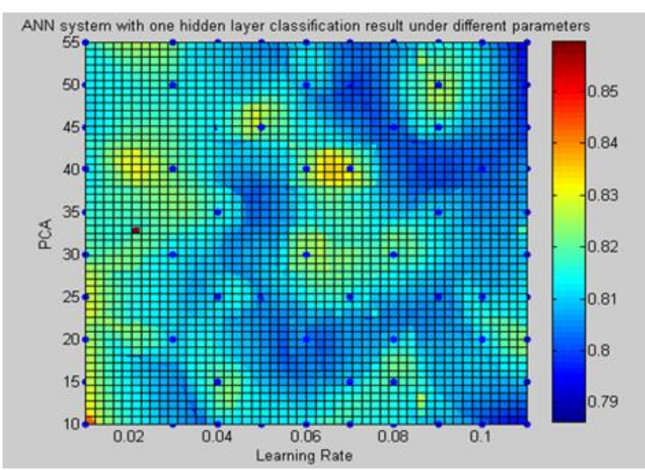

Figure 6. LOESS regression on the performance of ANN Format 2 with one hidden layer

The optimal values were found to be 39 for the number of inputs and 0.06520.088 for the learning rate. The corresponding R square value was acceptable at 0.8304.

Similarly Figure 7 plots the LOESS smoothed surface representing the proportion of correctly classified validation patterns for the two hidden layer ANN based on model format 2. In this case the optimal values were fund to be 40 for the number of PCA selected inputs and 0.0736 for the learning rate, with an R square value of 0.8304.

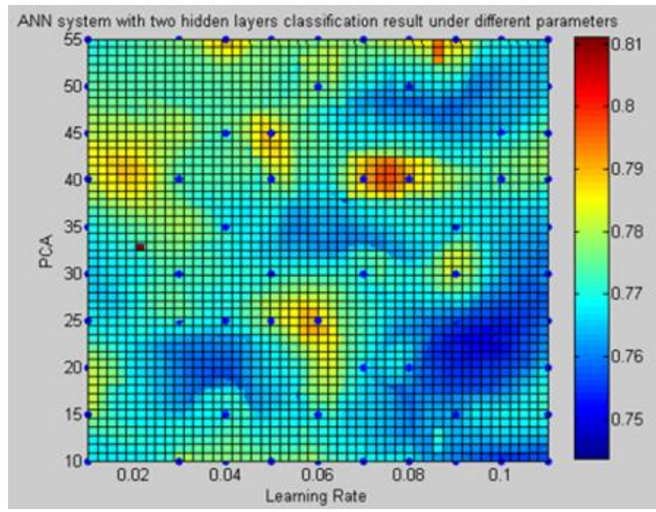

Figure 7. LOESS regression on the performance of ANN Format 2 with two hidden layers

C. SVM Development, Model Formats 2 and 3

The next set of experiments concerned development of the SVM models, the first using the one versus all strategy (model format 2, Figure 2) and the second using the one versus one strategy (model format 3, Figure 2).

Figure 8 shows the LOESS smoothed performance surface for the SVM based on model format 2. This 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 19 for the number of inputs and 3.6677 for the kernel scaling value, with an R square equal to 0.9984.

Similarly the SVM system based on model format 3 is plotted in Figure 9. The optimal values were 18 for the number of PCA selected inputs and 3.6960 for the kernel scaling value, with an R square value of 0.9999.

It can be seen from Figures 8 and 9 that the scaling value 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.

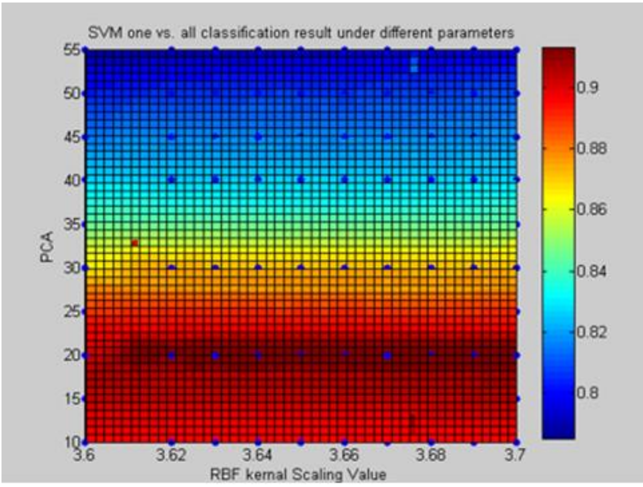

Figure 8. LOESS regression on the performance of SVM Format 2 with a one versus all strategy

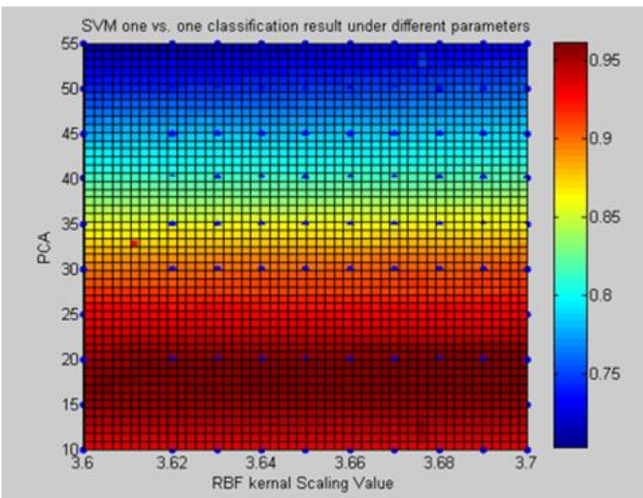

Figure 9. LOESS regression on the performance of SVM Format 2 with a one versus one strategy

IV. MODEL EVALUATION

The optimal values determined for the number of inputs and the learning rate or kernel value were used to develop the final versions of each of the 6 model forms. The performance of each of these models is compared in Figure 10 in terms of their ability to correctly classify the validation patterns. All 1,800 validation patterns generated across the 10 data sets were used for this purpose.

Clearly, the results demonstrate that the SVM models outperform the ANN models. Of the SVM models, the one versus all strategy was found to slightly outperform the one versus one strategy.

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

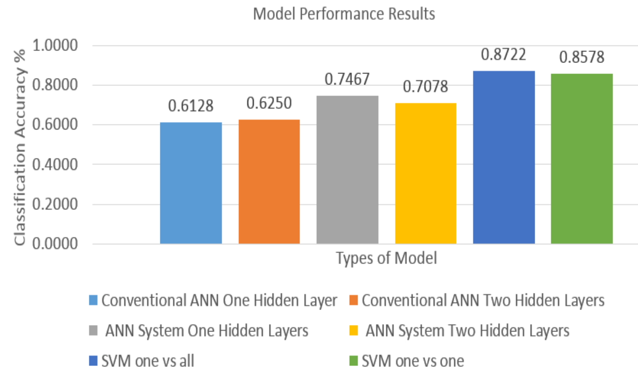

Figure 10. Comparison of optimal model performances for 1,800 validation patterns

For the ANN models, the number of hidden layers did not appear to have a significant impact on classification performance.

Figure 11 provides an analysis of the misclassified truck patterns for the single hidden layer ANN, model format 1, for the validation patterns. The blue arrows indicate the number and direction of the misclassifications. It can be seen from this figure that the misclassifications between truck type 7, 8 and 9 and truck type 1, 2 and truck type 3, 5 and 5, 6 contributed to the majority of the misclassification instances. As might be expected, it is also apparent from this that the misclassifications tended to occur between trucks with similar axle configurations.

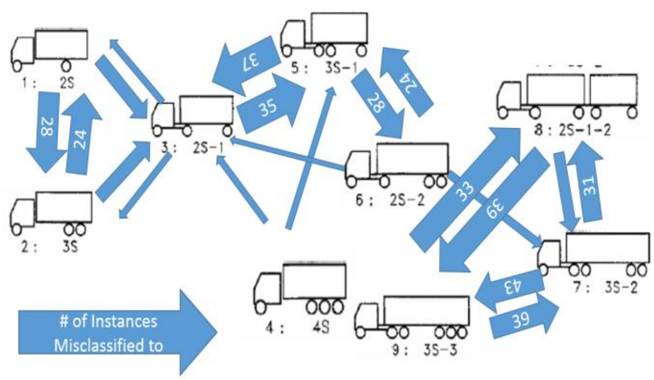

Figure 11. Analysis of truck missclassifications for the single hidden layer ANN, model format 1 (see Figure 2)

V. CONCLUSION AND FUTURE WORK

The study developed and compared the performances of 6 ANN and SVM based truck classifiers using weigh-in-motion data. The optimal versions of each model were determined using a LOESS based empirical modelling parameter selection schema. The results indicated that the SVM models significantly outperformed the ANN models in terms of the number of correct truck classifications.

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. Such models should also be able to estimate truck parameters such as axle loadings and spacing. 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.

ACKNOWLEDGMENT

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

REFERENCES

[1] I. Flood, and RRA. Issa, "Empirical Modeling Methodologies for Construction," J. Constr. Eng. Manage., vol. 136, pp. 36-48, August, 2009.

[2] N. Gagarin, I. Flood, and P. Albrecht, "Weighing trucks in motion using Gaussian-based neural networks," Proc. Int. Joint Conf. on Neural Networks, Baltimore, Md., 2, 1992, pp. 484-489.

[3] I. Flood, "Developments in Weigh-in-Motion Using Neural Nets," Proc. Comput. Civ. And building Eng., 2000, pp. 1133-1140, doi: 10.1061/40513(279)148.

[4] G. Vala, I. Flood and E. Obonyo, "Truck Weigh-in-motion using reverse modeling and genetic algorithms," Proc. Comput. Civ. Eng., 2011, pp. 219-226, doi: 10.1061/41182(416)27.

[5] E. Osuna, R. Freund and F. Girosi, "Train support vector machines: an application to face detection," Proc. IEEE Computer Vision and Patten Recognition, June, 1997, pp. 130-136, doi: 10.1109/CVPR.1997.609310.

[6] Q. Wang, W. Zhu and B. Wang, "Three-Dimensional SVM with latent variable: an application for detection of lung lesions in CT images," J. Med. Syst., 2015, pp. 39-171.

[7] SP. Karthikeyan, IJ. Raglend and KS. Kumar, "Application of SVM as classifier in estimating market power under deregulated electricity market," Proc. ICPERES, Nov 2014 pp. 1309-1317, doi: 10.1007/978-81-332-2119-7_127.

[8] N. Gargarin, I. Flood, and P. Albrecht, "Computing truck attributes with artificial neural networks," J. Comput. Civ. Eng., vol. 8, pp. 179-200, April 1994.