Modeling Asphalt Pavement Frictional Properties using Different Machine Learning Algorithms

Mohammad Ali Khasawneh
Civil Engineering, Prince Mohammad Bin Fahd University, Al Azeziya, Eastern Province, Kingdom of Saudi Arabia
mkhasawneh@pmu.edu.sa

Mohammad Ahmad Alsheyab
Civil, Construction and Environmental Engineering, Iowa State University, Ames, Iowa, USA
alsheyab@iastate.edu

Haneen Issa Al Akhrass
Civil Engineering, Jordan University of Science and Technology, Irbid, Jordan
hialakhrass18@eng.just.edu.jo

Abstract

The objective of this work is to use some machine learning algorithms and test its efficiency in developing models to predict Locked Wheel Skid Trailer (LWST) values from Dynamic Friction Tester (DFT) and Circular Texture Meter (CTM) measurements conducted on asphalt pavement surfaces. For this prediction, three models were developed using DFT measurements at different speeds starting from 20km/h (12.5 mph) up to 64 km/h (40 mph) and then same DFT measurements as combination with Mean Profile Depth (MPD) and the last model used the International Friction Index (IFI) parameters (F60 and SP). The machine learning techniques includes two supervised learning algorithms: the Multi-Layer Perceptron (MLP) type of Artificial Neural Networks (ANN) and M5P tree model. In addition to one lazy algorithm called the K Nearest Neighbor (KNN) or Instance-Based Learner (IBL). The results showed that MLP models are the best in terms of the correlation coefficient that resulted in 81% prediction power using DFT parameters. Additionally, it was shown that the result of tree models was close to ANN but with much simpler regression. However, KNN models were recommended for LWST prediction of similar data characteristics and it is expected that this algorithm will be more efficient as the training data set becomes larger.

Keywords: Friction; Texture; International Friction Index (IFI); Machine Learning Algorithms

1 Introduction

Pavement skid resistance is defined as the impeding force generated by the interaction between a tire and a pavement under a non-rotating wheel (ASTM 867-02, 2015). Since skidding occurs when the frictional demand exceeds the available friction force at the interface between a tire and pavement, it is important to measure the ability of a pavement to resist the skidding of a tire on a motor vehicle.

Lack of sufficient surface friction is one of the factors contributing to crashes on roads. The tire-pavement interaction has been a vital field of study due to the volume and severity of motor vehicle accidents on roads as in Hofko et al. (2019). Theoretically, the surface frictional force developed between a pavement and a tire is composed of two components: adhesion and hysteresis. Adhesion is only significant on dry surfaces at low vehicle speed and hysteresis has a considerable

significance at high vehicle speed on wet surfaces (Khasawneh & Alsheyab, 2020).

Multiple methods and procedures have been developed to evaluate the tire-pavement interaction and surface textures. There are two major devices used for the measurement of pavement surface friction in the laboratory, namely; British Pendulum Tester (BPT) as outlined in (ASTM E 303-93, 2002) and Dynamic Friction (DF) tester as specified in (ASTM E 1911-98, 2002). Both are highly portable, easy to handle and can be used in the field.

There are four basic types of full-scale friction-measuring devices. These are side-force, locked wheel, fixed-slip, and variable-slip devices. Unlike the portable devices, these devices can measure friction at or close to highway speeds, see (Henry, 2000).

To our knowledge, no available devices can measure pavement micro-texture directly. However, they can be measured indirectly at low-speed levels using laboratory friction devices, such as the British Pendulum Tester at 10km/h and the Dynamic Friction Tester at 20 km/h as demonstrated in (Khasawneh, 2008). However, three common methods are utilized for measuring pavement macro-texture: profilometers (ASTM E 1845-15, 2015), volumetric (ASTM E 965-96, 2001) and outflow (ASTM E 2380, 2005).

Statistical interpretation and Machine Learning (ML) techniques are vital in pavement surface evaluation nowadays. Multiple technicians collect and interpret the information for pavement friction could result in contradicting viewpoints and subjectivity, see (Marcelino et al., 2021). In addition, the cost of the equipment used for pavement surface evaluation is another important factor to be considered; this was addressed by (Hall et al., 2009). In essence, many researchers globally are utilizing these tools since they can make predictions and generalizations in real time and save effort and money for many industries. Accordingly, ML-based approaches, including Support Vector Machine (SVM), artificial neural network (ANN) and classification and regression tree (CART) are widely used in pavement surface condition and damage detection applications and this can be shown in several studies such as Majidifard et al., (2020); (Karballaeezadeh et al. 2020); and (Yang, 2017). A thorough study conducted by Majidifard and co-workers showed a full procedure to evaluate the Pavement Condition index (PCI) from images using Deep Learning (DL) and ML techniques. Karballaeezadeh and co-workers introduced another use of AI-based methods in the year 2020. In this study, the authors were able to generate models to predict PCI based on the Falling Weight Deflector (FWD) deflection results. The methods used are the single multi-layer perceptron (MLP) and radial basis function (RBF) neural networks as well as their hybrids. Both studies proved that using computer-based approaches can be significantly advantageous for pavement surface evaluation. Bashar and Torres-Machi (2021) utilized Artificial Neural Networks (ANN), Random Forest (RF) and Support Vector Machine (SVM) algorithms to predict International Roughness Index (IRI) and the study resulted in good prediction models for estimating IRI from images. In addition, researchers were able to use computer-based models in the friction aspect of pavement engineering. Yang (2017) utilized the Convolutional Neural Networks (CNN) to build a friction prediction model having the pavement surface spectrogram as an input and the friction number as an output. The model predicted the friction values with 90.63% testing accuracy. Zhan et al. (2020) utilized the Deep Residual Networks (ResNets) for the prediction of pavement friction and the resulted accuracy for the model was 91.95%.

Furthermore, Cummings (2010) presented a strong relationship between Locked Wheel Skid Trailer (LWST) and IRI values. Quiros et al. (2018) satisfactorily compared the network-level friction measurements obtained with the Sideway-force Coefficient Routine Investigation Machine (SCRIM)

with measurements made with the traditional LWST. Praticò & Vaiana (2015) investigated the correlation between Mean Profile Depth (MPD) and Mean Texture Depth (MTD) and it was shown that there is no linear relationship between the two. However, significant regression models were generated for estimation purposes. Kouchaki et al. (2018) investigated the correlation between Dynamic Friction Tester (DFT), Grip Tester, and Circular Texture Meter (CTM) and found a good correlation between DFT results at high- and low- speed levels. The study also showed a greater correlation between MPD and the friction values at high speed. However, the type of surface also plays in important role in the significance of the models. Yu et al. (2020) investigated the relationship between DFT and BPN or MPD and found that at speeds of 20 km/h, correlation with BPN was significant and as speed increases, the correlation becomes stronger with MPD.

As can be seen from literature, the computer-based models are vastly used for pavement functional and surface performance prediction, and hence, this article mainly aims at generating different ML models to predict LWST values from DFT and CTM measurements. The data was taken from actual measurements made in different counties within the state of Ohio. Different techniques of ML were employed and significant models were developed.

2 Methodology

After data collection, three different Machine Learning (ML) techniques were used to predict the LWST values from DFT and CTM measurements. Three models have been produced for each ML technique; the first model using DFT results at seven (7) different speeds and the second one using the IFI parameters (F60 and Sp) and the final one using DFT and CTM parameters.

The ML techniques are Multi-Layer Perceptron (MLP), which is a type of Artificial Neural Networks (ANN), the Decision Tree model (M5P), and a lazy learning algorithm called K-Nearest Neighbor (KNN) or Instance Based Clustering Learner (IBK). Waikato Environment for Knowledge Analysis (Weka version 3.8.5) software was used to conduct the above-mentioned models. Weka is known as powerful tool in data analysis due to its portability and comprising a huge collection of data modeling techniques, together with graphical user interfaces for easy access to these functions (Kulkarni & Kulkarni, 2016).

Multi-Layer Perceptron (MLP) is a feed-forward network used as a supervised learning algorithm without any brief assumption about the distribution and pattern in the data. It consists of at least three layers namely input layer, hidden layer, and the output layer. Each layer consists of several neurons or nodes that are fully connected to the next layer by weights (Gardner & Dorling, 1998). A weighted sum of the normalized parameters is the input to each node in the hidden layer. The hidden layer modified this sum by a certain activation function and transferred it to the output layer. In this study, sigmoid activation function was used to handle the nonlinearity in the data and due to its relatively easy computations and efficiency (Gardner & Dorling, 1998). The output layer signal is then a function of the weighted sum signals from the previous hidden layer signals that can be calculated using Eq. 1 below.

$$Y' = \sum_{i=1}^{n} (\omega_i * \mathrm{sig}(\alpha)_i) + \vartheta. \quad (1)$$

Y': The predicted normalized value.
n: Number of nodes in the hidden layer.
ω_i: Node weight.
$\mathrm{sig}(\alpha)_i$: Sigmoid value for the weighted sum of the inputs to the hidden layer.
ϑ: The output bias.

The MLP learns through the training data and assigns the weight for each connection in the model. The predicted value calculated using the previous equation and the weight is then adjusted based on the error term between the actual and predicted values and using one of the back-propagation training algorithms to find the weights with minimum error term (Hornik et al., 1989).

All decision tree models are also non-parametric supervised learning algorithms that use the training data to build a decision rule to predict the target variable in both classification and prediction issues. M5 tree model follows the classical process for building decision trees and was first developed by Quinlan (1992) then modified by Wang and Witten (1997) to handle missing values and attribute with fixed possible values. According to Shmueli et al. (2006), the tree is built by two steps: recursive partitioning and pruning. The recursive partitioning step starts by splitting the data into two branches according to the strongest variables and by trying different values (threshold) for that variable to get the one with the least sum of squared errors (SSE) from the mean. Then, the second strongest variable is used to split each branch into smaller branches, and the process is repeated until no further development achieved or one of the stopping criteria is met. Each split point becomes a node on the tree and is represented by a single or multiple linear regression (MLR) called tree model or by a numeric value and called tree regression.

The second step (pruning step) is an alternative for the stopping criteria to reduce the overfitting problem. After building a full tree where each leaf contains one or a few data points, it is pruned back by removing the nodes that do not improve the quality of the predictive model. In this step, the utility of each node on the tree is tested to generate a successively smaller tree. The best tree is the one with minimum sum of squared errors or minimum cost complexity. The resulting model will represent the non-linearity in the data by a piece-wise linear function with smoothing function to fill the gaps that formed at leaf nodes. In this study M5P tree model was used.

K Nearest Neighbor (KNN), or what is known in Weka as Instance-Based Learner (IBL), is a lazy learning algorithm. In the Lazy method, unlike other Machine Learning (ML) methods such as ANN and decision trees, no generalization or action is made on the data until a call to the system is done (Webb, 2016). It also means that this method generates a prediction for the test instance when it enters the system rather than building a model. KNN methods generate a prediction based on an average of a K number of instances of the nearest neighbor in the system data set (training data). The number of instances (K) is defined by the user or using the cross-validation option. The user chooses an integer value between 1 and the number of instances in the training data set. It is not recommended to use K as a large number to avoid giving a prediction equal to the mean of the training dataset for all the testing data points. Cross-validation defines the optimal K number between one and the number defined by the user based on mean absolute error or mean squared error.

Then, the nearest neighbor instances are chosen based on some search algorithms that calculate the distances between the test point and each value in the database. Different distance functions are available, including; Euclidean, Manhattan, Minkowsky for regression, and Hamming distance for classification. Euclidean is the simplest and most used function for regression purposes based on a mathematical calculation of the distance (length of segment) between the neighbor points using the Pythagorean Theorem. The main disadvantages of KNN are the high sensitivity to noise, outliers, and irrelevant data points. Another point is that a large training database and memory are needed to increase the accuracy of the results. In turn, this makes the process slower and less efficient. Weka 3.8.5 was used to conduct the KNN modeling due to its ability to provide different alternatives to overcome some of these problems and get the best results.

3 Results and Discussion

Before starting the modeling process, all the data were normalized to a value between [-1, 1] using the attribute normalize filter in the pre-processing analysis tools, so each attribute can have the same attention in the model. Outliers were also detected and removed to improve the prediction model without any significant change or improvement for the model or the error term. Then, data was split into two subsets; training for model building and validation for testing the performance of the models based on unseen data. Finally, three different models were generalized using each ML technique. The first model using DFT results at seven (7) different speeds and the second one using the IFI parameters (F60 and Sp) and the final one using the DFT and CTM parameters.

For MLP, one hidden layer was used as recommended and proved to be sufficient for any prediction problem by Goodfellow et al. (2016) and number of nodes were determined using the process suggested by (Shahin et al., 2002). The process based on starting with a small number of hidden layers and increasing it until there was no further improvement in the model. The model improvement was represented by the Mean Absolute Error (MAE) term and correlation coefficient. The results of optimization process are shown in Fig. 1. The process started with zero nodes in the hidden layer up to 20 nodes. Zero nodes means that the relation is expected to be linear between LWST and the predictors. The results using zero nodes in terms of R^2 was almost equal to results using MLR that presented in Khasawneh (2015) and this result was improved significantly using more nodes. It can be noticed that the optimum number of nodes were different for the three models and using DFT10 up to DFT64 as predictors for LWST provided the minimum MAE.

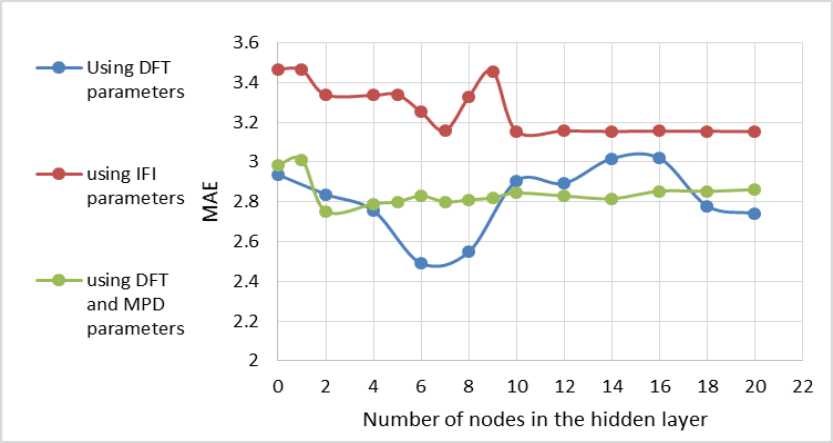


Fig. 1: Results of optimization process

For weight adjustment during the model building, small learning rate of 0.01 and a momentum of 0.1 were used, which is not uncommon in past research since the dataset and number of parameters are considered small,. Large training time (epochs) of 500,000 was assigned but with an early stopping criterion. The model building and weight adjustments stop when the error increases in the ValidationSetSize data and exceeds the validation threshold. In other words, if the error term in the training data increases for several consequence epochs (validation threshold), then the modeling stops. The weights before the last increase in error term are shown on the final model. Table 1 summarizes the final structure, training time, Root Mean Square Error (RMSE) and coefficient of determination (R^2) for the three generalized models in this study for both training and validation sets.

Table 1: Summary of the MLP results

Model	Structure	Training time (sec.)	RMSE	R^2 training	R^2 validation
DFT model	7-6-1	85.2	3.26	81.38	79.01
IFI model	2-7-1	12.9	4.16	69.60	82.64
DFT and MPD model	8-2-1	2.2	3.57	77.67	75.69

The first model is the best model, which explains 81% of the variance in the data with minimum RMSE. However, the third model gives also a close coefficient of determination with much smaller training time and simpler model with only two nodes in the hidden layer. The second model was with the least predictive power and even less than MLR models; see (Khasawneh, 2021).

Tree model was built based on 85% of data points. The remaining 15% were used as a validation set. For the first and third models, the data was represented by four (4) different linear regressions (rules) whereas one linear rule represents the data in the second model. In this case, where there is no split point in the data that can minimize the SSE from the mean, the final model of M5P is one linear rule that is very similar to the regular MLR. Table 2 summarizes the three final models.

Table 2: Summary of M5P models

Model	Linear rules	R^2 training	R^2 validation
DFT Model	$LWST_{rule1}$=0.035 DFT_{10}+0.642DFT_{64}+5.496	75.69	70.49
	$LWST_{rule2}$=0.129 DFT_0-0.322 DFT_{10}-1.269DFT_{64}+99.820		
	$LWST_{rule3}$=0.354 DFT_{10}-0.319DFT_{20}+0.435DFT_{40}+0.264DFT_{50} +0.316DFT_{64}+23.761		
	$LWST_{rule4}$=0.349 DFT_{30}+19.813		
IFI Model	$LWST_{rule1}$=0.05 Sp+0.851F_{60}+12.426	63.73	61.20
DFT and MPD Model	$LWST_{rule1}$=11.508 MPD+0.090DFT_{10}-0.247DFT_{10}+0.178DFT_{20} +0.114DFT_{50}+0.451DFT_{64}+27.8	77.19	84.64
	$LWST_{rule2}$=1.569 MPD-0.017$DFT1_0$-0.126 DFT_{10}+0.090 DFT_{20} +0.114 DFT_{50}+0.296 DFT_{64}+31.678		
	$LWST_{rule3}$=4.389 MPD+0.051DFT_0-0.441 DFT_{10}+0.566 DFT_{50} +0.462 DFT_{64}+23.860		
	$LWST_{rule4}$=1.010 MPD+ 0.349 DFT_{30}+ 0.232 DFT_{64}+20.020		

The best model for LWST prediction using M5P tree was the third model with R^2 of 77.19% and 84.64% for both training and validation sets, respectively. It was also found that DFT64 is a very important variable affecting the LWST and was included in most LWST rules. However, the second model still provided the least explanation and prediction of LWST data. Figure 2 shows the split point and ranges for each linear rule for the DFT and MPD Model.

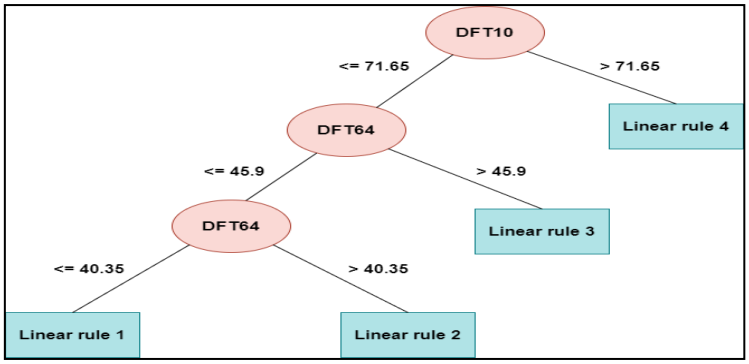


Fig. 2: Best M5P model tree

For the K Nearest Neighbor (KNN), the maximum number of nearest neighbor was set to 10 to avoid prediction values equal or near to the mean and cross-validation option was used to choose the best number of instances to be averaged. In addition, weighted average by one over the distance was used to give the closest case or point the highest weight during the prediction process. Using 85% of data set to train the KNN, the result of KNN modeling process was prediction values for each data point that were represented graphically in Figure 3 (a-b) for both training and validation datasets.

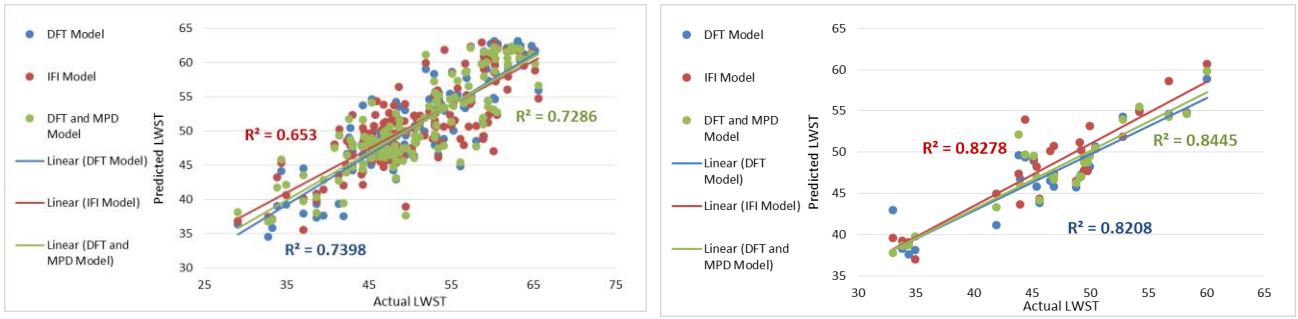
4 Conclusions

Using machine-learning techniques has a significant effect on the power of LWST prediction. Yet increasing the data points in the training set will continuously enhance the models. The best model in LWST prediction was the MLP model. The first model that used DFT parameters give the highest explanation for the variance in the data followed by DFT and MPD model. Including the MPD parameter in MLP model along with DFT parameters unexpectedly reduce the prediction power in term of R^2 from 81% to 78% but with less than 3% of training time of the first model.

However, if we are looking for simplicity, the M5P model could give good results, especially by using both DFT and MPD parameters that resulted with R^2 of 77%. Whereas, KNN model that gives the least coefficient of determination can be repeatedly improved by increasing the training data and would give the best results in case the tested points have similar characteristics to the data in training set.

For validation purposes, the results were very close using all models which can be explained by the fact that the validation dataset was chosen as a representative sample that follows almost the same distribution of the LWST data in the training set; so for each point there would be several nearest values in each model.

Finally, it can be concluded that MLP was the best technique in this study with maximum R^2 of 81% using DFT parameters to predict LWST values and that IFI parameters were not good enough to get a satisfied predictions.



(a) **(b)**

Fig. 3: (a) Graphical display of KNN results based on training data, (b) Graphical display of KNN results based on validation data

References

Alsheyab, M. A. & Khasawneh, M. A. (2022). "Quantifying the effect of modified mixture volumetrics and compaction effort on skid resistance of asphalt pavements," *Int. J. Pavement Eng.*, vol. 23, no. 5, pp. 1552–1560. doi: 10.1080/10298436.2020.1810688.

ASTM 867-02. (2015) "Standard Terminology Relating to Vehicle- Pavement Systems," doi:

https://doi.org/10.1520/e0867-02.

ASTM E 2380. (2005). “Standard Test Method for Measuring Pavement Texture Drainage Using an Outflow,” Am. Soc. Test. Mater., vol. i, pp. 2–5. https://doi.org/10.1520/e2380_e2380m-15.

ASTM E 303-93. (2002). “Standard Test Method for Measuring Surface Frictional Properties Using the British Pendulum Tester”. doi: 10.1520/E0303-93R13.2.

ASTM E 965-96. (2001). “Standard Test Method for Measuring Pavement Macrotexture Depth Using a Volumetric Technique”. https://doi.org/10.1520/e0965-15.

ASTM E1845-15. (2015). “Standard Practice For Calculating Pavement Macrotexture Mean Profile Depth”. https://doi.org/10.1520/e1845-09.

ASTM E1911-98. (2002). “Standard Test Method for Measuring Paved Surface Frictional Properties Using the Dynamic Friction Tester”. https://doi.org/10.1520/e1911-09ae01.

Bashar, M. Z. & Torres-Machi, C. (2021). “Performance of machine learning algorithms in predicting the pavement international roughness index,” *Transp. Res. Rec.* https://doi.org/10.1177/0361198120986171.

Cummings, P. (2010). “Modeling the Locked-Wheel Skid Tester to Determine the Effect of Pavement Roughness on the International Friction Index,” University of South Florida.

Gardner, M. W. & Dorling, S. R. (1998). “Artificial neural networks (the multilayer perceptron) - a review of applications in the atmospheric sciences,” *Atmos. Environ.*, vol. 32, no. 14–15, pp. 2627–2636, doi: 10.1016/S1352-2310(97)00447-0.

Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep Learning (Adaptive Computation and Machine Learning series)* Illustrated Edition, Illustrate. The MIT Press.

Hall et al. (2009). “Guide for Pavement Friction”. doi: 10.17226/23038.

Henry, J. J. (2000). “Evaluation of Pavement Friction Characteristics. NCHRP Synthesis of Highway Practice,” *Transp. Res. Board*, vol. 291. https://doi.org/10.5703/1288284314663.

Hofko et al. (2019). “A laboratory procedure for predicting skid and polishing resistance of road surfaces,” *Int. J. Pavement Eng.*, vol. 20, no. 4, pp. 439–447. doi: 10.1080/10298436.2017.1309191.

Hornik, K., Stinchcombe, M. & White, H. (1989). “Multilayer feedforward networks are universal approximators,” Neural Networks, vol. 2, no. 5, pp. 359–366. doi: 10.1016/0893-6080(89)90020-8.

Karballaeezadeh et al. (2020). “Intelligent road inspection with advanced machine learning; Hybrid prediction models for smart mobility and transportation maintenance systems,” *Energies*, vol. 13, no. 7. doi: 10.3390/en13071718.

Khasawneh, M. A. (2008). “The development and verification of a new accelerated polishing machine,” University of Akron.

___. (2015). “The prediction of LWST Values from DFT and CTM measurements using linear and nonlinear regression analyses,” *Jordan J. Civ. Eng.*, vol. 9, no. 4, pp. 487–498. doi: 10.14525/jjce.9.4.3121.

Khasawneh, M. A. & Alsheyab, M. A. (2020). “Effect of nominal maximum aggregate size and aggregate gradation on the surface frictional properties of hot mix asphalt mixtures,” *Constr. Build. Mater.*, vol. 244, p. 118355. doi: 10.1016/j.conbuildmat.2020.118355.

Kouchaki et al. (2018). “Field Investigation of Relationship between Pavement Surface Texture and Friction,” *Transp. Res. Rec.*, vol. 2672, no. 40, pp. 395–407. doi: 10.1177/0361198118777384.

Kulkarni, E. G. & Kulkarni, R. B. (2016). “WEKA Powerful Tool in Data Mining General Terms,” *Int. J. Comput. Appl.*, 2016.

Majidifard, H., Adu-Gyamfi, Y. & Buttlar, W. G. (2020). “Deep machine learning approach to develop a new asphalt pavement condition index,” *Constr. Build. Mater.*, vol. 247. doi: 10.1016/j.conbuildmat.2020.118513.

Marcelino et al. (2021). “Machine learning approach for pavement performance prediction,” *Int. J. Pavement Eng.*, vol. 22, no. 3, pp. 341–354. doi: 10.1080/10298436.2019.1609673.

Praticò, F. G. & Vaiana, R. (2015). "A study on the relationship between mean texture depth and mean profile depth of asphalt pavements," *Constr. Build. Mater.*, vol. 101, pp. 72–79. doi: 10.1016/j.conbuildmat.2015.10.021.

Quinlan, J. R. (1992). "Learning with continuous classes," *Aust. Jt. Conf. Artif. Intell.*, vol. 92, pp. 343–348.

Quiros, et al. (2018). "Interconversion of Locked-Wheel and Continuous Friction Measurement Equipment (CFME) Friction Measurements," *Transp. Res. Rec.*, vol. 2672, no. 40, pp. 452–462. doi: 10.1177/0361198118797455.

Shahin, M. A., Maier, H. R. & Jaksa, M. B. (…). "Predicting Settlement of Shallow Foundations using Neural Networks," *J. Geotech. Geoenvironmental Eng.*, vol. 128, no. 9, pp. 785–793.

Shmueli, G., Patel, N. R. & Bruce, P. C. (2006). Data mining for business intelligence: concepts, techniques, and applications in Microsoft Office Excel with xlminer, 1st edition. John Wiley & Sons.

Wang, Y. & Witten, I. H. (1997). "Inducing Model Trees for Continuous Classes," *Eur. Conf. Mach. Learn.*, pp. 1–10.

Webb, G. I. (2016). "Lazy Learning," in Encyclopedia of Machine Learning and Data Mining, pp. 2016–2017.

Yang, G. (2017). "Exploring Pavement Texture and Surface Skid Resistance Using Soft Computing Techniques," Oklahoma State University.

Yu et al. (2020). "Dynamic friction coefficient between tire and compacted asphalt mixtures using tire-pavement dynamic friction analyzer," *Constr. Build. Mater*, vol. 258, doi: 10.1016/j.conbuildmat.2020.119492.

Zhan et al. (2020). "Friction-ResNets: Deep Residual Network Architecture for Pavement Skid Resistance Evaluation," *J. Transp. Eng. Part B Pavements*, vol. 146, no. 3, p. 04020027. doi: 10.1061/jpeodx.0000187.

Cite as: Khasawneh M.A., Alsheyab M.A. & Al Akhrass H.I., "Modelling Asphalt Pavement Frictional Properties using Different Machine Learning Algorithms", *The 2nd International Conference on Civil Infrastructure and Construction (CIC 2023)*, Doha, Qatar, 5-8 February 2023, DOI: https://doi.org/10.29117/cic.2023.0075