IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 12, No. 1, March 2023, pp. 66~78
ISSN: 2252-8938, DOI: 10.11591/ijai.v12.i1.pp66-78

Neural network-based parking system object detection and predictive modeling

Ziad El Khatib[1], Adel Ben Mnaouer[1], Sherif Moussa[1], Omar Mashaal[1], Nor Azman Ismail[2], Mohd Azman Bin Abas[2], Fuad Abdulgaleel[2]
[1]Department of Electrical and Computer Engineering, Canadian University Dubai, Dubai, United Arab Emirates
[2]Department of Computer Science, Universiti Teknologi Malaysia, Johor Bahru, Malaysia

Article Info

Article history:

Received Nov 6, 2021
Revised Aug 31, 2022
Accepted Sep 19, 2022

Keywords:

Hyperparamter optimization
Predictive modeling
Real-time object detection
Regularization technique
Yolo neural network

ABSTRACT

A neural network-based parking system with real-time license plate detection and vacant space detection using hyper parameter optimization is presented. When number of epochs increased from 30, 50 to 80 and learning rate tuned to 0.001, the validation loss improved to 0.017 and training object loss improved to 0.040. The model means average precision mAP_0.5 is improved to 0.988 and the precision is improved to 99%. The proposed neural network-based parking system also uses a regularization technique for effective predictive modeling. The proposed modified lasso ridge elastic (LRE) regularization technique provides a 5.21 root mean square error (RMSE) and an R-square of 0.71 with a 4.22 mean absolute error (MAE) indicative of higher accuracy performance compared to other regularization regression models. The advantage of the proposed modified LRE is that it enables effective regularization via modified penalty with the feature selection characteristics of both lasso and ridge.

Corresponding Author:

Ziad El Khatib
Department of Electrical and Computer Engineering, Canadian University Dubai
City Walk, Dubai, United Arab Emirates
Email: Ziad.Elkhatib@cud.ac.ae

1. INTRODUCTION

With the rapid increase of car users the need for parking system with predictive modeling that allows learning by analyzing past user information is becoming essential. Prediction models are required since machine learning algorithms can predict future parking demand and the behavior of the parking users. Moreover, with trained neural network-based parking system license plate detectors, the parking system license plate detection and classification will improve with you only look once (YOLO) neural-network algorithm. YOLO provides real-time license plate detection [1]–[5]. Du *et.al.* [1] published work emphasis the need to have a multi-plate processing and detection [1] where YOLO can do that with increased speed and accuracy [6]–[8]. Masood *et.al.* published [9] work presents license plate detection and recognition using convolution neural network (CNN) with only 93.4% accuracy. Silva and Jung [3] published work presents license plate detection and recognition using CNN without optimizing neural-network hyperparameters [3]. Hyperparameter optimization is performed on the proposed general-purpose graphic unit (GPU)-based neural-network real-time parking system object detection. Also, other published work [4], [10] do not include hyperparameter optimization in their neural network processing. Nyambal and Klein their automated parking space detection using CNN achieving a 95.5% accuracy without license plate detection [10]. Fukusaki *et.al.* [11] also presented their published work on parking space detection using CNN without license plate detection [11]. Acharya *et.al.* [12] published work descirbes parking system with parking space neural network detection achieving high accuracy of 99.7% without any predicitive modeling processing [12]. Lin *et.al.* [13] and Idris *et.al.* [14] and Sarangi *et.al.* [15] in their published work presented a survey of smart parking system without any proposed design implementation. Applying existing machine learning in smart parking applications is investigated in [16]–[20]. However, they look at the data analytics without proposing a system implementation. Other published work [21]–[23] do not propose a machine learning model algorithm. Simhon *et.al.* [24] present smart parking system with predictive modeling in their published work without neural network object detection [24]–[26] published work describes parking system with predicitive modeling without license plate and parking space neural network detection. Other published work [27], [28] propose parking system implementation without looking at the machine learning data analytics algorithms and do not propose predictive modeling in their post processing system.

2. REAL-TIME NEURAL NETWORK OBJECT DETECTION WITH HYPERPARAMETER OPTIMIZATION

Hyperparameter optimization is applied to determine the optimal values of hyperparmaters such as optimal learning rate and the number of epoch in order to improve precision and accuracy [29]–[32]. Figure 1 shows neural network-based parking system real-time license plate detection with YOLO. YOLOv5 which is based on PyTorch framework provides real-time object detection with high accuracy and speed [6], [7].

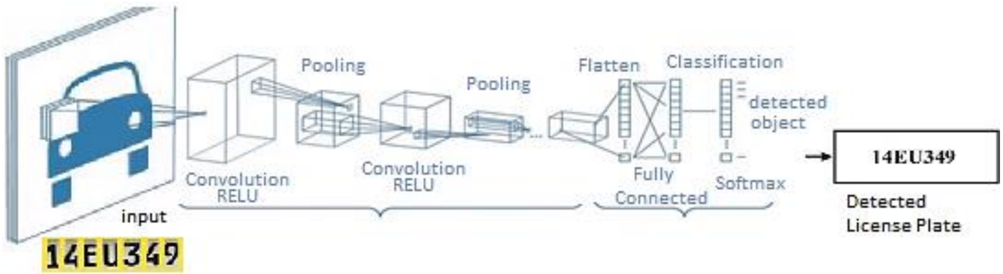

Figure 1. Neural network-based parking system real-time license plate detection

As shown in Figure 1 neural network-based parking system real-time license plate detection with YOLO is based on PyTorch framework provides real-time object detection with higher accuracy and speed [5], [7], [29]–[32]. YOLO takes the in a single instance by the framework and divides it into a grid with each grid having a dimension of n by n. Then places bounding box in the residual blocks and then determines the intersection over union (IOU). YOLO uses IOU to provide an output box that surrounds the object. YOLO then predicts the class probabilities for these boxes and their coordinates unlike CNN. After classification and localization are applied on each grid then the data that is labelled are passed to the model in order to train it. We determine YOLO loss function with (1).

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} (x_i - x'_i)^2 + (y_i - y'_i)^2 \tag{1}$$

Then the bounding box location (x, y) is determined with (2), when there is object the 1_{ij}^{obj} is 1 and 0 when there is no object.

$$+ \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [\left(\sqrt{w_i} - \sqrt{w'_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{h'_i}\right)^2] \tag{2}$$

The bounding box size (w, h) when there is object can be determined with (3).

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} (C_i - C'_i)^2 \tag{3}$$

The confidence when there is object is determined with (4).

$$+ \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{noobj} (C_i - C'_i)^2 \tag{4}$$

1_{ij}^{noobj} is 1 when there is no object, 0 when there is object. The class probabilities when there is object is determined with (5).

$$+\sum_{i=0}^{S^2} 1_{ij}^{obj} \sum_{c \in classes} \left(p_i(c) - p'_i(c)\right)^2 \quad (5)$$

3. REAL-TIME OBJECT DETECTION WITH HYPERPARAMETER OPTIMIZATION

In training neural network algorithm during learning model, it is important to look at the loss function in order to get intuition about how the neural network detection and classification are learning. Hyperparameter optimization is applied to determine the optimal values of hyperparmaters such as optimal learning rate and the number of epochs in order to improve precision and accuracy [33]. In training algorithm, epoch training setting start with 10 epochs then 50 and then 80 epochs. Epoch can be defined as how many times you pass once for learning the entire complete dataset through the neural network. The model is incrementally trained with more epoch which is increased in intervals of 10, 30, 50 and 80. Figure 2 shows the loss function with optimized hyperparameter learning rate as the number of epochs is increased. If we train the model with a lot of epochs this leads to overfitting of training model, whereas if we train the model with little epochs this leads to underfit model.

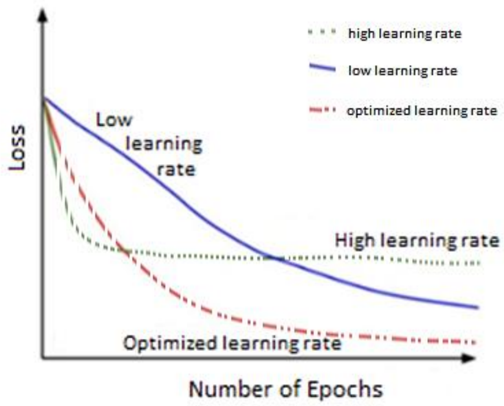

Figure 2. Loss function with optimized hyperparameter learning rate and number of epochs [33]

The learning-rate parameter decided how big step should be taken when searching for an optimal solution. The learning rate is tuned during hyperparameter optimization to improve the loss-function as shown in Figure 2 less learning-rate would require lots of epochs which increase the training time, however more learning-rate require fewer epochs. We adjust the learning rate during training incremently from high to low once we get closer to the optimal solution. We adjust the learning rate during training from high to low once we get closer to the optimal solution.

Validation loss is the loss calculated on the validation set, when the data is split using cross-validation [33]. If validation loss gets worse that indicates overfitting as can be seen in Figure 2. As long as the validation loss and training loss continues to improve, we keep optimizing the hyperparameters. The objective is to make the validation loss as low as possible to improve model accuracy. Learning rate adjust the weights and it will converge slower with lower value of the learning.

One of the most common evaluation metrics that is used in neural network object detection is 'mAP', which stands for 'mean average precision. A good mAP indicates a stable consistent model. Figure 3(a) and Figure 3(b) show the precision and the mean average precision for both training loss and validation loss performance as epoch is increased. We want to make the validation loss as low as possible to improve model accuracy. Figure 3(a) shows precision performance and Figure 3(b) shows mAP performance as the number of epochs is increased. As can be seen in Figure 3 when number of epochs increased from 30 to 50 and then to 80 the model mean average precision mAP_0.5 is improved to 0.99 and the precision is improved to 99%. Figure 4(a) shows training object loss and Figure 4(b) shows training class loss performance as the number of epochs is increased.

Figure 3. Accuracy performance as the number of epochs is increased (a) precision and (b) mAP

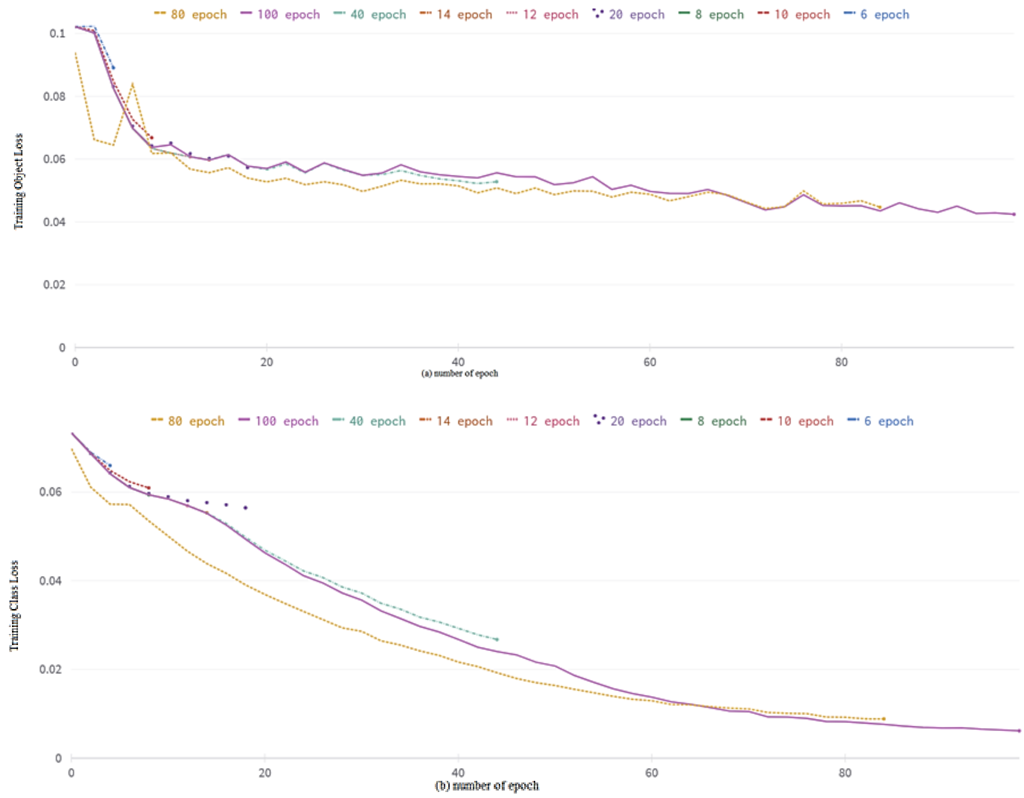

Figure 4. Accuracy performance as the number of epochs is increased (a) training object and (b) training class loss

As can be seen in Figures 5(a) and 5(b) when number of epochs increased from 30 to 50 and then to 80 and learning rate tuned to 0.001, the validation object loss improved to 0.017 and training object loss improved to 0.040. Figure 5(a) shows validation object loss and Figure 5(b) shows validation class loss performance as the number of epochs is increased. As can be seen in Figures 6(a) and 6(b) when number of epochs increased from 30 to 50 and then to 80 and learning rate tuned to 0.001, the validation box loss improved to 0.018 and training box loss improved to 0.017. Figure 6(a) shows training object loss and Figure 6(b) shows validation object loss performance as the number of epochs increased.

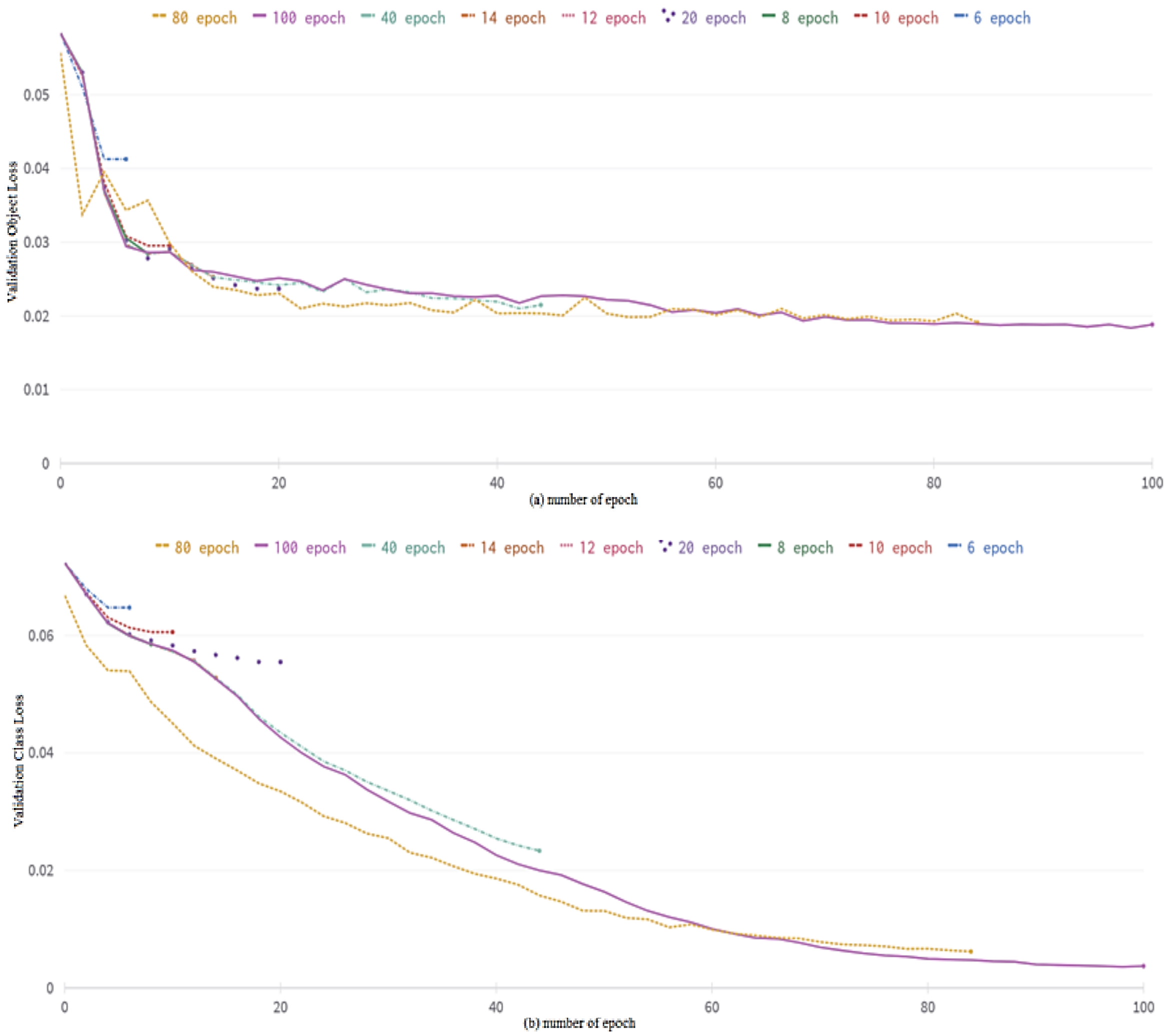

Figure 5. Accuracy performance as the number of epochs is increased (a) validation object loss and (b) validation class loss

The GPU-based neural network parking system real-time license plate detection with hyper parameter optimization model accuracy performance is shown in Table 1. When number of epochs increased from 30 to 50 and then to 80 and learning rate tuned to 0.001, the validation loss improved to 0.017 and training object loss improved to 0.040. Model mean average precision mAP_0.5 is improved to 0.988 and the precision is improved to 99%.

Table 1. Real-time neural-network object detection accuracy performance

Epoch	Learning Rate	mAP	Validation Object Loss	Training Object Loss	Precision
30	0.00001	0.652	0.030	0.060	0.50
50	0.0001	0.966	0.014	0.034	0.91
80	0.001	0.988	0.017	0.040	0.99

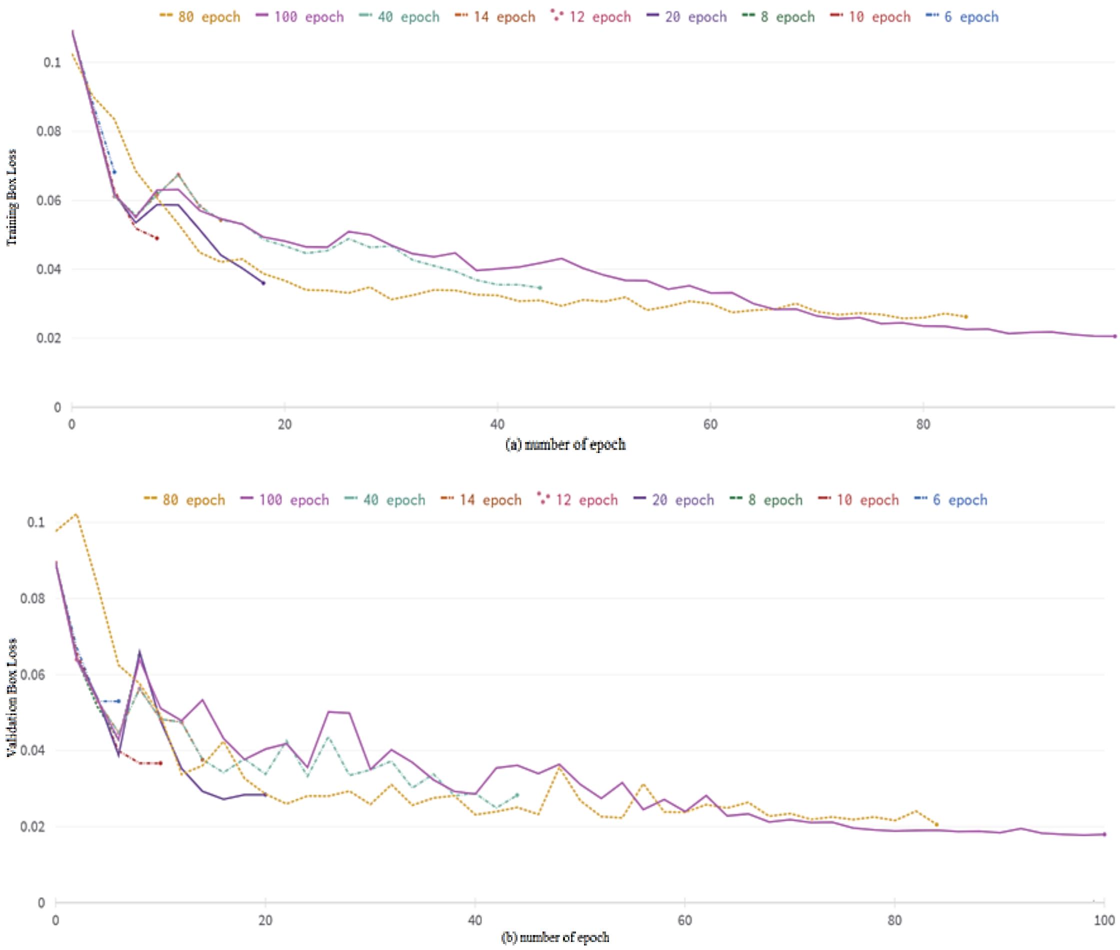

Figure 6. Accuracy performance as the number of epochs is increased (a) training box loss and (b) validation box loss

4. PREDICTIVE MODELING WITH REGULARIZATION TECHNIQUES

Figure 7 shows the proposed parking system with GPU processing using Nvidia Jetson Nano connected to artificial intelligence (AI) camera. The machine learning linear models provide a simple approach to predictive modeling. An overfit model is a model that fits the training dataset well but not the testing dataset as shown in Figure 8. The proposed parking system shown in Figure 7 uses Nvidia Jetson Nano connected to AI camera ce IMX 219 module. The AI camera connected to the Jetson board. A live video from the AI camera provides the real-time feed for vacant space detection. The 128-core Maxwell architecture-based GPU process the real-time processing analytics on the Jetson nano. Machine learning linear models provide a simple approach to predictive modeling [34]. An overfit model is a model that fits the training dataset well but not the testing dataset as shown in Figure 8. Overfitting causes low model accuracy. Regularization techniques solve the problem of overfitting [33], [35]–[37].

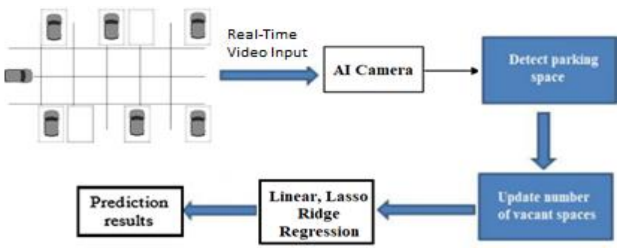

Figure 7. Proposed neural network-based parking system with real-time parking space detection and predictive modeling

4.1. Regularization techniques and overfitting

The objective is to have a machine learning model that has low bias and has low variability to produce consistent predictions across different datasets. Regularization is used in the proposed model to find the sweet spot between a simple model and a more complex model. Figure 8 shows the trade off between variance and bias in minimizing the prediction error [33], [35]–[37].

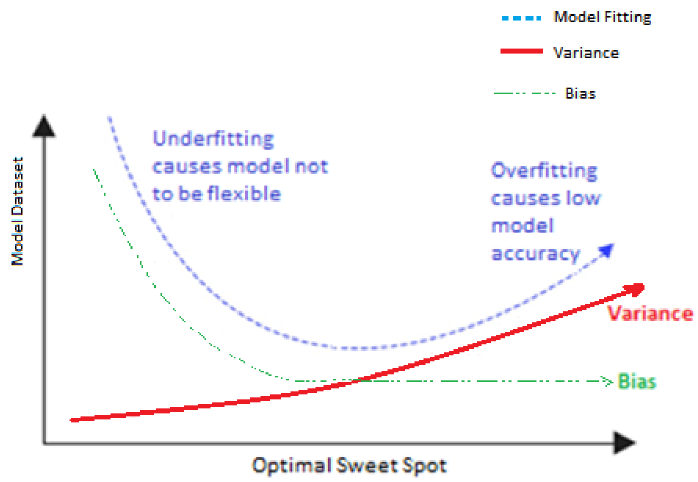

Figure 8. Regularization techniques solve the problem of overfitting

5. PROPOSED REGRESSION REGULARIZATION TECHNIQUE

Multiple linear regression uses a set of predictor variables and a response variable to fit a model of the form [33], [35]–[37].

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p \quad (6)$$

Y is the response variable and X is the predictor variable and βj is the slope. The values for β coefficients are chosen using the least square method which minimizes the residual sum of squares (RSS) [33], [35]–[37]. Regularization techniques function by penalizing the magnitude of coefficients along with minimizing the error between predicted and actual observations [38]. LASSO refers to least absolute shrinkable and selection operator [25]–[29], [31]. LASSO regularization is the process of adding a small modification to the cost function prevent the over-fitting problem as shown in (7) [33], [35]–[37].

$$J(m) = \Sigma\,(yi - \hat{y}i)^2 + \lambda.|\text{slope}| \quad (7)$$

Where λ is the tuning parameter.
Least squares regression attempts to find coefficient estimates that minimize the RSS. The yi is the actual and ŷi is the predicted value for the ith observation based on the multiple linear regression model [33], [35]–[37].

$$RSS = \Sigma\,(yi - \hat{y}i)^2 \quad (8)$$

Linear regression loss function is represented with Mean Squared Error function given by (9)

$$RSS = \frac{1}{n}\,\Sigma\,(yi - \hat{y}i)^2 \quad (9)$$

Lasso is analogous to linear regression however it shrinks the coefficients of determination towards zero [33], [35]–[37]. Lasso lets you shrink and regularize these coefficients work on multiple datasets. Lasso regression seeks to minimize the following. Lasso lets you regularize these coefficients to work on different datasets. The second term in (5) is known as a shrinkage penalty. Lasso regression performs L1 regularization value. Ridge regularization is a variation of LASSO as the term added to the cost function is depicted (10) and (11). Ridge regression cost function model is given by[33], [35]–[37].

$$RSS + \lambda.\ \Sigma\ |\beta_j| \qquad (10)$$

$$J(m) = \Sigma\ (yi - \hat{y}i)^2 + \lambda.|slope|^2 \qquad (11)$$

Ridge regression instead tries to minimize (12)

$$RSS + \lambda.\ \Sigma\ \beta_j^2 \qquad (12)$$

Ridge regression performs L2 regularization as shown in (7). A generalization of the ridge and lasso penalties, called the elastic net, combines the two penalties in (5) and (7). Elastic net regression seeks to minimize the following. The proposed modified lasso ridge elastic (LRE) regression model combines together both L1 and L2 regularization instead tries to minimize (13) and (14).

$$RSS + \lambda.\ \Sigma\ \beta_j^2 + \lambda.\ \Sigma\ |\beta_j| \qquad (13)$$

$$RSS + \lambda^2.\ \Sigma\ \beta_j^{3/2} + \lambda.\ \Sigma\ |\beta_j| \qquad (14)$$

The advantage of the modified LRE penalty is that it enables effective regularization via modified penalty with the feature selection characteristics of lasso and ridge penalty.

6. REGRESSION MODAL PARTIAL DERIVATIVES

Linear regression equations needed to calculate the partial derivatives with respect to parameters of the loss function [38]. The values of model parameters m and b are updated using (15) and (16) [33], [35]–[37]. The updated values will be the values with which each step reduces the difference between the true and predicted values.

$$\frac{\partial}{\partial m} = \frac{2}{\mathrm{N}} \sum_{i=1}^{N} -x_i\big(y_i - (mx_i + b)\big) \qquad (15)$$

$$\frac{\partial}{\partial b} = \frac{2}{\mathrm{N}} \sum_{i=1}^{N} -\big(y_i - (mx_i + b)\big) \qquad (16)$$

The values of model parameters m and b are updated using (17) to (20) [33], [35]–[37]. The updated values will be the values with which each step reduces the difference between the true and predicted values. Ridge regression equations needed to calculate the partial derivatives with respect to parameters of the loss function [33], [35]–[37].

$$m\ = \mathrm{m} - L_r.\frac{\partial L}{\partial m} \qquad (17)$$

$$b\ = \mathrm{b} - L_r.\frac{\partial L}{\partial b} \qquad (18)$$

$$\frac{\partial L}{\partial \beta_o} = -\sum_{i=1}^{N} 2\big(y_i - \beta_o - \sum_{j=1}^{p} \beta_j x_j\big) \qquad (19)$$

$$\frac{\partial L}{\partial \beta_j} = -\sum_{i=1}^{N} 2\big(y_i - \beta_o - \sum_{j=1}^{p} \beta_j x_j\big)\, x_i + 2\lambda\beta_j \qquad (20)$$

The proposed modified LRE regression model equations needed to calculate the partial derivatives with respect to parameters of the loss function. The advantage of the modified LRE penalty is that it enables effective regularization via modified penalty with the feature selection characteristics of lasso and ridge penalty as shown in Figure 9. Jupyter python was used for coding the machine learning regularization regression modified LRE model algorithm. Figure 9 shows the proposed modified LRE regression model with dataset and with different tuning parameter.

$$\frac{\partial L}{\partial \beta_o} = -\sum_{i=1}^{N} 2\big(y_i - \beta_o - \sum_{j=1}^{p} \beta_j x_j\big) \qquad (21)$$

$$\frac{\partial L}{\partial \beta_j} = -\sum_{i=1}^{N} 2\big(y_i - \beta_o - \sum_{j=1}^{p} \beta_j x_j\big)\, x_i + 1.5\lambda^2\beta_j^{1/2} + \lambda \qquad (22)$$

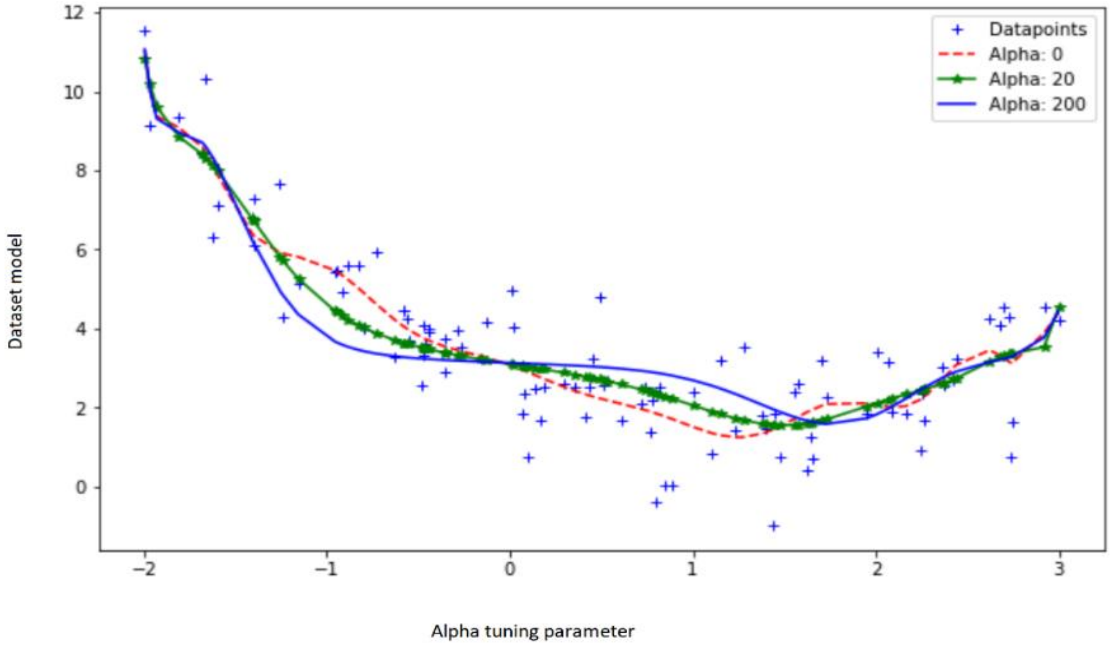

Figure 9. Proposed modified LRE regression model with dataset and with different tuning parameter

Figure 10 shows the linear regression predictive modeling forecasts in orange and green. The orange and green curves in both Figure 10 and Figure 11 are the future forcasts of the proposed predictive model the modified LRE indicating its effectiveness. Figure 11 shows proposed modified LRE regression predictive modeling forecasts in orange and green. Table 2 shows the proposed modified LRE regularization technique accuracy of 5.21 root mean square error (RMSE) and an R-Square of 0.71 with a 4.22 mean absolute error (MAE) compared to other regularization models.

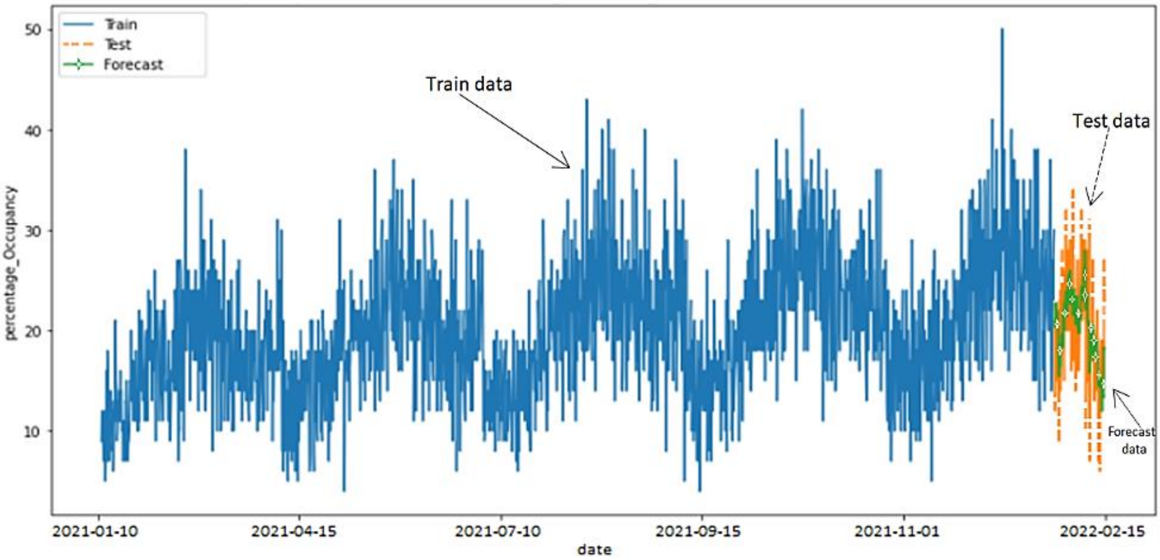

Figure 10. Linear regression predictive modeling forecasts in orange and green for test predictions

The neural network-based parking system with real-time license plate detection and vacant space detection using hyper parameter optimization is presented. When number of epochs increased from 30, 50 to 80 and learning rate tuned to 0.001, the validation loss improved to 0.017 and training object loss improved to 0.040. The model mean average precision mAP_0.5 is improved to 0.988 and the precision is improved to

99%. The proposed neural network-based parking system also uses a regularization technique for effective predictive modeling. The proposed modified LRE regularization technique provides a 5.21 RMSE and an R-square of 0.71 with a 4.22 MAE indicative of higher accuracy performance compared to other regularization regression models. The advantage of the proposed modified LRE is that it enables effective regularization via modified penalty with the feature selection characteristics of both lasso and ridge.

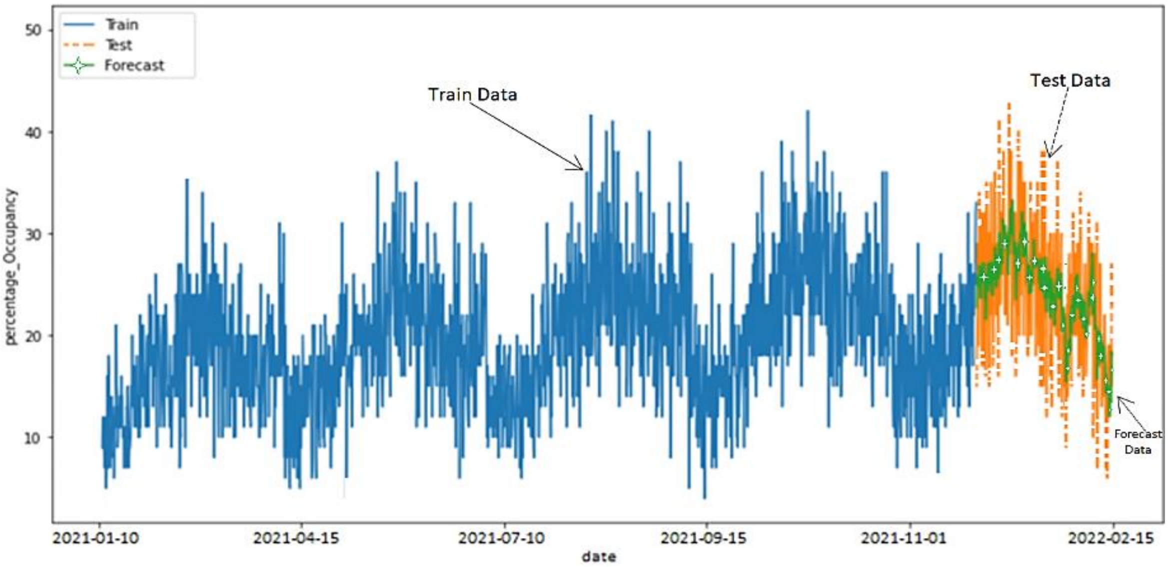

Figure 11. Proposed modified LRE regression predictive modeling forecasts in orange and green for test predictions

Table 2. Machine learning regression model accuracy performance for test predictions

Regression model	RMSE	R-Squared	MAE
Lasso Regression	5.45	0.66	4.52
Ridge Regresson	5.48	0.65	4.54
Elastic net Regresson	5.37	0.68	4.42
Linear Regression	5.72	0.63	4.67
Proposed Modified LRE Regression	5.21	0.71	4.22

7. CONCLUSION

The A neural network-based parking system with real-time license plate detection and vacant space detection using hyper parameter optimization has been presented. The model means average precision mAP_0.5 is 0.988 and the precision is 99%. The proposed neural network-based parking system uses a regularization technique for effective predictive modeling. The proposed modified LRE regularization technique provides a 5.21 RMSE and an R-square of 0.71 with a 4.22 MAE indicative of higher accuracy performance compared to other regularization regression models. The advantage of the proposed modified LRE is that it enables effective regularization via modified penalty with the feature selection characteristics of both lasso and ridge.

ACKNOWLEDGEMENTS

The authors thank all who contributed to the CUD-UTM research funding.

REFERENCES

[1] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic license plate recognition (ALPR): A state-of-the-art review," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 2, pp. 311–325, Feb. 2013, doi: 10.1109/TCSVT.2012.2203741.

[2] C. Gou, K. Wang, Y. Yao, and Z. Li, "Vehicle license plate recognition based on extremal regions and restricted boltzmann machines," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1096–1107, Apr. 2016, doi: 10.1109/TITS.2015.2496545.

[3] S. M. Silva and C. R. Jung, "Real-time Brazilian license plate detection and recognition using deep convolutional neural networks,"

in *Proceedings - 30th Conference on Graphics, Patterns and Images, SIBGRAPI 2017*, Oct. 2017, pp. 55–62, doi: 10.1109/SIBGRAPI.2017.14.
[4] O. Bulan, V. Kozitsky, P. Ramesh, and M. Shreve, “Segmentation-and annotation-free license plate recognition with deep localization and failure identification,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2351–2363, Sep. 2017, doi: 10.1109/TITS.2016.2639020.
[5] R. Panahi and I. Gholampour, “Accurate detection and recognition of dirty vehicle plate numbers for high-speed applications,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 767–779, Apr. 2017, doi: 10.1109/TITS.2016.2586520.
[6] J. Redmon and A. Farhadi, “YOLO9000: better, faster, stronger,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 6517–6525, 2017, doi: 10.1109/CVPR.2017.690.
[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: unified, real-time object detection,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2016, vol. 2016-Decem, pp. 779–788, doi: 10.1109/CVPR.2016.91.
[8] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: optimal speed and accuracy of object detection,” *arXiv*, 2020, doi: 10.48550/arXiv.2004.10934.
[9] S. Z. Masood, G. Shu, A. Dehghan, and E. G. Ortiz, “License plate detection and recognition using deeply learned convolutional neural networks,” *arXiv*, Mar. 2017, doi: 10.48550/arXiv.1703.07330.
[10] J. Nyambal and R. Klein, “Automated parking space detection using convolutional neural networks,” in *2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech)*, Nov. 2017, vol. 2018-Janua, pp. 1–6, doi: 10.1109/RoboMech.2017.8261114.
[11] T. Fukusaki, H. Tsutsui, and T. Ohgane, “An evaluation of a CNN-based parking detection system with Webcams,” in *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2020 - Proceedings*, 2020, pp. 100–103.
[12] D. Acharya, W. Yan, and K. Khoshelham, “Real-time image-based parking occupancy detection using deep learning indoor/outdoor seamless modelling, LBS, mobility view project The ISPRS benchmark on indoor modelling view project.” 2018, [Online]. Available: https://www.youtube.com/watch?v=Ft94ypd4HxE.
[13] T. Lin, H. Rivano, and F. Le Mouel, “A Survey of smart parking solutions,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3229–3253, Dec. 2017, doi: 10.1109/TITS.2017.2685143.
[14] M. Y. I. Idris, Y. Y. Leng, E. M. Tamil, N. M. Noor, and Z. Razak, “Car park system: a review of smart parking system and its technology,” *Information Technology Journal*, vol. 8, no. 2, pp. 101–113, Feb. 2009, doi: 10.3923/itj.2009.101.113.
[15] M. Sarangi, S. K. Das, and K. S. Babu, “Smart parking system: survey on sensors, technologies and applications,” in *1st IEEE International Conference on Advances in Information Technology, ICAIT 2019 - Proceedings*, Jul. 2019, pp. 250–255, doi: 10.1109/ICAIT47043.2019.8987378.
[16] N. Singh, S. Bawa, and H. Kaur, “Enhanced economy based smart parking system using machine learning,” *Proceedings of Industry Interactive Innovations in Science, Engineering & Technology (I3SET2K19)*, 2020, doi: 10.2139/ssrn.3516600.
[17] J. Fan, Q. Hu, and Z. Tang, “Predicting vacant parking space availability: an SVR method with fruit fly optimisation,” *IET Intelligent Transport Systems*, vol. 12, no. 10, pp. 1414–1420, Oct. 2018, doi: 10.1049/iet-its.2018.5031.
[18] Z. Zhao and Y. Zhang, “A comparative study of parking occupancy prediction methods considering parking type and parking scale,” *Journal of Advanced Transportation*, vol. 2020, pp. 1–12, Feb. 2020, doi: 10.1155/2020/5624586.
[19] F. M. Awan, Y. Saleem, R. Minerva, and N. Crespi, “A comparative analysis of machine/deep learning models for parking space availability prediction,” *Sensors (Switzerland)*, vol. 20, no. 1, p. 322, Jan. 2020, doi: 10.3390/s20010322.
[20] I. Masmoudi, A. Wali, A. Jamoussi, and A. M. Alimi, “Vision based system for vacant parking lot detection: VPLD,” in *VISAPP 2014 - Proceedings of the 9th International Conference on Computer Vision Theory and Applications*, 2014, vol. 2, pp. 526–533, doi: 10.5220/0004730605260533.
[21] G. Amato, F. Carrara, F. Falchi, C. Gennaro, and C. Vairo, “Car parking occupancy detection using smart camera networks and deep learning,” in *Proceedings-IEEE Symposium on Computers and Communications*, Jun. 2016, vol. 2016-Augus, pp. 1212–1217, doi: 10.1109/ISCC.2016.7543901.
[22] Q. Wu, C. Huang, S. Y. Wang, W. C. Chiu, and T. Chen, “Robust parking space detection considering inter-space correlation,” in *Proceedings of the 2007 IEEE International Conference on Multimedia and Expo, ICME 2007*, Jul. 2007, pp. 659–662, doi: 10.1109/icme.2007.4284736.
[23] S. Lee, D. Yoon, and A. Ghosh, “Intelligent parking lot application using wireless sensor networks,” in *2008 International Symposium on Collaborative Technologies and Systems, CTS'08*, May 2008, pp. 48–57, doi: 10.1109/CTS.2008.4543911.
[24] E. Simhon, C. Liao, and D. Starobinski, “Smart parking pricing: a machine learning approach,” in *2017 IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2017*, May 2017, pp. 641–646, doi: 10.1109/INFCOMW.2017.8116452.
[25] J. C. Provoost, A. Kamilaris, L. J. J. Wismans, S. J. van der Drift, and M. van Keulen, “Predicting parking occupancy via machine learning in the web of things,” *Internet of Things (Netherlands)*, vol. 12, p. 100301, Dec. 2020, doi: 10.1016/j.iot.2020.100301.
[26] J. Barker and S. U. Rehman, “Investigating the use of machine learning for smart parking applications,” Oct. 2019, doi: 10.1109/KSE.2019.8919291.
[27] A. Zacepins, V. Komasilovs, and A. Kviesis, “Implementation of smart parking solution by image analysis,” in *VEHITS 2018 - Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport Systems*, 2018, vol. 2018-March, pp. 666–669, doi: 10.5220/0006629706660669.
[28] Y. Ma, Y. Liu, L. Zhang, Y. Cao, S. Guo, and H. Li, “Research review on parking space detection method,” *Symmetry*, vol. 13, no. 1, pp. 1–18, Jan. 2021, doi: 10.3390/sym13010128.
[29] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask R-CNN,” in *Proceedings of the IEEE International Conference on Computer Vision*, Oct. 2017, vol. 2017-Octob, pp. 2980–2988, doi: 10.1109/ICCV.2017.322.
[30] R. L. Galvez, A. A. Bandala, E. P. Dadios, R. R. P. Vicerra, and J. M. Z. Maningo, “Object detection using convolutional neural networks,” in *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, Oct. 2019, vol. 2018-Octob, pp. 2023–2027, doi: 10.1109/TENCON.2018.8650517.
[31] Ulagamuthalvi, J. B. J. Felicita, and D. Abinaya, “An efficient object detection model using convolution neural networks,” in *Proceedings of the International Conference on Trends in Electronics and Informatics, ICOEI 2019*, Apr. 2019, pp. 142–147, doi: 10.1109/ICOEI.2019.8862698.
[32] M. A. Shehab, A. Al-Gizi, and S. M. Swadi, “Efficient real-time object detection based on convolutional neural network,” in *2021 International Conference on Applied and Theoretical Electricity, ICATE 2021 - Proceedings*, May 2021, pp. 1–5, doi: 10.1109/ICATE49685.2021.9465015.

[33] P. Bühlmann and T. Hothorn, "Boosting algorithms: Regularization, prediction and model fitting," *Statistical Science*, vol. 22, no. 4, pp. 477–505, Nov. 2007, doi: 10.1214/07-STS242.
[34] F. J. W. M. Dankers, A. Traverso, L. Wee, and S. M. J. van Kuijk, "Prediction modeling methodology," in *Fundamentals of Clinical Data Science*, Springer International Publishing, 2018, pp. 101–120.
[35] D. Schreiber-Gregory and H. M. Jackson Foundation, "Regulation techniques for multicollinearity: lasso, ridge, and elastic nets," in *Proceedings of the SAS Conference Proceedings: Western Users of SAS Software*, 2018, pp. 1–23.
[36] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, vol. 67, no. 2, pp. 301–320, Apr. 2005, doi: 10.1111/j.1467-9868.2005.00503.x.
[37] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, no. 1, pp. 1–22, 2010, doi: 10.18637/jss.v033.i01.
[38] A. V Dorugade and D. N. Kashid, "Alternative method for choosing ridge parameter for regression," *Applied Mathematical Sciences*, vol. 4, no. 9, pp. 447–456, 2010.

BIOGRAPHIES OF AUTHORS

Dr. Ziad El-Khatib PhD in Electrical and Computer Engineering from Carleton University Canada. Assistant professor at Canadian University Dubai. Dr. Ziad El-khatib received his Bachelor of Science in Electrical Engineering from University of Ottawa Canada and his M.A.Sc. and PhD in Electrical and Computer Engineering from Carleton University Canada. He has several years of design experience in the field of communication integrated circuits at various companies including Nortel Networks Harris Corporation Chrysalis-ITS Semiconductor Itron Inc. and was adjunct professor in USA. He is currently assistant professor in the faculty of Electrical and Computer Engineering at Canadian University Dubai. His research interests include silicon based integrated circuits for radio frequency and microwave communications and power electronics integrated circuits. He has a book published through Springer on radio frequency amplification and linearization techniques and numerous IEEE journal and conference papers. He can be contacted at email: ziad.elkhatib@cud.ac.ae.

Dr. Adel Ben Mnaouer PhD in Computer Engineering Networking from Yokohama University Japan. Professor at Canadian University Dubai. Dr. Mnaouer holds a PhD in Computer Engineering Networking from Yokohama National University, Yokohama, Japan. He also obtained a Master of Engineering (Petri Nets) from Fukui University, Japan and a BSc in Computer Science from Ecole Superieur de Communications de Tunis. Prior to joining CUD, Dr. Mnaouer was Associate Professor and Vice Dean of Research at Dar Al Uloom University, Saudi Arabia. Prior to this he held academic posts at a range of institutions, including the University of Trinidad and Tobago, Carthage University, Tunis, Nanyang Technological University, Singapore and Sultan Qaboos University, Singapore. He can be contacted at email: adel@cud.ac.ae.

Dr. Sherif Moussa PhD in Electrical and Computer Engineering from University of Quebec Trois-Riviers, Canada. Associate professor at Canadian University Dubai. Dr. Sherif Moussa received his PhD in Electrical and Computer Engineering from University of Quebec Trois-Riviers, Canada, and his MSc degree in Electrical and Computer Engineering form University of Waterloo, Canada. His research areas are wireless communication, computer networks, and VLSI design. His research specifically focuses on MIMO-OFDM algorithms, multiple access OFDM, FPGA design and optimization. Dr. Moussa joined CUD in 2007 where he currently is working as an Assistant Professor at School of Engineering. Prior to joining CUD, he was a lecturer at School of Engineering, Centennial College, Toronto, Canada. Dr. Moussa is currently an active researcher who published in many international journals and conferences related to his field and he also currently serve as a reviewer and technical committee member for many international conferences. Dr. Moussa is the winner of 2015 CUD research excellence award and the founder of the flagship CUD robotics club. He can be contacted at email: smoussa@cud.ac.ae.

Omar Mashaal Masters in Communications Engineering from University of Technology Malaysia. Lecturer at Canadian University Dubai. Mr. Mashaal holds M.Eng in Communication Engineering from University of Technology, Malaysia and a BSc in Electrical Engineering - Communication Engineering from Ajman University. He worked as a voice and network engineer for two years and attained different industrial certificates from Alcatel-Lucent and Avaya. Mr. Mashaal attended several technical trainings, and he is a certified security intelligence analyst–educator by IBM. Mr. Mashaal is a member of the Jordan Engineers Association. His research interests are in antenna engineering and communication systems. He can be contacted at email: omar.mashaal@cud.ac.ae.

Dr. Nor Azman Ismail PhD in Computer Science and Computer Engineering from Loughborough. Associate Professor at University Teknologi Malaysia. Deputy Director of Office of Corporate Affairs (Web Director) and an academic staff at Computer Graphics and Multimedia Department, Universiti Teknologi Malaysia (UTM) for about thirteen years. Prior to my appointment as a University Web Director, He was Research Coordinator of Computer Graphics and Multimedia Department. In the earliest stages of my career, He was employed by Conner Peripherals a company that manufactured hard drives for personal computers and then Perak Department of Education. He can be contacted at email: azman@utm.my.

Dr. Mohd Azman bin Abas PhD in Computer Science and Computer Engineering from University Teknologi Malaysia. Professor at University Teknologi Malaysia. Director, Automotive Development Centre (ADC), Institute for Vehicle Systems and Engineering (IVeSE) Universiti Teknologi Malaysia (UTM). Research areas/interest, internal combustion engine, vehicle drive cycle, vehicle driving behaviour, autonomous driving behaviour, diver behaviour, motorsports. He can be contacted at email: azman.abas@utm.my.

Dr. Fuad Abdulgaleel PhD in Computer Science and Computer Engineering from University Teknologi Malaysia. Professor at University Teknologi Malaysia. Senior Lecturer at School of Computing, Faculty of Engineering and member in the Information Assurance and Security Research Group (IASRG). My research interests are – cyber threat intelligence, network security, misbehavior detection, anomaly detection, and malware analysis. He can be contacted at email: mdfarid@utm.my.