Tire Condition Classification Based on Tread Depth using Machine Learning

Mostafizur Rahman
Department of Electical Electronic and System Engineering
Faculty of Engineering and Built Enviroment, University Kebangsaan Malaysia, Selengor, Malaysia
Email:p111469@siswa.ukm.edu.my

Noorfazila Kamal
Department of Electical Electronic and System Engineering
Faculty of Engineering and Built Enviroment, University Kebangsaan Malaysia, Selengor, Malaysia
Email: fazila@ukm.edu.my

Nor Fadzilah Abdullah
Department of Electical Electronic and System Engineering
Faculty of Engineering and Built Enviroment, University Kebangsaan Malaysia, Selengor, Malaysia
Email: Fadzilah.abdullah@ukm.edu.my

Fikri Mat Hussin
Departmen of Electical Electronic and System Engineering
Faculty of Engineering and Built Enviroment, University Kebangsaan Malaysia, Selengor, Malaysia
Email: a164850@ukm.edu.my

2022 IEEE 20th Student Conference on Research and Development (SCOReD) | 978-1-6654-7488-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/SCORED57082.2022.9973993

***Abstract*— As a crucial component for vehicle performance and stability, automotive manufacturers adopt specific tire inspection procedures to verify its quality and integrity. Tires stay in good condition when they are checked frequently for things like tread depth and pressure. Moreover, tire tread depth has a significant impact on a vehicle's braking ability and should be routinely evaluated to ensure driver safety. In order to identify and classify tire condition, a state-of-the-art research approach is carried out using machine learning algorithms: Extreme Gradient Boosting (XGBoost), AdaBoost, Naïve Bayes, Support Vector Machine (SVM), Decision Tree and Random Forest. A primary data set is taken from tire tread depth measurements on 100 tires using tread depth gauges and tire pressure gauges. The results recorded 95% accuracy and recall, 95.35% F1 and 96.67% precision scores for both XGBoost and SVM. The outcome shows the successful implementation of machine learning algorithms for tire condition classification.**

Keywords—trade depth, tire, machine learning, svm, xgboost, adaboost, random forest.

I. INTRODUCTION

The tire tread depth is designed to improve the friction between the tire tread and the road and to remove water from between the tire and the road, preventing the vehicle from slipping. Numerous studies have examined the effect of tire properties on vehicle behavior, including wheel diameter, tire tread depth, and tire pressure [1-3]

In the past few years, governments worldwide have emphasized tire testing, especially to find the tire pressure and tread depth, to make driving safer. If the tire has no tread, it won't grip the road well enough, which makes driving less safe [4]. For safety reasons, the tread depth of a vehicle's tires can't be less than 1.6 millimeters or 2/6 inches [5] .

Usually, a depth indicator is used to measure the depth of the tire tread when the vehicle is stopped. This type of measurement is time consuming, person-dependent, and inconvenient [6]. However, in order to increase productivity and reduce manual labor, some are trying to standardize the measurements of aligned tire treads to derive the tire profile [7]. For automatic measurement, it is necessary to be able to identify the tread on the tire's outermost surface.

Lack of early diagnosis of tire health can cause the tire to be in a dangerous condition. Currently, there is an automatic tire pressure detection system developed to monitor the pressure in the tire. Several works have also been done to estimate tire tread depth. An automatic tire tread depth monitoring system is one way to keep drivers informed about their tires and improve road safety. However, automatic tire condition classification based on tread depth has yet to be explored. Recent improvements in machine learning and computer power have made it possible for tires to be monitored fully automatically.

In this research, a machine learning-based approach was used to determine tire condition classification. The main contribution of this work is to create a primary dataset and compare the most popular machine learning algorithms to develop a benchmark to accelerate research for tire condition classification based on tire tread depth.

The rest of the sections of the paper are organized as follows: Sect. II discusses the tread depth classification techniques. Sect. III describes the methodology along with implemented models. Section IV presents the experimental results, and Sect. V concludes this research work and scopes for future improvement.

II. TREAD DEPTH ESTIMATION

There have been a very small number of academic papers written about the analysis of tire tread images. Most papers are about classifying images of tire tread patterns and machine vision. The majority emphasizes textured features, such as the graphical representation of various tire tread patterns.

Machine vision can be used to come up with a new way to measure the depth of a tire's tread [8]. The methods may be implemented in a LabVIEW vision module-based program. Based on the created mathematical model and identification algorithms, the measurement procedure comprises picture capture and processing and the identification and computation of the grooves' depth. The experiment shows that the groove in a tire tread can be found, and its depth can be measured. Furthermore, the absolute error in depth is less than 0.2 mm, so it can be used to measure the depth of tire treads. But this system needs to check the depth in a specific place. Predicting the depth of a tire's tread is challenging and costly.

Authorized licensed use limited to: UNIV OF ENGINEERING AND TECHNOLOGY LAHORE. Downloaded on April 30,2024 at 18:02:57 UTC from IEEE Xplore. Restrictions apply.

In the tire vulcanization process, the authors [9] focus on a problem involving a deviation between the tread and design patterns. They provide an innovative approach for extracting and reconstructing tread pattern features from tire tread scan data. In the initial step of the segmentation process, the pattern pitch arrangement is employed to analyze the data. The contour of the tire crest is then obtained after this step. And as the last step, the point cloud is converted into a two-dimensional grayscale image. The properties of the pattern's border are recognized and extracted following the growing rhythm of the design. In order to discretize the pattern border, the cubic boundary spline interpolation method is utilized in a series of central coordinate points.

Another work in [10] also uses machine vision and image triangulation to detect tread depth to decrease human costs and enhance the convenience of measuring based on the epipolar plane. The experiments demonstrated that tire tread images could be measured more accurately with a brightness level of 7500 lux. The LabVIEW stereo vision SGM algorithm was used in this work. The correct tire tread depth could be determined from these images, which could be transferred to a user's smartphone or tablet via the MySQL database. However, this system is still in the experimental stage, with no real-life implications yet, and if it necessary to check the tire tread depth, it must go to a specific place.

Work in [11] provides a tread pattern classification system with transfer learning-based feature fusion. In the first stage of the procedure, the information of a previously trained convolutional neural network (CNN) model is transferred to a new model using the ImageNet dataset. Using tread pattern image information, parameters of the new model are modified. In the second stage of the method, features are taken from numerous linked layers. These are high-level characteristics of tread pattern photographs. Histogram of oriented gradients (HOG) is calculated as the low-level characteristic of the tread pattern image in the third stage. In the fourth stage, CNN model features are fused with HOG characteristics as a fusion feature. Finally, the fused features are utilized to train an SVM classifier for image categorization of tread pattern. The notion of transfer learning, however, overcomes the problem of a small training dataset.

In [12], a strategy is provided for visual, non-contact, automated tire tread depth inspection based on existing image processing techniques. For tire tread depth examination, the suggested approach employs the histogram-oriented gradient and a support-vector machine. The acquired classification findings, with a 99% accuracy and an F1-score of 0.9922, are encouraging. In addition, the testing of classifier robustness has provided positive results. Nonetheless, this dataset needs expansion and more experiments with other tire kinds. The system is, therefore, not 100% automated.

III. Methodology

This section discusses the methodology of tire condition classification based on tread depth using machine learning algorithms and processes.

A. Data Collection

The dataset consists of 100 tire tread depth values collected using tread depth gauge. For each tire three tread depth values are measured, which are outer, center, inner of the tire surface. Based on the tread depth values, the tire condition is categorized into five final classes as stated in Table 1. Machine learning is used to classify the criteria to identify tire conditions.

TABLE I. Criteria listing for target class.

Final class	Criteria
1	Tires are in good condition
2	Tires need to be changed
3	Over inflate
4	Under inflate
5	Alignment

B. Data Preprocessing

Preprocessing is performed to prepare a standardized dataset for the classifiers. The stage begins by creating a feature indexing array. Then a boxplot diagram is used to show the overall response patterns for the five criteria, as shown in Fig. 1. It provides a valuable way to visualize the range and other characteristics of responses for large classes. The heatmap is generated to understand the correlation between features, as shown in Fig. 2, where the three tire tread depths show a weak correlation with tire pressure at the initial stage. To fix the imbalance issue, two class names, LC, and RC, are calculated by measuring the difference between left and center (LC) and right and center (RC). The values are then standardized based on the following conditions:

- If, $-0.3 \leq LC/RC \leq 0.3$; ClassLC/Class RC = 0
- If, $LC/RC < -0.3$; ClassLC/Class RC = 1
- If, $LC/RC > -0.3$; ClassLC/Class RC = 2

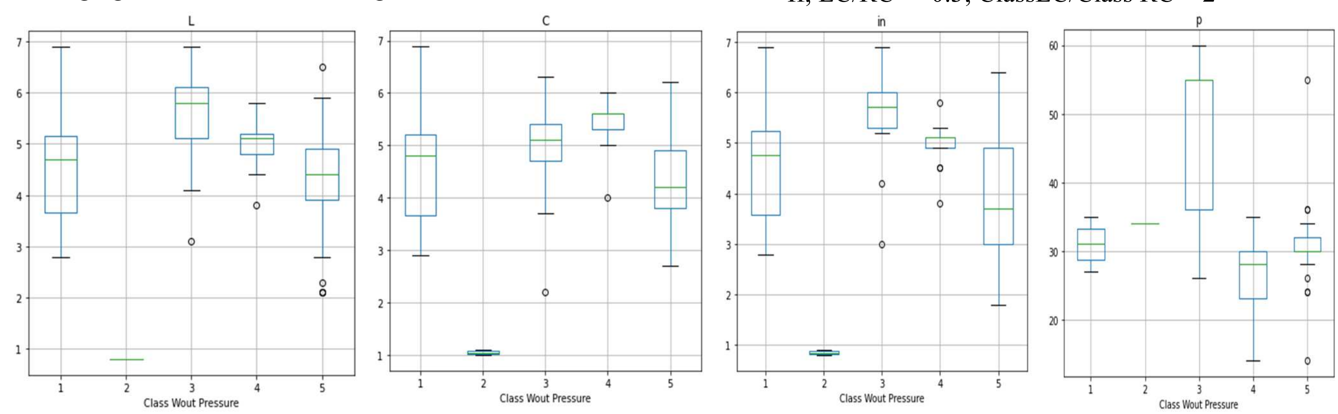


Fig. 1. Boxplot based data visualization

Authorized licensed use limited to: UNIV OF ENGINEERING AND TECHNOLOGY LAHORE. Downloaded on April 30,2024 at 18:02:57 UTC from IEEE Xplore. Restrictions apply.

- If, 25 ≤ Pressure ≤ 30; ClassP = 0
- If, Pressure < 25; ClassP = 1
- If, Pressure > 30; ClassP = 2

The correlation after standardization shows sufficient dependency between the features, improving from negative to positive correlations. Further, the dataset has been split into 80:20 ratio for training and testing.

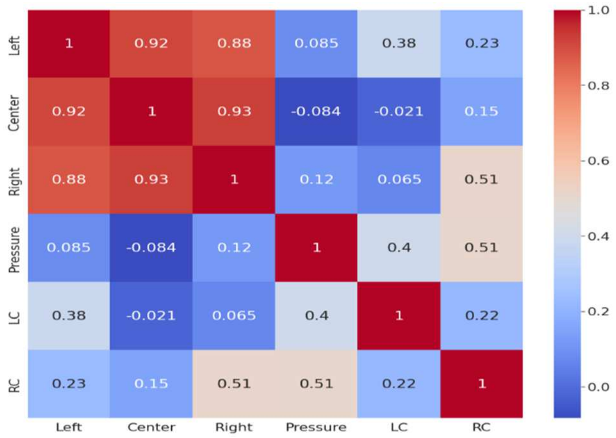


Fig. 2. Correlation heatmap based on the primary dataset.

C. *Implementation of Algorithms*

Several machine learning algorithms were selected based on the literature review to be used and the results were compared to build a suitable classifier to determine the tire condition based on the tire tread depth. In this study, the selected algorithms are SVM, Extreme Gradient Boosting (XGBoost), Random Forest, AdaBoost, Decision Tree and Naïve Bias [13-19]. SVM was selected due to its short training period with its global optimizer. Whereas, gradient tree-based approaches such as basic decision trees, XGBoost and AdaBoost algorithms are implemented after using a 'bagging' ensemble meta-algorithm that aggregates predictions from several decision trees using majority voting. This bagging method is derived to create a forest or collection of decision trees by randomly selecting features. Naive Bias and Random Forest are procedures based on selecting the most likely hypothesis. This method aids the initial classification process since it implicitly optimises any loss function over the entire ensemble.

To tune the parameters to make the algorithms more optimized with the dataset, a grid search was used. Based parameters differ as they are known to be hyper-parameters and are not directly learned in the estimators themselves [20]. After establishing the classifier from python skarn module, a parameter grid is created that searches through to find the maximum optimization of hyper-parameters. The model will refer to the combination of all grid items provided by the trail-and-error method, different for each algorithm with different learning rate.

For proper feature engineering and feature selection, Recursive Feature Elimination (RFE) was applied, which uses all features to fit a model. The elements will, in turn, be ranked according to the order of their importance [21]. Estimators were trained, and each feature is assigned a particular weight. After each round of recursive model creation, the elements with the smallest weights are removed. Components that perform very well are kept. As recursive feature elimination acts like a meta-algorithm, it uses a secondary machine learning algorithm. In this study the process for RFE is based on the Logistic Repressor. The implementation is performed by adopting the Python programming language 3.7 version on the Jupiter Notebook platform.

After the preprocessing, the selected machine learning algorithms was applied, where the model was trained on the training dataset of 80% and 20% was used for the test dataset, where a total of 6 variables were under the first criteria, 1 for the second, 2 for the third and fourth, and 9 for the fifth criteria. The Scikit-learn library was used to implement machine learning algorithms and fine tuning to develop the proper classifier.

D. *Evaluation Matrix*

The confusion matrix provides a comprehensive perspective of the performance, including a breakdown of right and wrong predictions for each class [22, 23]. The performance is evaluated by comparing expected and actual result values. In a confusion matrix, True positives (TP) correspond to the number of positive examples correctly predicted by the model, False negatives (FN) represent the number of positive examples incorrectly predicted as negative, False positives (FP) refer to the number of negative examples incorrectly predicted as positive, and True negatives (TN) represent the number of negative examples correctly predicted. Based on the confusion matrix, the following equation from 1 to 4 are applied to calculate accuracy, F1 score, recall and precision score.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

$$Recall = \frac{TP}{TP+TN} \tag{2}$$

$$Precision = \frac{TP}{TP+FP} \tag{3}$$

$$F1 = 2 * \left(\frac{Precision * Recall}{Precision + Recall}\right) \tag{4}$$

IV. Result and Discussion

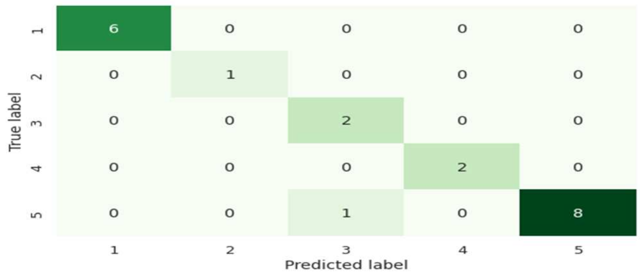
The evaluation process applying confusion matrix is illustrated in Fig 3 for all the algorithms.

TABLE II. Performance comparison of selected algorithms

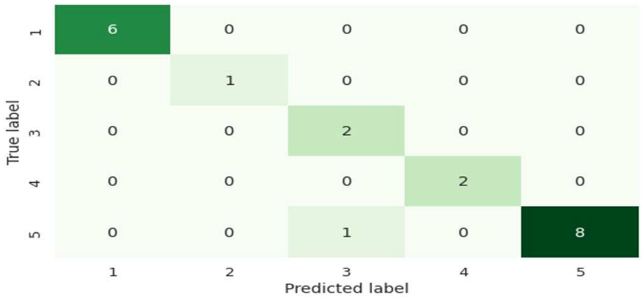
Algorithm	Accuracy (%)	F1 Score (%)	Recall (%)	Precision (%)
SVM	95	95.35	95	96.67
AdaBoost	50	33.93	50	26.31
Naïve Bayes	90	88	90	86.67
Decision Tree	90	90.37	90	93.33
XGBoost	95	95.35	95	96.67
Random Forest	85	83.72	85	84.37

Authorized licensed use limited to: UNIV OF ENGINEERING AND TECHNOLOGY LAHORE. Downloaded on April 30,2024 at 18:02:57 UTC from IEEE Xplore. Restrictions apply.

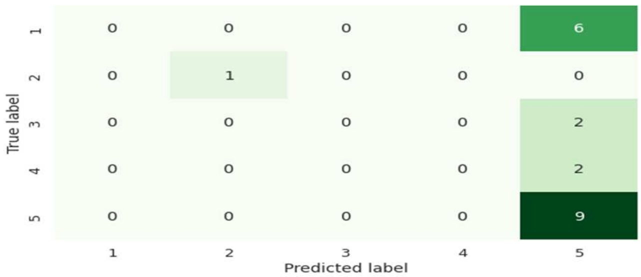
The confusion matrix shows that SVM and XGBoost together demonstrated the best performance. From Table II, the results are compared where it shows both SVM and XGBoost scores are the highest for all four measurements where the accuracy and recall are 95%, F1 score is 95.35% and precision is 96.67%.



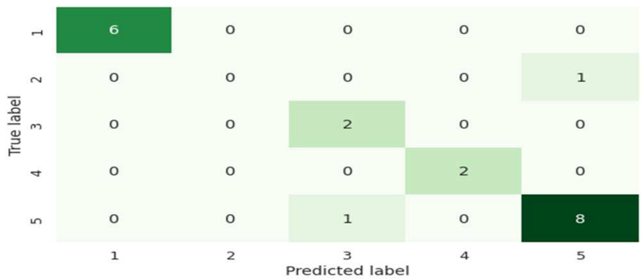
(a)



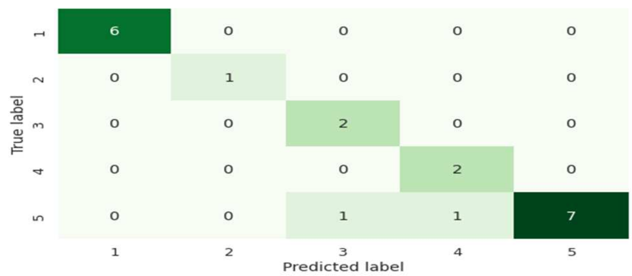
(b)



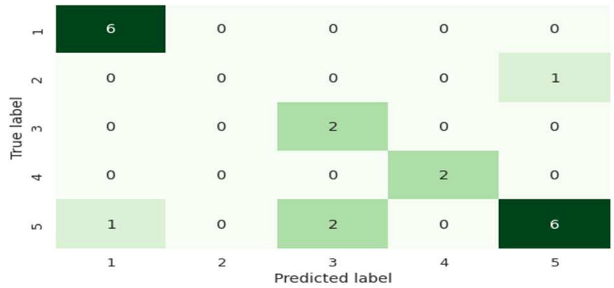
(c)



(d)



(e)



(f)

Fig. 3. Confusion matrix evaluation of (a) SVM, (b) XGBoost, (c) AdaBoost, (d) Naïve Bayes, (e) Decision Tree and (f) Random Forest

The overall performance indicates that SVM and XGBoost show almost similar results which are efficient to build tire condition classification after being evaluated by F1 score, precision score, sensitivity score and accuracy. With a total of 20 testing tire data, all classes obtain full accuracy predicting the 5th class and both predict the 3rd class as a true negative. The evaluations show that the true positive rate and true negative rate are sustainable for deploy in tire health classification. However, AdaBoost performed worst with 50% accuracy and 26.31% precision score due to its tendency to classify the fifth criteria every single time. Furthermore, Naïve Bayes and Decision Tree obtain the same accuracy of 90%, however, there is a slight difference in precision scores, resulting in Naïve Bayes' struggle for the second criteria. For all classifiers, it appears that most errors occur due to a tendency to classify the fifth criteria as the third in most cases.

V. Conclusion

The implementation of the machine learning algorithm on the tire tread depth dataset shows a successful demonstration for tire condition classification where SVM and XGBoost obtained the best performance scores of 95% accuracy and recall, 95.35% F1 and 96.67% precision scores. This is the first attempt to deploy a machine learning algorithm on tread depth for tire condition classification. This approach can be improved in the future and make more robust systems for safety driving. Expanding the dataset and testing on time series datasets is planned for future development. Moreover, performance can be improved by creating weights and implementing large testing dataset after fine tuning and feature engineering.

References

[1] H. Cui and G. Liu, "How noncoding RNAs contribute to macrophage polarization," in MicroRNAs and Other Non-Coding RNAs in Inflammation: Springer, 2015, pp. 59-84.

[2] S. Sivaramakrishnan, K. B. Singh, and P. Lee, "Influence of Tire Operating Conditions on ABS Performance," Tire Science and Technology, vol. 43, no. 3, pp. 216-241, 2015.

[3] Z. Du, X. Wen, D. Zhao, Z. Xu, and L. Chen, "Numerical Analysis of Partial Abrasion of the Straddle-type Monorail Vehicle running Tyre," Transactions of FAMENA, vol. 41, no. 1, pp. 99-112, 2017.

[4] W. Blythe and T. D. Day, "Single vehicle wet road loss of control; effects of tire tread depth and placement," SAE Technical Paper Series, 2002.

[5] V. Krishnappa, H. S. Matthews, and Y. Liu, "Data-driven analysis to support revised tire Tread Inspection Standards," Transportation Research Record: Journal of the Transportation Research Board, vol. 2673, no. 11, pp. 517–528, 2019.

[6] M. H. Bhamare and A. J. L. Khachane, "Quality Inspection of Tire using Deep Learning based Computer Vision," vol. 6, no. 11, 2019.

Authorized licensed use limited to: UNIV OF ENGINEERING AND TECHNOLOGY LAHORE. Downloaded on April 30,2024 at 18:02:57 UTC from IEEE Xplore. Restrictions apply.

[7] G. de León, L. G. Del Pizzo, L. Teti, A. Moro, F. Bianco, L. Fredianelli, and G. Licitra, “Evaluation of tyre/road noise and texture interaction on rubberised and conventional pavements using CPX and profiling measurements,” Road Materials and Pavement Design, vol. 21, no. sup1, 2020.

[8] X.-B. Wang, A.-J. Li, Q.-P. Ci, M. Shi, T.-L. Jing, and W.-Z. Zhao, "The study on tire tread depth measurement method based on machine vision," Advances in Mechanical Engineering, vol. 11, no. 4, 2019.

[9] M. Vagač, M. Povinský, and M. Melicherčík, "Obtaining tire tread model from its real world photo," in 2019 IEEE 15th International Scientific Conference on Informatics, 2019, pp. 000167-000170: IEEE.

[10] S.-Y. Huang, Y.-C. Chen, and J.-K. Wang, "Measurement of tire tread depth with image triangulation," in 2016 International Symposium on Computer, Consumer and Control (IS3C), 2016, pp. 303-306: IEEE.

[11] Y. Liu, S. Zhang, D. Li, J. Fan, and W. Liu, "An effective tread pattern image classification algorithm based on transfer learning," in Proceedings of the 3rd International Conference on Multimedia Systems and Signal Processing, 2018, pp. 51-55.

[12] E. Petrović, D. Ristić Durrant, M. Simonović, Ž. Ćojbašić, and V. Nikolić, “Vision-based inspection of Tyre Tread Depth,” Transactions of FAMENA, vol. 45, no. 3, pp. 19–28, 2021.

[13] A. Michalíková and M. Vagač, "A tire tread pattern detection based on fuzzy logic," in Flexible Query Answering Systems 2015: Springer, 2016, pp. 381-388.

[14] M. I. Pavel, S. M. Kamruzzaman, S. S. Hasan, and S. R. Sabuj, "An IoT based plant health monitoring system implementing image processing," in 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), 2019, pp. 299-303: IEEE.

[15] M. Azad, F. Khaled, and M. I. Pavel, “A novel approach to classify and convert 1D signal to 2D grayscale image implementing support vector machine and empirical mode decomposition algorithm.,” International Journal of Advanced Research, vol. 7, no. 1, pp. 328–335, 2019.

[16] J. Tang, A. Henderson, and P. Gardner, “Exploring AdaBoost and random forests machine learning approaches for infrared pathology on unbalanced data sets,” The Analyst, vol. 146, no. 19, pp. 5880–5891, 2021.

[17] M. Nabipour, P. Nayyeri, H. Jabani, S. S., and A. Mosavi, “Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis,” IEEE Access, vol. 8, pp. 150199–150212, 2020.

[18] X. Huang, Z. Li, Y. Jin, and W. Zhang, “Fair-AdaBoost: Extending AdaBoost method to achieve fair classification,” Expert Systems with Applications, vol. 202, p. 117240, 2022.

[19] T. Wisanwanichthan and M. Thammawichai, “A double-layered hybrid approach for network intrusion detection system using combined naive Bayes and SVM,” IEEE Access, vol. 9, pp. 138432–138450, 2021.

[20] C. Kim and T. Park, “Predicting determinants of lifelong learning intention using Gradient Boosting Machine (GBM) with grid search,” Sustainability, vol. 14, no. 9, p. 5256, 2022.

[21] M. Lee, J.-H. Lee, and D.-H. Kim, “Gender recognition using optimal gait feature based on recursive feature elimination in normal walking,” Expert Systems with Applications, vol. 189, p. 116040, 2022.

[22] M. I. Pavel, R. I. Rumi, F. Fairooz, S. Jahan, and M. A. Hossain, “Deep residual learning approach forplant disease recognition,” International Conference on Mobile Computing and Sustainable Informatics, pp. 511–521, 2020.

[23] D. A. Muhtasim, M. I. Pavel, and S. Y. Tan, “A patch-based CNN built on the VGG-16 architecture for real-time facial liveness detection,” Sustainability, vol. 14, no. 16, p. 10024, 2022.

Authorized licensed use limited to: UNIV OF ENGINEERING AND TECHNOLOGY LAHORE. Downloaded on April 30,2024 at 18:02:57 UTC from IEEE Xplore. Restrictions apply.