2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)

Driving Behavior Profiling and Prediction in KSA using Smart Phone Sensors and MLAs

Absar Abdul Rahman
Al-Yamamah University
Riyadh, Saudi Arabia

Waleed Saleem
Al-Yamamah University
Riyadh, Saudi Arabia

Dr. Vani Vasudevan Iyer
Al-Yamamah University
Riyadh, Saudi Arabia

***Abstract*—Driving behavior plays an important role in traffic safety and eco driving. In this paper, using low-cost smartphone sensors driving profiles (dataset) of various drivers were collected to predict aggressive behavior while driving in the Kingdom of Saudi Arabia (KSA). The prediction was performed using various Machine Learning Algorithms (MLAs) on the dataset with different tools which helped to identify the prediction model that outperforms. The predicted model can be further used in upcoming self-driving cars, prototypes and various driving applications in KSA to predict the aggressive nature of the driver and traffic.**

Keywords—*Driving Behavior, Smartphone, Aggressive, Sensors, Dataset, Machine Learning Algorithms, Profiling, Prediction*

I. Introduction

Saudi Arabia is an oil based derived economy, but for now have changed their directions to increase the economy. Eco driving cars are being introduced and soon Tesla will be in the Kingdom **[1]**. The country is reported as the most accident rated, including road rage which leads to the furious attitude towards driving. This work proposes a dataset collected on the roads of Riyadh using low-cost smartphone sensors which will be assessed using machine learning algorithms.

Driving is considered to be one of the activities in Saudi Arabia where young generation takes pride in and the extendibility of cars starting from cheapest to the more luxury brands on the road. Driving is a necessity in the Kingdom where every outing and household activity requires a car. The number of cars have increased significantly over the past decade.

All over the world researchers have been working on how to reduce road accidents using low-cost smartphone sensors and machine learning **[2]**. Detecting all the actions of a driver and the car is hard enough for a smartphones sensor to detect, due to non-fixed position of a smartphone it makes it difficult to get a proper dataset generated. The work proposed **[3]** states that detecting lane changes is quite sensitive as the sensors cannot detect slight changes.

In this paper, the approach is to collect a dataset to preprocess the data manually by using low-cost smartphone sensors and then to create a dataset where aggressive events and non-aggressive events can be generated. The idea is to make the machine learn of either the events are aggressive or not. Hence it will prove the efficiency of the dataset collected and to experiment the classifiers of machine learning where most used algorithms will be compared. The findings will deliver the algorithm best for the dataset, flaws of other models against the best one and the logic behind wrong predictions generated by the algorithms.

II. Related Work

Jiangpeng Dai **[4]** developed such a system that can notice driving under influence by only using the accelerometer of the smartphone. Their inspiration for outlining such a framework is the way that more often than not alcoholic driving goes unnoticed by the experts, which puts numerous individuals security in danger. They compressed alcoholic driving related practices from an examination done by the United States National Highway Traffic Safety Administration. There are two classes of behavioral signs which compare to a high likelihood of alcoholic driving. The primary classification is identified with path situating issues, for example, floating and swerving. The second classification is identified with speed control issues, for example, sudden increasing speed or sporadic braking. Both these classes of signals can be recognized by utilizing an accelerometer to delineate prompts into the parallel and longitudinal increasing velocities of a vehicle. The system is designed with four software components: a monitoring daemon module, calibration module, pattern matching module and an alert module.

Likewise, there was another study conducted in 2011 at the Virginia Polytechnic Institute and State University on automatic traffic accident detection and notification with smartphones **[5]**. Their whole idea generates automated detection of traffic accidents and traffic notifications which is updated using the smartphone and is able to detect using sensors. The study focuses on the smartphone which falls from a mobile holder and sends an SOS to authorities to react rapidly if an accident occurs. This study also uses machine learning algorithms to learn the patterns of the mobile falls and the intensity of a jerk which is detected by the smartphone sensors to declare an event as an emergency.

DriveSafe is an application based in Spain which detects driving behavior using machine learning. It detects the lane changes; records events generated and results a score of the driver's behavior **[6]**. Using DriveSafe, a dataset was collected of 500 minutes and then was processed using machine learning **[7]**. The application delivered the scores based on how normal, drowsy or aggressive the behavior was.

As per the region, only aggressive maneuvers will be assessed for this paper as the dataset will contain both aggressive and non-aggressive events for the algorithms to differentiate.

One of the latest research works was conducted by MIT researchers **[8]** which was published in 2017. Their methodology had a purpose of analyzing and profiling driving behavioral aspects. The duration of the driving data collected for their paper was of about 70 minutes. However, the main goal of the paper was to come up with the best machine learning algorithms. The results were compared among top algorithms and random forest was considered to be the best performing in 35 algorithms.

III. Dataset

The capital of Saudi Arabia, Riyadh is under going through a vibrant change. The city is being blessed with a total revamped local transportation system. Riyadh Metro is being constructed since 2013 **[9]**. The system includes buses, metro and local drivers for chauffeuring local families to the bus and metro stations. The state of the city to be under construction made the data collection quite difficult. The roads being constructed and then re-constructed, the non-levelling of asphalt on the surface has made the roads scatter.

There is no such dataset in the region which enables to study the machine learning algorithms. The region under goes the pressure of heavy traffic. The dataset is collected on the main roads (3+ lanes) which are the most utilized and maintained roads in the city. The selection of the roads was approached based on the regions traffic approach, people tend to travel via main roads instead of the roads which are signal enabled. The main roads also jam at peak timings. All around the Middle East, almost a similar set of patterns is followed to build roads and as the region has a great wide road network, the dataset can also be used in the other neighboring countries.

The data was collected in a Toyota Corolla 2015 using a mobile holder in the car at the right A/C vent. The driving in various manners was performed in order to achieve real-time events. The roads selected had an official speed of 120km/h, 100km/h and 80km/h respectively. Problems were faced due to the roads construction which also made the data collection tasks difficult. Falling of the smartphone due to sudden potholes on the road made the dataset a waste and then was dropped.

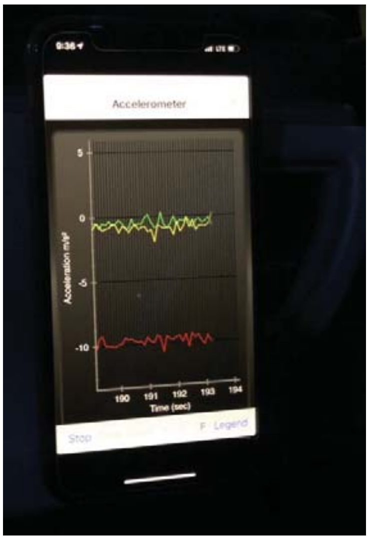

Fig. 1. Setup for Data Collection

Dataset was collected using the Sensor Kinetics **[10]** application which allows the sensors to record data in (x, y, z axis). The dataset was recorded at **50Hz**. The application enables various sensors to record but due to the limitation of the sensors only inertial sensor i.e. **Accelerometer** and **Gyroscope** were used.

Data collected from Accelerometer is used to detect sudden accelerations and brakes as the y-axis from the tri axis increasing up-to 0.5 is acceleration at x-axis and decreased up-to -0.5 is braking. Retrieving the data is a huge effort based on the road structures of the capital, Riyadh due to the metro construction. Numbers of bumps and potholes affected the recordings which made instant changes in the data due to that the datasets were discarded in this scenario. Gyroscope alone is capable to detect the right and left turns. When taking a right, the z-axis will go up-to -0.5 at x-axis and up-to 0.5 if it's a left in basic formulation. Right's and left's can also be detected using accelerometer graphs, but gyroscope provides exact readings. Dataset was normalized to achieve (0 to 1) range using Min-Max Normalization. The normalization was done in order to make the dataset easy on the algorithms to achieve great scores.

n = (Original Value – Old Minimum) / (Old Maximum – Old Minimum)

The z-axis of tri axis in accelerometer was normalized using the formula which indicated the acceleration and braking from the sensor. Gyroscope's y-axis uniquely identifies the right and left turns and was also normalized. The normalization was set to change the values from 0 to 1 where 0 being the brake/left and 1 being the acceleration and left. The events are then shuffled according to aggressive and non-aggressive events. The collected raw data was then filtered to opt out the events which were aggressive and non-aggressive. The dataset had four numeric and one nominal attributes.

A R	A L	A A	A B	Aggressive
0.38495138	0.28695479	0.58107334	0.58602546	yes
0.41140462	0.28108345	0.64074785	0	yes
0.41140462	0.31670245	0.64074785	0.33675951	yes
0.41695937	0.31712085	0.96061868	0.69591296	yes
0.68645314	0.33053027	0.58549333	1	yes
0.46936866	0.25594507	0.84609162	0.99975519	yes
0.46414003	0.25008059	0.51858587	0.64023477	yes
0.51084487	0.26893609	0.32710462	0.61600602	yes
0.46871642	0.28443752	0.85654032	0.70802734	yes
0.45793838	0.33178548	0.20413954	0.98433712	yes
0.48994103	0.22116974	0.61502932	0.69591296	yes
0.47491807	0.30287462	0.46292938	0.69591296	yes
0.46904254	0.34184083	0.55736382	0.66727861	yes
0.49549578	0.3070655	0.80188834	0.50636333	yes
0.45532406	0.25888074	0.61985128	0.81522293	yes
0.44716034	0.29449288	0.74241523	0.74987798	yes
0.41515234	0.27438217	0.20594714	0.73274623	yes
0.46512374	0.27396377	1	0.58602546	yes
0.49810475	0.3460317	0.35443019	0	yes

Fig. 2. Final Dataset

The total number of aggressive events add up to 72 from the collected 350 minutes. The extracted raw data was studied thoroughly to create a simplified idea of detecting 4 maneuvers; right, left, acceleration and braking.

TABLE I. Aggressive Events Retrieved

Aggressive Right	Aggressive Left	Aggressive Acceleration	Aggressive Braking
13	3	22	32

The collected and preprocessed datasets are available on GitHub **[11]**.

IV. Proposed Work

3.1 Machine Learning Algorithms (MLA)

3.1.1 Bayesian Networks

Bayesian Networks represents a specific type of graphical model which are used for a limited data collected. The graphs which are structured to represent information. They can model sequences of variables. Bayesian networks use 3 main inference tasks: inferring unobserved variables, parameter learning and structure learning **[12]**. It understands a problem well and forecasts the consequences.

Bayesian Networks, also known as *belief networks* (or Bayes Net for short), belong to the family of probabilistic Graphical Models (GM). These graphical structures are used to represent knowledge about an uncertain domain. In particular, each node in the graph represents a random variable, while the edges between the nodes represent probabilistic dependencies among the corresponding random variables. These conditional dependencies in the graph are often estimated by using known statistical and computational methods. GM with undirected edges are called Markov Networks. These networks provide a simple definition of independence between any two distinct nodes based on the concept of a Markov Blanket. Markov Networks are popular in fields such as computer science and statistical physics.

3.1.2 Trees

Trees classification method is an approach to the problem of learning from a set of independent instances leading naturally to a style of representation. This classification works on a divide and conquer approach but is quite vague in terms of its operations. Nodes of a tree either test a particular attribute or compare the attributes with each other. Classification methods of a tree or a probability distribution is applied on the leaf nodes which implemented on all the instances of a dataset, reaches the leaf. The selection of the nodes and leaf nodes depends on the algorithm and its identification but will result out the instances anyway.

If the dataset being trained contains numeric data, the nodes determine to set a constant and then apply two-way splits or more based on the dataset. Three-way split creates more possibilities to the algorithm but on other hand a huge unpruned tree will be resulted. If the dataset is considered to be nominal, the number of children of a leaf comes out to be possible instances of the attributes. If only one branch is generated of the outcomes leaf, the instances will not be tested again in the flow.

3.1.3 Multi-Layer Perceptron (MLP)

An artificial neural network deep technique composed of more than one perceptron's. They operate as input layer and output layer, where the input layer is a signal and the output signal is a decision based on the input, between the input layer and output layer are many hidden layers which play the essential part in the MLP technique **[13]**. It works in back and forth motion. There are two passes, forward pass and back pass where forward pass sends the signal which moves through the hidden layers whereas in the back pass the information flowed via signals is solved using chain rules of calculus to come out with a gradient or landscape of error results.

3.2 Approach

The work proposed by the MIT team focuses on the collection of raw data using low-cost smartphones and then working on certain techniques to generate a score. In this paper we made an attempt to collect data and assess it using machine learning, but the dataset is a specialty to this. The dataset will help the upcoming people to study more about the regions traffic style and behavior. The raw dataset extracted by the smartphone's sensor is normalized. The normalized dataset will go through the machine learning algorithms and then be assessed. The dataset will contain aggressive and non-aggressive events. These events will enter the following machine learning algorithms using Weka **[14]**:

- Bayesian Network
  1. Bayes Net
  2. NaiveBayes
- Trees
  1. J48
  2. Random Forest
- Multi-Layer Perceptron

These algorithms will be assessed using the dataset that which algorithm is more effective. The results will be tested to check the inferences of the algorithms. The algorithm which will support the dataset and predict the greatest number of instances to be true. The scores will be comparatively studied to check the efficiency and predictability of each algorithm that how it connects to the dataset. The dataset will be tested using 10 folds cross-validation.

Bayesian Networks predicts using continuous input of numeric attributes as the network within produces outcome using the probability estimation depending on the instances. J48 being a C4.5 based classifier will be studied deeply after the tree being pruned at 25% of fixed rate. The functions classifier of Weka, MLP has a limited setting which can be tuned in accordance to the dataset being fed. However, now the market has new libraries which allow extended ability to the algorithm.

In the experiments conducted, the approach was to test the dataset which was preprocessed differently and was collected in a different region with correspondence to Saudi Arabia's Vision 2030 **[15]**. The vision enables the growth of technology in the country and by the help of this paper, upcoming projects or applications in the industry can follow a similar approach to implement machine learning which is great for the market and a first contribution to a step forward.

V. Implementation

Using Weka, the dataset was input to perform classifications. The dataset contains --- aggressive events and --- non-aggressive events. The tests were run on 10 folds cross-fold validation as it learns from the first 9 folds and predicts on the last fold **[16]**. Confusion Matrix or a contingency table is an output of the model which represents four or more outputs depending on the input to be. For our case the matrix is 2x2 which represents four outputs as True Positive, False Positive, False Negative and True Negative. True Positive displays the number of instances which are truly predicted by the model, false positive are the instances which are classified as positive where as they are not, false negative are the instances which are not true but are classified as positive and true negatives are the values which are actually negative.

Bayesian Network with global search algorithms i.e. K2 and Repeated Hill Climber were set to test the dataset. The model was also tested using the same local searching algorithms but had no effect on the results. Global searching algorithms implement the network on all the folds but local focuses on each and every instance in every fold. Due to the nature of the dataset to be numeric, it relieves the efforts of the algorithms. In Bayesians case, the tests predicted 98% of the instances to be correct on all the search algorithms used. The BayesNet using K2 learning algorithm returns 927 correctly classified instances out of 944. The mean absolute error is 0.0258% which means while the dataset was being trained, the model understood the input and classified properly the instances and hence the metrics calculated 98.1992% of instances to be validated correctly. The confusion matrix denotes 13 false positive and 4 false negative. BayesNet using Repeated Hill Climber learning algorithm returns 927 correctly classified instances out of 944. The mean absolute error is 0.0258% which means while the dataset was being trained, the model understood the input and classified properly the instances and hence the metrics calculated 98.1992% of instances to be validated correctly. The confusion matrix denotes 13 false positive and 4 false negative.

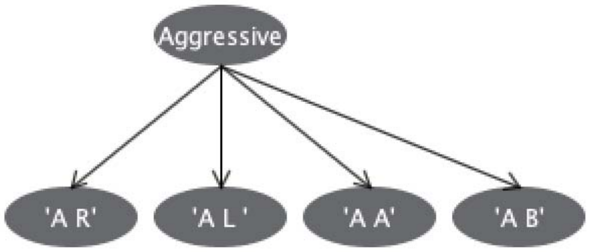

Fig. 3. BayesNet Graph

NaiveBayes being another branch of Bayesian Networks predicted 95% of the instances to be classified truly using its default settings. The results are high due to the nature of the algorithm, it works flawlessly with a small numeric based dataset. The NaiveBayes using its default settings returns 891 correctly classified instances out of 944. The mean absolute error is 0.0864% which means while the dataset was being trained, the model understood the input and classified properly the instances and hence the metrics calculated 94.3956% of instances to be validated correctly. The confusion matrix denotes 21 false negatives and 32 false positive.

Trees classification in this case is operating by comparing each other attributes. The algorithms present in Weka follow the basic approach to create a tree by selecting the most identified attribute by the algorithm and build the tree. Major differences in the tree algorithms are basically identified on how it is pruned and the handling of a multivariate tree which means a tree has more than one attribute. J48 being the most workhorse algorithm developed using C4.5 followed by the updated C5.0 is the only algorithm which prunes a tree in accordance to the attributes. Random Forest on the other hand is build out of another algorithm, Decision Tree. Multiple decision trees come out to build a Random Forest. Decision Tree takes more processing power and time than any classifier. Random Forest doesn't view a tree as it is unpruned and cannot be visualized.

In the first test of Trees, the sub algorithm selected is J48 with the learning 1, confidence factor as 0.25 and 1 for generating random seeds. The J48 using its settings returns 885 correctly classified instances out of 944. The mean absolute error is 0.0873% which means while the dataset was being trained, the model understood the input and classified properly the instances and hence the metrics calculated 93.75% of instances to be validated correctly. The confusion matrix denotes 34 false negatives and 25 false positive.

J48 algorithm resulted 93.75% of the instances to be correctly classified. The pruned tree maintains aggressive left (A L) as the most important attribute. The tree goes down in a way that the instance (<= 0.40017) are considered to be accelerated and (> 0.40017) to be braking. Similarly, the attributes down the tree compare with each other based on the selected constant and then break downs to the last leaf.

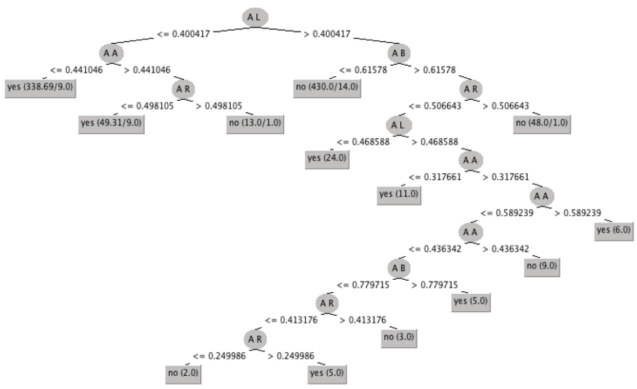

Fig. 4. Pruned J48 Tree

In the second test of Trees, the sub algorithm selected is Random Forest with the learning set to 1 for generating random seeds. The Random Forest using its default settings returns 902 correctly classified instances out of 944. The mean absolute error is 0.075% which means while the dataset was being trained, the model trained had classified properly the instances and hence the metrics calculated 95.5508% of instances to be validated correctly. The confusion matrix denotes 23 false negatives and 19 false positive

The last classifier implemented was MLP, the algorithm is expected to work great with big data and nominal datasets. MLP was run using the Sigmoid activation function which activates a non-linear function to learn the data. The number of hidden nodes were set to 1 and 20 in separate experiments. The learning rate was set to 0.3, the experiments conducted were tested with the upmost models but in result had no change. The MLP using its settings returns 888 correctly classified instances out of 944. The mean absolute error is 0.0666% which means while the dataset was being trained, the model understood the input and classified properly the instances and hence the metrics calculated 94.0678% of instances to be validated correctly. The confusion matrix denotes 27 false negatives and 29 false positive.

VI. Inferences

In assessed models, the evaluation is based on four bases. Kappa Statistic, Mean Absolute Error, Root Mean Squared Error and Confusion Matrix. Kappa Statistic measures the prediction using the true/yes class, any value printed by the model which is near to 1 represents highly achieved kappa statistic which means the accuracy is great. Mean Absolute Error calculates the average error linearly which means that it takes difference between absolute values and predicted. All the differences are weighted equally in the average. Root Mean Squared Error calculates the average error quadratically which means it is most useful for largely error expected. The RMSE is calculated by taking the difference between absolute values and predicted values are each squared following the averaged instances of the dataset and then the square root of the average is printed. If the RMSE is equal or larger to MAE it means that the errors are of the same magnitude **[17]**.

TABLE II. RESULTS

Classifiers	Learning Algorithms	Kappa Statistic	Mean Absolute Error	Root Mean Squared Error	Accuracy
BayesNet	K2	0.9638	0.0258	0.119	0.982
BayesNet	Repeated Hill Climber	0.9638	0.0258	0.119	0.982
NaiveBayes	N/A	0.8869	0.0864	0.2121	0.944
Trees	J48	0.8745	0.0873	0.2417	0.938
Trees	Random Forest	0.9106	0.075	0.1905	0.956
Multi-Layer Perceptron	N/A	0.8807	0.0666	0.2211	0.941

After all the algorithms and models were evaluated, Bayesian Algorithm was far most the best algorithm for this dataset and model. Due to the Bayesian Networks statically and graphically calculated nodes allowed each of the input from the dataset to be learned and the global search algorithms made sure that all values were learned in the folds. Whereas Trees classification divide and conquer approach made Random Forest at the best from the J48 classifier.

BayesNet being a branch of Bayesian Network which works statistically and using plot graphs. BayesNet's different search algorithm K2 and Repeated Hill Climber were opted due to the optimal graph topology. Cross validation option divides data equally into selected separate folds. The result is 927 correctly classified instances (98.199%) and 17 incorrectly classified which end being the same for both the search algorithms. The mean absolute error is 0.0258. The result being the same might be due to the algorithms being restricted in K2 and not in Repeated Hill Climber.

Trees work using divide and conquer approach. Using the Trees classification with different algorithm such as J48 and Random Forest comes up with a difference of producing pruned and de-pruned trees or the algorithm itself following the trees. The result is 885 correctly classified instances (93.75%) and 59 incorrectly classified for the J48 classifier while the result for Random Forest is 902 correctly classified instances (95.55%) and 42 incorrectly classified. The mean absolute error for J48 is 0.0873 and Random Forest is 0.075

NaiveBayes being the branch of Bayesian Network follow the directed acrylic graph method to predict the variances while Multi-Layer Perceptron uses the nodes and hidden layers technique to predict which falls under Artificial Neural Network (ANN). The result is 891 correctly classified instances (94.38%) and 53 incorrectly classified for NaiveBayes while the result for MLP is 888 correctly classified instances (94.06%) and 56 incorrectly classified. The mean absolute error for J48 is 0.864 and Random Forest is 0.0666.

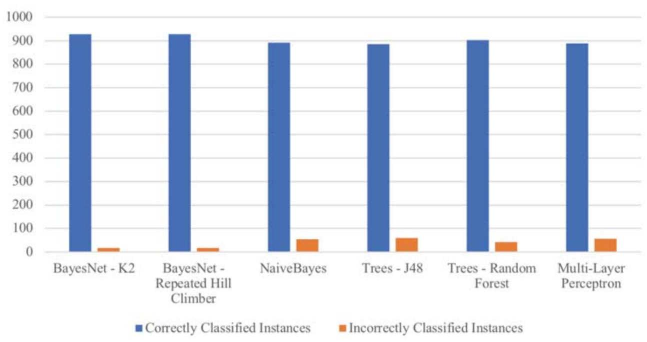

Fig. 5. Classified Instances

The outcomes of these inferences are classified; however, the results were tested using another small dataset and the predictions were true based on the implemented and trained dataset.

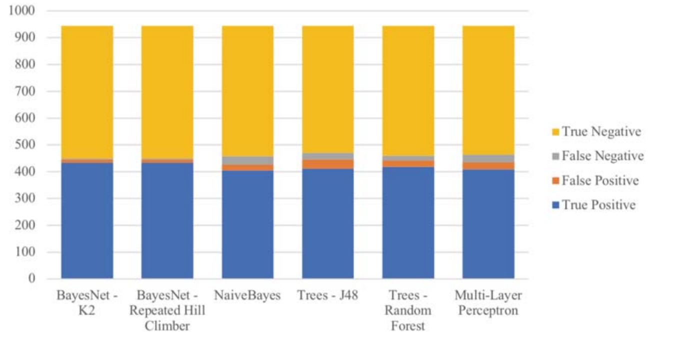

Fig. 6. Predictions by Confusion Matrix

The confusion matrix adds up to total instances and the classified errors predicted. Outcomes of the confusion matrix are supposedly based on the tests conducted. Another dataset was recorded in order to prove the reasons behind the incorrectly classified instacnes i.e. True Negative and False Positives. The values were not classified as during the data collection process, the roads were not good enough to make the sensors not record any sudden movements or jerks which was caused and those values created specific issues. The values of the recorded dataset varied when a pothole was detected which generated similar values as of aggressive acceleration and was difficult to remove.

However, the dataset being normalized and tested with different settings up to an extent made it clear that the results are accuracte. J48 resulted a bit low than the Decision Trees but due to the small size of dataset for the corresponding algorithm, the result varied with other models. MLP known as a good model for any type of attributes held up pretty well.

Overall the algorithms which were conducted in this test were close to over-fitting, but the outcomes were true as tested with the originated dataset. Bayesian Network was the 1st and 2nd classifier with holding 98% accuracy with two searching algorithms, K2 and Repeated Hill Climber were distinguished as a great methodology for our dataset. The best model using Bayesian Network are fed up with numeric datasets and others for nominal type **[18]**. The 3rd stand up classifier was from the classification of Trees – Random Forest with 95.55% accuracy, followed by the NaiveBayes 4th classifier under the branch of Bayesian Network with 94.3856% accuracy. Artificial Neural Network's branch, Multi-Layer Perceptron (MLP) held the 4th position with 94.0678% of accuracy. J48 of Trees classifiers held up at last with 93.75% of accuracy.

VII. Conclusion

The dataset which was collected on the streets of the capital is now a suitable and small dataset which can be used to make any reference regarding to driving in Saudi Arabia and can help many upcoming ideas. The research provided a great initiative for the upcoming researchers in the country and making a tested dataset available for new self-car prototypes which can initially learn about a different style of traffic. The algorithms executed on the dataset and the results point to the knowledge of numeric datasets which are easier to learn for a model than a nominal dataset, Bayesian Algorithms being the best in this case due to its statistical execution.

MLP being the vastly used machine learning algorithm is a better approach for our dataset in a long run. Researchers opt to work on deep algorithms using MLP than to follow old algorithms but MLP being the fastest and intelligent algorithm has its own place.

The dataset was made very easy for the algorithms to make them understand as all the algorithms were more than 90% accurate and not being less. This shows that the algorithms didn't either under-fit or over-fit. The dataset was trained and was double-checked to learn that is the algorithm being efficient or not. The tested instances were relatively falling under the categories of confusion matrix in Weka when the algorithm were tested to check the predicted outcome. In this paper, it was concluded that the classification results obtained through the various data mining algorithms were sufficient, and they were satisfactory to prove the strength of data mining when used to predict driving performance in the datasets.

Acknowledgment

We wish to thank the management of Al-Yamamah University **[19]** for providing sponsorship to present our paper in the conference.

References

[1] Tesla in the Gulf *(http://www.arabnews.com/node/1119926/motoring)*

[2] White, J., Thompson, C., Turner, H. et al. Mobile Netw Appl (2011) 16: 285. https://doi.org/10.1007/s11036-011-0304-8

[3] Liu X, Mei H, Lu H, Kuang H, Ma X. A Vehicle Steering Recognition System Based on Low-Cost Smartphone Sensors. *Sensors (Basel)*. 2017;17(3):633. Published 2017 Mar 20. doi:10.3390/s17030633

[4] J. Dai, J. Teng, X. Bai, Z. Shen and D. Xuan, "Mobile phone based drunk driving detection," *2010 4th International Conference on Pervasive Computing Technologies for Healthcare*, Munich, 2010, pp. 1-8. https://doi.org/10.4108/ICST.PERVASIVEHEALTH2010.8901

[5] White, J., Thompson, C., Turner, H. et al. Mobile Netw Appl (2011) 16: 285. https://doi.org/10.1007/s11036-011-0304-8

[6] L. M. Bergasa, D. Almería, J. Almazán, J. J. Yebes and R. Arroyo,"DriveSafe: an App for Alerting Inattentive Drivers and Scoring Driving Behaviors", in IEEE Intelligent Vehicles Symposium

[7] E. Romera, L. M. Bergasa and R. Arroyo, "Need data for driver behaviour analysis? Presenting the public UAH-DriveSet," *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, 2016, pp. 387-392. doi: 10.1109/ITSC.2016.7795584

[8] Ferreira J Júnior, Carvalho E, Ferreira BV, de Souza C, Suhara Y, Pentland A, et al. (2017) Driver behavior profiling: An investigation with different smartphone sensors and machine learning. PLoS ONE 12(4): e0174959. https://doi.org/10.1371/journal.pone.0174959

[9] Riyadh Metro *(http://riyadhmetro.sa/en/)*

[10] Sensor Kinetics – Roto View *(https://www.rotoview.com/sensor_kinetics.htm)*

[11] Data Collection of 350 Minutes Drivers Dataset collected in Riyadh *(https://github.com/absarar/ruhdriversdataset)*

[12] An Introduction to Bayesian Network Theory And Usage by Todd A. Stephenson *(http://publications.idiap.ch/index.php/publications/show/784)*

[13] Why MultiLayer Perceptron/Neural Network? *(http://courses.media.mit.edu/2006fall/mas622j/Projects/manu-rita-MAS_Proj/MLP.pdf)*

[14] Eibe Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016. https://www.cs.waikato.ac.nz/ml/weka

[15] Saudi Arabia's Vision 2030 *(https://vision2030.gov.sa/en)*

[16] Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. OpenML: networked science in machine learning. SIGKDD Explorations 15(2), pp 49-60, 2013. https://www.openml.org/a/estimation-procedures/1

[17] Weka Introduction *(https://spu.fem.uniag.sk/cvicenia/ksov/fuskova-ulicna/Data%20mining/cvicenie8_classification/Cv8_classification_some%20interpretations.pdf)*

[18] Moran, Stuart; He, Yulan and Liu, Kecheng (2009). Choosing the best Bayesian classifier: an empirical study. *IAENG International Journal of Computer Science*, 36(4) pp. 322–331. http://oro.open.ac.uk/23788/

[19] Al-Yamamah University *(https://www.yu.edu.sa)*