Contents

1. Summary:

In order to predict driving behaviors that are aggressive, this paper explores the use of deep learning techniques using accelerometer and gyroscope data from cellphones. It explores crucial topics including developing an accurate prediction model and optimizing its performance using hyperparameters. The research highlights the significant efficiency of deep learning algorithms in identifying risky driving behaviors and clarifies the vital role that hyperparameter adjustment plays in enhancing model performance. In the end, the results highlight how deep learning methods may improve driving safety and how important it is to carefully modify parameters to get the best possible model performance.

2. Introduction:

Driving aggressively is a serious threat to road safety as it causes a significant number of collisions and the deaths that follow. Thus, it becomes critical to quickly identify and anticipate such conduct, providing a powerful means of preventing accidents and saving lives. Using the wealth of accelerometer and gyroscope data that is easily available from modern cellphones, this study investigates deep learning techniques to predict aggressive driving behavior. By utilizing these widely distributed sensors, we want to develop a predictive model that can identify warning indicators of aggressive driving, such as instances of speeding excessively, sudden braking, and sharp changes in direction that are indicative of risky driving behavior.

This project is motivated by its critical role in bolstering efforts related to road safety and reducing the number of car accidents. Our goal is to provide powerful tools for proactive intervention and accident mitigation measures to transportation authorities, car manufacturers, and road safety advocates by fusing state-of-the-art deep learning algorithms with the abundance of data derived from smartphone sensors. This research project is a collaborative effort between public welfare and technical innovation, with the ultimate goal of creating a safer and more secure driving environment for all users of the road. We aim to fully utilize deep learning for aggressive driving prediction by carefully adjusting parameters and conducting thorough analyses of predictive modeling approaches. This will help us steer toward a future with reduced road accidents and increased road safety.

3. Current Research:

Deep learning approaches have attracted a lot of interest lately in a variety of fields and have proven to be quite effective in a wide range of applications, such as the prediction of driver behavior. Hafidi (2019) performed a groundbreaking study that utilized smartphone sensor data to analyze driver behavior. The study specifically focused on instances of aggressive driving. By utilizing convolutional neural networks (CNNs), the research showed impressive precision in anticipating instances of aggressive driving, highlighting the significant potential of deep learning techniques to improve road safety.

Similar to this, Lokaman (2022) made a significant contribution to this developing field with their ground-breaking work on the onboard sensor-based real-time identification of aggressive driving behaviors. The researchers demonstrated the model's capacity to quickly and effectively identify instances of aggressive driving using a deep learning framework, providing a proactive way to intervene and reduce possible road risks. These results support

the effectiveness of deep learning methods in predicting driving behavior and highlight their critical significance in enhancing driving safety precautions.

The foundation of deep learning's application to driver behavior prediction is the accumulation of data from sensors integrated into contemporary automobiles. Specifically, gyroscope and accelerometer data are crucial training inputs for predictive models that allow for the identification of minute patterns suggestive of aggressive driving behaviors. Researchers have extracted valuable insights from these data streams by using advanced deep learning architectures, such CNNs and recurrent neural networks (RNNs). This has made it easier to identify and anticipate aggressive driving episodes in a timely manner. There is a ton of promise for practical applications when deep learning methods are included into driver behavior prediction systems. The insights gained by deep learning models can spur revolutionary advances in the field of road safety, from improving the capabilities of onboard driver aid systems to guiding the construction of safer automotive technology. Transportation authorities and automakers may create proactive measures to reduce the hazards connected with aggressive driving by utilizing deep learning's predictive capacity. This will ultimately create a more secure and safe transportation environment for all users of the roads.

A number of obstacles still stand in the way of the practical use of deep learning-based driver behavior prediction systems, notwithstanding the encouraging developments in this area. The most important of these difficulties is the requirement for solid and varied datasets that cover a broad spectrum of driving situations and circumstances. Furthermore, there is also considerable worry about the interpretability of deep learning models, especially in safety-critical applications where model conclusions need to be clear and understandable to stakeholders. Given these difficulties, future studies should focus on resolving the shortcomings of the current deep learning models while investigating novel strategies to improve their resilience and interpretability. The advancement and implementation of deep learning-based driver behavior prediction systems would not be possible without cooperation between industry, academia, and regulatory agencies. This will usher in a new era of vehicle innovation and road safety. The goal of safer roads and fewer traffic accidents may become a reality with coordinated efforts and multidisciplinary collaboration.

The effectiveness of deep learning techniques in predicting driver behavior has been highlighted by recent research projects, especially when it comes to detecting aggressive driving. Researchers have made great progress in improving driving safety and reducing the dangers related to aggressive driving behaviors by utilizing deep learning architectures and smartphone sensor data. To solve current issues and fully utilize deep learning for driving behavior prediction, additional research is necessary. This will open the door to a more secure and safe transportation environment.

4. Data Collection / Model Development:

Careful attention to both the data collecting and model building procedures is crucial in the quest of creating an efficient prediction model for identifying aggressive driving behavior. In this work, we utilize the motion.csv dataset, which we downloaded from Kaggle and contains smartphone accelerometer and gyroscope data. Critical characteristics like AccX, AccY, AccZ, GyroX, GyroY, and GyroZ are included in these data streams, providing information on the vehicle's acceleration and gyroscope readings along different axes. A crucial target variable, "Class," which indicates the kind of driving behavior displayed, is also included in the dataset.

It might indicate anything from normal driving movements marked as "NORMAL" to potentially dangerous activities categorized as "AGGRESSIVE." The addition of a "Timestamp" feature to the dataset enhances it by enabling temporal reference and guaranteeing the integrity of time-sensitive studies.

Given the intrinsic complexity of driving behavior prediction problems, the use of deep learning approaches in model creation is a prudent strategy. To do this, we made use of a Sequential model architecture, which is a distinguishing feature of deep learning paradigms and is distinguished by the sequential ordering of layers. We designed a number of Dense layers in this architecture, each of which was equipped with Rectified Linear Unit (ReLU) activation functions. Rectified Linear Units are a type of nonlinear activation function that is useful for adding nonlinearity and enabling the model to learn intricate patterns from the input. Dense layers are channels via which information is propagated, enabling input data to be transformed into meaningful representations that support predictive inference. The last output layer is positioned above this hierarchical configuration of Dense layers and is identified by a sigmoid activation function. This crucial layer is the channel by which the model generates binary predictions, and the sigmoid activation function provides the ability to interpret the results probabilistically. The sigmoid activation function provides the model with probabilistic predictions by restricting output values to the [0, 1] interval. This allows for more nuanced understanding of the probability that observed actions would be categorized as "AGGRESSIVE" or "NORMAL."

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
AccX	AccY	AccZ	GyroX	GyroY	GyroZ	Class	Timestamp							
0	0	0	0.059407	-0.17471	0.101938	NORMAL	3581629							
-1.62486	-1.08249	-0.20418	-0.02856	0.051313	0.135536	NORMAL	3581630							
-0.59466	-0.12241	0.220502	-0.01939	-0.02932	0.087888	NORMAL	3581630							
0.738478	-0.22846	0.667732	0.069791	-0.02993	0.054902	NORMAL	3581631							
0.101741	0.777568	-0.06673	0.030696	-0.00367	0.054902	NORMAL	3581631							
0.15847	0.345891	0.355274	0.021533	0.115454	0.014584	NORMAL	3581632							
0.078171	-0.34918	0.270652	0.034361	-0.03054	0.050625	NORMAL	3581632							
-0.66275	-0.46278	-0.0998	0.008705	-0.00977	0.04696	NORMAL	3581633							
0.467318	-0.23464	0.134139	-0.04872	-0.02077	-0.03001	NORMAL	3581633							
-0.1719	-0.40889	0.414653	0.009316	0.058032	-0.01962	NORMAL	3581634							
0.852679	-0.41678	0.491008	-0.00107	0.042761	-0.07216	NORMAL	3581634							
0.308851	0.239022	1.517034	-0.02917	0.159436	-0.07277	NORMAL	3581635							
-0.40298	-0.04039	-1.04802	-0.01695	0.004276	-0.04589	NORMAL	3581636							
-0.60461	0.840231	-0.84475	0.020311	0.005498	-0.07827	NORMAL	3581636							
0.163595	0.475107	0.006773	0.00504	-0.01344	-0.06116	NORMAL	3581637							
-0.25059	1.722196	0.529089	-0.00229	0.029932	-0.00557	NORMAL	3581637							
0.798495	0.182108	0.20606	-0.01145	0.017715	-0.0856	NORMAL	3581638							
-0.29117	1.694341	1.475946	0.007483	0.087965	-0.08987	NORMAL	3581638							
1.180107	1.619935	1.054108	-0.02306	0.032376	-0.01352	NORMAL	3581639							
-1.26628	-1.41811	0.402697	-0.01145	0.001833	-0.02818	NORMAL	3581639							
-1.94127	-1.33331	-0.80128	0.008094	0	-0.01168	NORMAL	3581640							
-0.16774	-1.09272	-0.44389	-0.01573	0.048258	0.001756	NORMAL	3581640							
0.099909	-0.90457	-0.58918	0.024587	-0.03115	-0.01535	NORMAL	3581641							
-0.23394	-0.09211	-0.21145	-0.0029	-0.01527	0.010308	NORMAL	3581641							
-0.74943	-0.33967	-0.79016	-0.01512	0.013439	0.023747	NORMAL	3581642							
-0.4812	0.726062	-0.46177	0.012981	-0.01466	0.001145	NORMAL	3581643							

During the compilation stage, selecting the right optimizer and loss function is crucial to the model's effectiveness. In this case, we choose binary cross-entropy loss, which measures the discrepancy between expected and actual class labels and is a good option for binary classification problems. Simultaneously, the model is optimized by the Adam optimizer, which is recognized for its fast convergence characteristics and adaptive learning rate mechanism. It directs the model to the best possible parameter configuration that minimizes the selected loss function. The core of our predictive modeling strategy is the combination of four components: a carefully selected dataset, an advanced model architecture, and a wise

selection of optimizer and loss function. By combining these elements in a harmonic way, we want to develop a predictive model that can recognize minute differences in driving behavior and provide useful information that is essential for improving road safety initiatives.

The suggested strategy is promising, but it is not without difficulties and things to think about. Specifically, the dataset's validity and representativeness demand close examination, which calls for careful data pretreatment and validation methods to reduce biases and guarantee the stability of the resulting model. Furthermore, empirical validation via rigorous experimentation and performance evaluation against benchmark datasets and alternative approaches is required to confirm the effectiveness of the selected model architecture and hyperparameters. Our attempt to forecast aggressive driving behavior is based on the combination of advanced data gathering techniques and ethical model construction procedures. We hope to provide transportation stakeholders with a powerful toolkit for proactive intervention and accident mitigation strategies by carefully considering model configuration and applying deep learning techniques sparingly. This will help to create a safer and more secure driving environment for all road users.

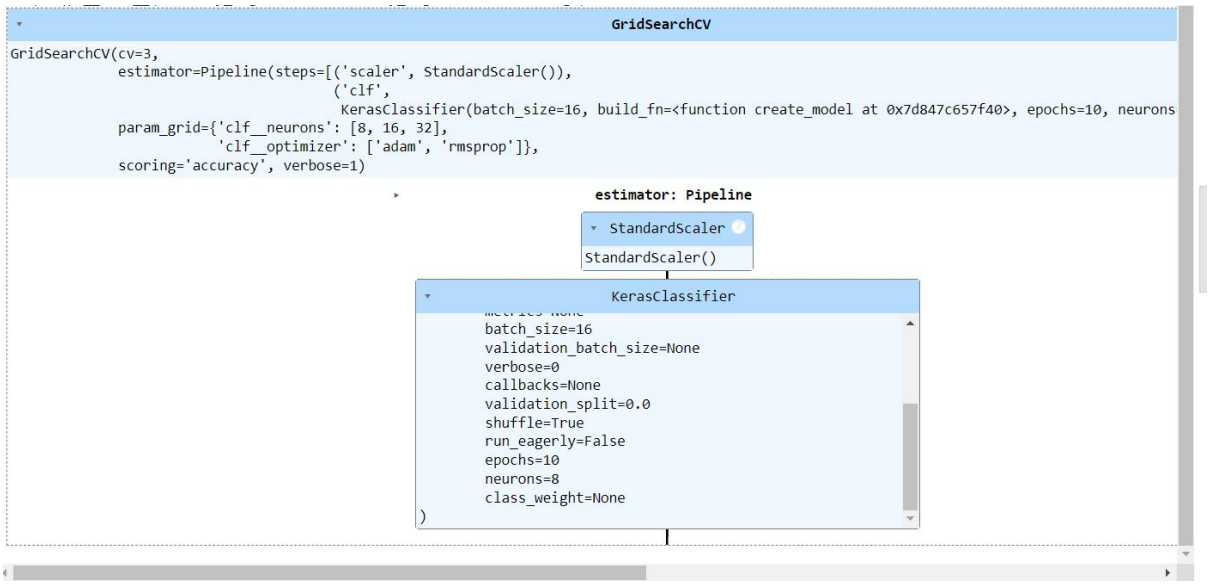
5. Analysis:

The purpose of the analysis done on the created predictive model was to see how well it could anticipate aggressive driving behavior, using hyperparameter tuning strategies to maximize performance and offer more subtle insights into the behavior of the model. One essential component of developing a machine learning model is tuning the hyperparameters. This involves exploring different configurations of the hyperparameters systematically in order to find the best set that optimizes prediction performance.

We performed hyperparameter tweaking in our investigation using the GridSearchCV technique from the scikit-learn toolkit. With the use of cross-validation, this approach allows for a thorough search throughout a given hyperparameter grid, identifying the ideal configuration by assessing every combination. In particular, we looked at hyperparameters—important variables that significantly affect the behavior and performance of the model—like the quantity of neurons in the Dense layers of the model and the optimizer selection. The first step in the hyperparameter tuning procedure was to build a parameter grid with various values for the hyperparameters that were being considered. To enable a thorough investigation of the model's architectural complexity, we, for example, provided a range of feasible options for the number of neurons, ranging from lesser values like 4 or 8 to bigger values like 16 or 32. In a similar vein, we have incorporated a range of optimizers, including Adam and RMSprop, which are recognized for their unique optimization characteristics and possible influence on model convergence and performance.

After defining the parameter grid, we created an instance of GridSearchCV, set it up to use an appropriate evaluation measure (accuracy, precision, or F1-score), and used a cross-validation technique to reliably estimate model performance across several data splits. After that, the GridSearchCV object methodically examined the parameter grid, fitting the model with every possible combination of hyperparameters and assessing its effectiveness using cross-validation. The GridSearchCV object performed several iterations of fitting the model with various hyperparameter configurations and evaluating their performance using the designated evaluation measure during the hyperparameter tuning phase. We were able to determine the ideal set of hyperparameters that produced the best results on the validation data through this

iterative method. After adjusting the hyperparameters to our satisfaction, we were able to use the GridSearchCV object to determine the optimal hyperparameter configuration. The hyperparameter values that optimized the model's performance, as shown by the selected evaluation metric, were contained in this ideal configuration. The neural network design with eight neurons in the Dense layers and the Adam optimizer were the optimal hyperparameters in our scenario.



After determining the optimal hyperparameters, we tested the improved model's performance using a hold-out test dataset. This entailed evaluating the model's predicted performance on data that had not yet been observed and fitting the model with all of the training data while utilizing the best hyperparameters. The test dataset's evaluation metrics allowed for a thorough examination of the model's generalization capacity and practical applicability. A comprehensive picture of the model's predictive power was provided by the evaluation metrics that were calculated on the test dataset. These metrics included accuracy, precision, recall, and F1-score, among other elements of model performance. Additional information on the distribution of classification outcomes and possible areas for improvement was provided by the confusion matrix, a tabular representation of the model's predictions in comparison to the ground truth labels. Upon analyzing the assessment metrics and confusion matrix, a number of significant revelations concerning the model's ability to forecast aggressive driving behavior surfaced. First, the test accuracy gave an overall indication of the predictive power of the model by quantifying the percentage of properly categorized occurrences. Furthermore, the model's accuracy in identifying instances of aggressive driving behavior while limiting false positives and false negatives was demonstrated by the precision and recall measures.

The categorization report provided detailed insights into the model's performance across several driving behavior categories. It included accuracy, recall, and F1-score data for each class label. Notably, the accuracy measure provided information on the model's capacity to prevent misclassifications by quantifying the percentage of true positive predictions across all occurrences predicted as belonging to a certain class. On the other hand, the recall metric revealed how well the model captured examples of a certain class by calculating the percentage of true positive predictions among all instances that belonged to that class. The harmonic mean of accuracy and recall, or the F1-score, offered a fair assessment of the

model's performance that took false positives and false negatives into account. This measure was especially useful in situations when striking a compromise between recall and accuracy is critical, such when predicting aggressive driving behavior.When analyzing the confusion matrix, we looked at how the categorization results were distributed throughout the various driving behavior categories, such as "AGGRESSIVE," "NORMAL," and "SLOW." The confusion matrix showed the cases when the model misclassified as well as how well it classified occurrences of each behavior category. This helped direct future cycles of model development and hyperparameter tuning by offering insightful information about possible areas for model improvement and refinement.

After using hyperparameter tuning approaches to maximize performance and give detailed insights into model behavior, the examination of the generated predictive model showed promising results in predicting aggressive driving behavior. We paved the path for improved road safety and accident prevention initiatives by carefully examining hyperparameter setups and rigorously evaluating model performance indicators. This allowed us to acquire important insights into the model's predictive capabilities and possible areas for development.

6. Conclusions and Recommendations:

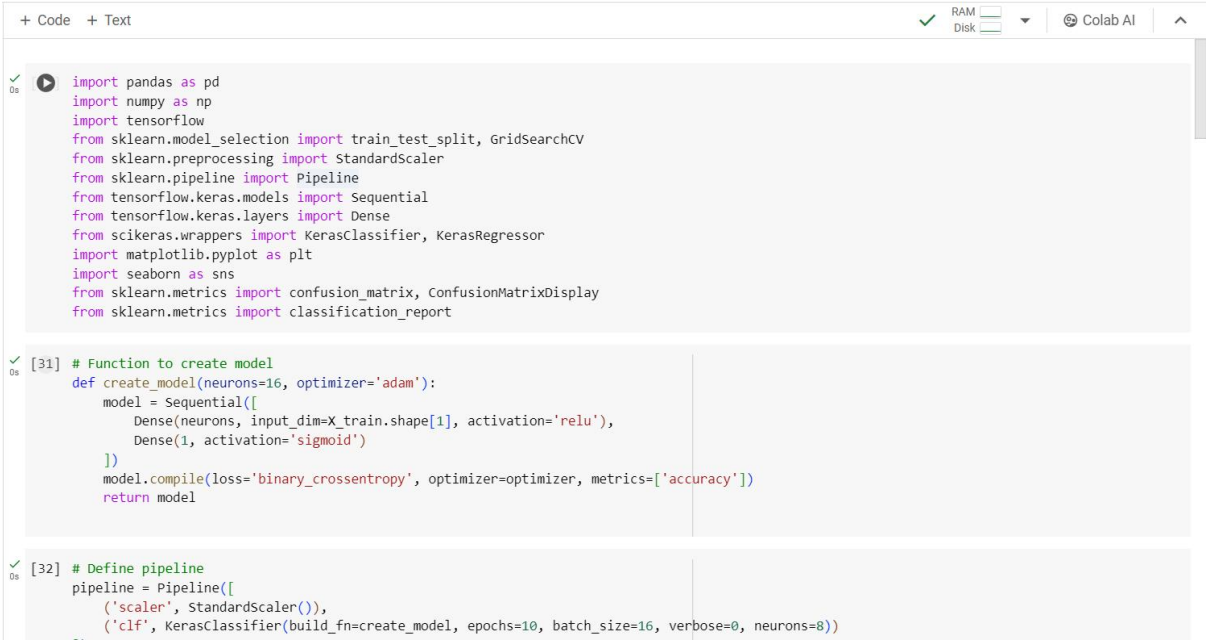
The attempt to anticipate aggressive driving behavior by utilizing deep learning approaches is a noteworthy advancement in improving road safety and reducing the frequency of traffic accidents. By means of methodical data gathering, model building, and hyperparameter adjustment, we have attempted to construct a predictive model that can identify minute differences in driving behavior and provide useful information essential for proactive intervention tactics. Important insights into the effectiveness and performance features of the built predictive model have been obtained from the study carried out on it. Through the application of hyperparameter tuning approaches, we were able to maximize prediction accuracy while optimizing the model's performance and determining the ideal hyperparameter configuration. The model now has the ability to accurately identify aggressive driving behavior thanks to the iterative exploration of hyperparameter configurations that was aided by strict evaluation metrics. This process also identified the best neural network architecture and optimizer choice. Although the obtained results are encouraging, a number of directions for more study and improvement become apparent. First off, the dataset's accuracy and representativeness need to be reviewed and improved upon over time. Future versions of the research will aim to include more cases and a wider range of driving situations to the dataset. This would improve the model's ability to generalize to new data and enable a more thorough investigation of driving behavior patterns.

The confusion matrix illustrates how well the model performs in various driving behavior categories, which highlights the need for focused efforts to fix model deficiencies. To be more precise, in order to create a more inclusive and balanced model architecture, future research projects may involve adding examples of driving behaviors that are not commonly seen to the dataset, such "SLOW" driving. Using ensemble learning strategies like stacking or bagging offers a viable way to improve model resilience and predictive performance. Ensemble learning approaches provide a powerful mechanism to mitigate overfitting and enhance generalization capacity, hence strengthening the model against unknown data instances. This is achieved by merging the predictions of numerous base models trained on distinct subsets of the data.

It is crucial to pay close attention to the interpretability and transparency of predictive models in addition to model development, especially in safety-critical domains like transportation. Therefore, in order to shed light on the model's underlying decision-making processes and foster confidence among stakeholders and end users, future research projects may involve the application of explainable AI techniques like SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations). Predictive models may become more contextually and temporally aware through the fascinating new field of sensor fusion approaches combined with real-time data streams. Predictive models can provide a more comprehensive knowledge of driving behavior by utilizing a wide range of sensor modalities, including as GPS, video, and vehicle telemetry data. This allows for proactive intervention tactics and real-time feedback mechanisms.

The goal of using deep learning techniques to anticipate aggressive driving behavior is a complex undertaking that includes model building, hyperparameter optimization, and data acquisition. By combining these elements and continuously improving predictive models, our goal is to provide a more secure and safe driving environment, which will reduce the frequency of traffic incidents and protect the safety of drivers.

Appendix:



\+ Code + Text

```
[32] # Define pipeline
     pipeline = Pipeline([
         ('scaler', StandardScaler()),
         ('clf', KerasClassifier(build_fn=create_model, epochs=10, batch_size=16, verbose=0, neurons=8))
     ])
```

```
[33] # Define hyperparameters grid
     param_grid = {
         'clf__neurons': [8, 16, 32],
         'clf__optimizer': ['adam', 'rmsprop']
     }
```

```
# Perform hyperparameter tuning
grid_search = GridSearchCV(pipeline, param_grid, cv=3, scoring='accuracy', verbose=1)
grid_search.fit(X_train, y_train)
```

```
Fitting 3 folds for each of 6 candidates, totalling 18 fits
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:925: UserWarning: ``build_fn`` will be renamed to ``model`` in a future release, a
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:925: UserWarning: ``build_fn`` will be renamed to ``model`` in a future release, a
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:925: UserWarning: ``build_fn`` will be renamed to ``model`` in a future release, a
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:925: UserWarning: ``build_fn`` will be renamed to ``model`` in a future release, a
```

\+ Code + Text

```
  X, y = self._initialize(X, y)
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

GridSearchCV

```
GridSearchCV(cv=3,
             estimator=Pipeline(steps=[('scaler', StandardScaler()),
                                       ('clf',
                                        KerasClassifier(batch_size=16, build_fn=<function create_model at 0x7d847c657f40>, epochs=10, neurons
             param_grid={'clf__neurons': [8, 16, 32],
                         'clf__optimizer': ['adam', 'rmsprop']},
             scoring='accuracy', verbose=1)
```

estimator: Pipeline

StandardScaler

```
StandardScaler()
```

KerasClassifier

```
        batch_size=16
        validation_batch_size=None
        verbose=0
        callbacks=None
        validation_split=0.0
        shuffle=True
        run_eagerly=False
        epochs=10
        neurons=8
        class_weight=None
)
```

+ Code + Text

RAM Disk | Colab AI

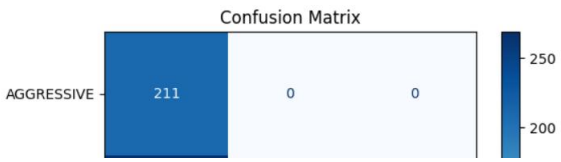
```
[35] # Best hyperparameters
     print("Best hyperparameters:", grid_search.best_params_)
```

```
Best hyperparameters: {'clf__neurons': 8, 'clf__optimizer': 'adam'}
```

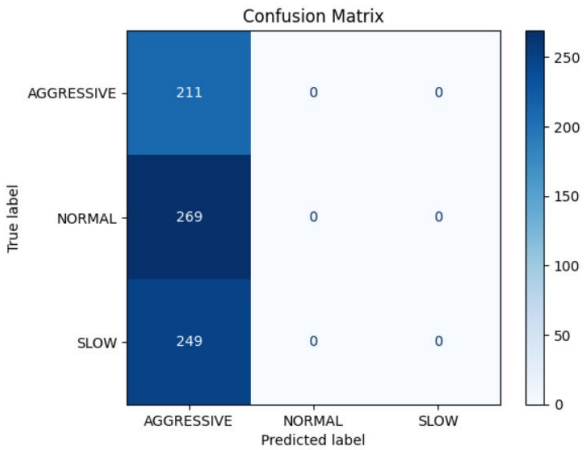
```
[36] # Evaluate model
     best_model = grid_search.best_estimator_
     test_accuracy = best_model.score(X_test, y_test)
     print("Test Accuracy:", test_accuracy)
```

```
Test Accuracy: 0.289437585733882
```

```
# Plot confusion matrix
y_pred = best_model.predict(X_test)
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=best_model.classes_)
disp.plot(cmap='Blues', values_format='d')
plt.title('Confusion Matrix')
plt.show()
```




```
disp.plot(cmap='Blues', values_format='d')
plt.title('Confusion Matrix')
plt.show()
```



+ Code + Text

RAM Disk | Colab AI

```
# Classification report
print("Classification Report:")
print(classification_report(y_test, y_pred, target_names=best_model.classes_))
```

```
Classification Report:
              precision    recall  f1-score   support

  AGGRESSIVE       0.29      1.00      0.45       211
      NORMAL       0.00      0.00      0.00       269
        SLOW       0.00      0.00      0.00       249

    accuracy                           0.29       729
   macro avg       0.10      0.33      0.15       729
weighted avg       0.08      0.29      0.13       729

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1509: UndefinedMetricWarning: Precision is ill-defined and being set t
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1509: UndefinedMetricWarning: Precision is ill-defined and being set t
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1509: UndefinedMetricWarning: Precision is ill-defined and being set t
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

References:

Balan, G. *et al.* (2022) "An improved deep learning-based technique for driver detection and driver assistance in electric vehicles with better performance," *International transactions on electrical energy systems*, 2022, pp. 1–16. doi: 10.1155/2022/8548172.

Bonnet, A. (2023) *Fine-tuning models: Hyperparameter optimization*, *Encord.com*. Encord Blog. Available at: https://encord.com/blog/fine-tuning-models-hyperparameter-optimization/ (Accessed: May 7, 2024).

Elgeldawi, E. *et al.* (2021) "Hyperparameter tuning for machine learning algorithms used for Arabic sentiment analysis," *Informatics (MDPI)*, 8(4), p. 79. doi: 10.3390/informatics8040079.

bin Jamal Mohd Lokman, E. H. *et al.* (2022) "Driving event recognition using machine learning and smartphones," *F1000Research*, 11, p. 57. doi: 10.12688/f1000research.73134.2.

Pannakkong, W. *et al.* (2022) “Hyperparameter tuning of machine learning algorithms using response surface methodology: A case study of ANN, SVM, and DBN,” *Mathematical problems in engineering*, 2022, pp. 1–17. doi: 10.1155/2022/8513719.

Wang, H. *et al.* (2022) “A recognition method of aggressive driving behavior based on ensemble learning,” *Sensors (Basel, Switzerland)*, 22(2), p. 644. doi: 10.3390/s22020644.

Zahid, M. *et al.* (2020) “Predicting risky and aggressive driving behavior among taxi drivers: Do spatio-temporal attributes matter?,” *International journal of environmental research and public health*, 17(11), p. 3937. doi: 10.3390/ijerph17113937.

(No date a) *Researchgate.net*. Available at: https://www.researchgate.net/publication/334769246_Aggressive_Driving_Detection_Using_Deep_Learning-based_Time_Series_Classification (Accessed: May 7, 2024).

(No date b) *Researchgate.net*. Available at: https://www.researchgate.net/publication/354495368_Machine_learning_model_optimization_with_hyper-parameter_tuning_approach (Accessed: May 7, 2024).