**Project Topic:** **Mining the Opinion of Internet Users on Self Driving Cars**

**Abstract:**

This study examines public perceptions of self-driving cars by analyzing a Twitter dataset with sentiment analysis. Using TextBlob, 28,893 tweets are labeled as positive or negative. Preprocessing involves data cleaning, including deduplication, HTML tag removal, lemmatization, and tokenization. WordCloud exploratory analysis uncovers prevalent topics like safety, innovation, and control. Model evaluation maintains a 3.86% class imbalance for robustness testing.

Classical machine learning models (Random Forest and Gradient Boosting) trained on bag-of-words, Word2Vec embeddings achieve ~75% balanced accuracy. Gradient Boosting with bag-of-words excels, hitting 84% accuracy and a strong 0.95 negative recall. In deep learning, LSTM, BiLSTM, and BERT models are implemented. LSTM achieves 93.61% training accuracy and 86.76% validation accuracy over 5 epochs. BiLSTM shows overfitting despite reaching 98.14% training accuracy. BERT stands out with 91% accuracy, >0.89 precision and recall, leveraging transfer learning effectively.

The study's models outperform some previous study’s benchmarks mentioned in the literatures reviewed and offer actionable insights for automakers and regulators to track public opinions on self-driving cars. BERT particularly captures nuanced sentiments as autonomous technologies evolve. Future research should expand labeled datasets, refine neural architectures, and utilize this framework to mine internet users' opinions on self-driving cars.

**Introduction:**

Autonomous vehicles, commonly referred to as self-driving cars, have the capability to navigate highways without human intervention. These vehicles employ a sophisticated array of sensors, artificial intelligence (AI), and computer vision systems to autonomously perceive their surroundings and make informed decisions. This transformative technology holds the potential to grant greater mobility to individuals facing limitations due to age or disability, allowing them to drive independently, thereby reducing their dependence on others (Kawitkar & Deshpande, 2017).

The advent of autonomous or self-driving vehicle technology has ignited robust discourse within the general populace. With prominent entities like Tesla, Waymo, and Cruise competing to bring autonomous vehicles into the mainstream, it is imperative to grasp the prevailing sentiment of the public regarding this potentially transformative innovation. While the technology holds the promise of enhancing safety, accessibility, and reducing environmental impact, it also gives rise to a spectrum of concerns, encompassing technical glitches, safety hazards, legal implications, privacy infringements, and ethical dilemmas.

These concerns persist alongside the acknowledged benefits. As Mattar (2022) highlights, apprehensions regarding the potential vulnerabilities to hacking and software failures in self-driving vehicles linger as reservations to the widespread adoption of this technology."

Understanding public opinions, apprehensions, and acceptance is important for a smooth transition as self-driving technology develops. To achieve this, we need to find a way to extract people’s opinion around self-driving cars. Social media platforms like Twitter provide a rich source of grassroots data that can be mined to gauge public sentiment and discourse on self-driving cars. The opinion mining and analysis of people’s opinion on the internet provides insightful feelings, concerns, and perceptions around self-driving cars by utilising the lively dialogue of internet users. Analysis of these sentiments can enable automakers to monitor their product life cycle and obtain a data-driven perspective on consumer attitudes and feedback (Dutta and Das, 2020). It will also help to allay the fears of the populace or point out to all the stakeholders the areas where improvements are needed.

Comprehending public sentiments, reservations, and receptivity stands as a pivotal facet in facilitating the seamless evolution of self-driving cars. Realizing this, there arises a need to establish mechanisms for distilling the collective opinions of the public concerning self-driving automobiles. In this endeavour, social media platforms such as Twitter emerge as invaluable repositories of grassroots data that can be harnessed to gauge the prevailing public sentiments and discussions regarding self-driving cars.

The practice of mining and analysing public opinions online yields a treasure trove of valuable insights, encompassing feelings, concerns, and perceptions pertaining to self-driving vehicles, and does so by leveraging the dynamic interactions of online users. By systematically scrutinizing these sentiments, automobile manufacturers can proactively oversee the lifecycles of their products, thereby cultivating a data-driven perspective on consumer attitudes and feedback, as underscored by Dutta and Das (2020). Furthermore, this approach can play an instrumental role in alleviating public apprehensions or illuminating areas of potential enhancement to all relevant stakeholders."

To address gaps and concerns, we adopt a comprehensive approach involving NLP and conventional machine learning. This approach constructs a predictive model for categorizing tweets as positive or negative. We conduct a thorough literature review and explore supervised learning methodologies like Random Forest, Gradient Boosting, Bag of Words, Word2Vec, LSTM, BiLSTM, and BERT to dissect public opinions on self-driving cars. This ensures the selection of suitable algorithms for opinion mining and analysis.

**Research Objectives:**

The primary objectives of this work are:

* To build a predictive model that will be used to assess overall sentiment polarity (positive, negative, or neutral) towards self-driving cars.
* To thoroughly examine text categorization methods (supervised learning like Random Forest, Gradient Boosting, and deep learning with LSTM, BiLSTM, BERT) to determine the most effective approach for evaluating public sentiment on self-driving cars using metrics and confusion matrices.

**Related Work:**

**Supervised Learning:** In the realm of supervised learning, extensive research has been conducted in the domain of text classification, focusing on the analysis of sentiments and public opinions concerning self-driving cars. As articulated by Sadiq and Khan (2018), Twitter emerges as a valuable platform replete with embedded features conducive to polarity assessment. Their study encompassed the training of four distinct models employing the Random Forest Classifier on a Twitter dataset, facilitating the prediction of Twitter users' sentiments. Their endeavours culminated in noteworthy classification results, with a precision of 0.562, a recall of 0.585, an F1-Score of 0.490, and an accuracy of 62.24%.

Similarly, Kohl et al. (2018) delved into the intricate evaluation of the risk and benefit perceptions associated with self-driving cars, utilizing Twitter data. Their research harnessed the capabilities of supervised learning techniques to anticipate the adoption trends of emerging technologies, such as self-driving cars. Within their model, the Support Vector Machine (SVM) played a pivotal role in classification, yielding a notably high accuracy metric of 0.925. Regrettably, their work did not disclose the precision, F1-Score, and recall, precluding a comprehensive analysis of the classification report.

In the landscape of related research, Ahmad et al. (2017) employed SVM to train a pre-labelled Twitter dataset, purposefully curated to mine the opinions of Twitter users regarding self-driving cars. This dataset encompassed a total of 7,156 entries, pre-labelled into categories including Very Negative, Slightly Negative, Neutral, Slightly Positive, Very Positive, and Irrelevant. The outcomes of their study culminated in average classification results across the pre-labelled dataset, with precision at 0.558, recall at 0.559, an F1-Score of 0.572, and an accuracy of 59.91%.

**Deep learning:** In the domain of sentiment analysis related to self-driving cars, deep learning methodologies including Long Short-Term Memory (LSTM), Bidirectional LSTM (BILSTM), and BERT have gained considerable traction. Dutta and Das (2020) affirmed the efficacy of deep learning tools, particularly Long Short-Term Memory (LSTM), in discerning user emotions pertaining to self-driving cars. Their endeavours yielded commendable results, with an accuracy of 85.5% for the training dataset, 82.6% for the validation dataset, and 82.8% for the test dataset.

In a parallel vein, Trivyza (2018) undertook extensive experimentation by training various BERT models, encompassing BERT (base), DistilBERT (base), RoBERTa (base), and BERTTweet (base), on a tweet dataset, aimed at extracting the sentiments of internet users regarding self-driving cars. The achieved accuracies for the trained BERT models, including BERT (base), DistilBERT (base), RoBERTa (base), and BERTTweet (base), were 66%, 62%, 62%, and 65%, respectively. The author also explored the realm of deep learning by employing BILSTM, obtaining an accuracy score of 61%. It's noteworthy that this study was conducted on unlabelled data, a similarity shared with the current research endeavour.

Numerous studies have employed sentiment analysis to assess opinions drawn from diverse sources such as Twitter, YouTube, Reddit, and polls. Notably, Gupta et al (2022) study titled 'Analysis of Public Perception of Autonomous Vehicles Based on Unlabelled Data from Twitter using VADER’s sentiment intensity analyser' stands out. This research harnessed a substantial dataset of 35,476 Twitter entries to categorize public opinions into negative, positive, and neutral categories.

Moreover, in a study entitled 'Exploring Trust in Self-Driving Vehicles Through Text Analysis,' Lee et al. (2019) delved deeply into the multifaceted determinants shaping public perceptions of self-driving cars. The investigation unveiled a spectrum of factors contributing to negative sentiments regarding autonomous vehicles, encompassing dimensions related to trust in automation, the yearning for control, societal influences, experimental aspects, and the foundational elements of trust.

The result obtained from this work will be used to compare the results of related works mentioned here and use them to make inform decisions.

**Data Collection:**

In pursuit of current public sentiment analysis, a recent Twitter dataset of 28,893 entries was meticulously gathered and analyzed to reveal public sentiment on self-driving cars. Findings hold potential to inform manufacturers, governments, and stakeholders, fostering a comprehensive understanding. The University of Hull's Department of Artificial Intelligence, Data Science, and Modelling (DAIM) facilitated curation, employing specific hashtag keywords like 'self-drivingcars' and 'autonomouscars' for data collection.

**Methodology:**

To enable a thorough and robust examination of user-generated material on twitter platform regarding self-driving cars, this study took a mixed methods approach combining both computational text mining approaches and qualitative human analysis. The steps listed below were taken to arrive at our objectives.

* **Text Cleaning: Text cleaning techniques** such tokenization, lowercasing, lemmatization, html removal, stopword removals and special character normalization helps to improve the quality input data and the performance of NLP models by lowering text data noise (Kim, 2022). These techniques were used during the data preprocessing.
* **Supervised Learning:** The two major algorithms used in this work are the Random Forest and GradientBossting**.** These ensemble models function to reduce variability in predictive model proficiency and improve overall predictive performance, surpassing individual members within the ensemble (Brownlee, 2020). They excel at addressing non-linear patterns and intricate data correlations through ensemble techniques, enhancing precision in sentiment prediction. Because of a little variation in the class balance of 3.86%, Random Forest algorithm was first used to train the model without balancing the data and later was retrained on balanced data. To balance the data, SMOTE balancing algorithm was used. The dataset for all the algorithms trained under supervised learning were portioned into 70% training and 30% data. Some of the algorithms trained under supervised learning learnings are Random-Forest on Bag of words, Random-Forest on Bag of words with max-dept tuned to 4, Random-Forest on Word2Vec embeddings with Skip-gram, Random-Forest on Word2Vec embeddings with Continuous bag of words, Gradient-Boosting on Bag of words, Gradient-Boosting on Continue Bag of words and Gradient-Boosting on Skip-gram.
* **Deep learning:** The algorithms used in this work under unsupervised learning are Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM) and Bidirectional Encoder Representations from Transformers (BERT). This is because of their proven result towards text classification. For example, LSTM algorithm performs well in text classification using python due to its ability to capture lengthy dependencies and analyze sequences (Shekhar, 2021). BiLSTM on the other hand uses two LSTM models - one processing the input sequence forward, the other backward. This bidirectional approach provides more contextual information to the network, improving performance. Models based on transformers, such as BERT, have excelled at extracting context and semantics from textual data (Muller, 2022). The architecture of each of these algorithms are explained below:
  + Long Short-Term Memory (LSTM): This LSTM model has reduced complexity to 128 embedding dim, 64 LSTM units and dropout rates of 0.4 and 0.3 for regularization. L2 kernel regularization further prevents overfitting. Early stopping after 2 epochs without validation improvement and model checkpointing are used. The model is trained for 10 epochs with Adam optimizer, binary crossentropy loss, batch size 32, and 20% validation split. Overall, techniques like reduced parameters, regularization, and early stopping aim to improve generalization of this LSTM text classification model.
  + Bidirectional LSTM (BiLSTM): The setup includes an embedding layer with a 10,000-word vocabulary, 128-dimension embeddings, and a 20-sequence input. Incorporating various dropout layers, Bidirectional LSTMs, and a final output layer, the model is optimized using binary cross-entropy and Adam optimizer, trained for 10 epochs with specific batch and validation split sizes.
  + Bidirectional Encoder Representations from Transformers (BERT): A pre-trained BERT model ('bert-base-uncased') and its corresponding tokenizer was used to train the model.

Tools like Python. Sklearn, Keras, tensorflow, transformers, Pandas, NumPy, worldCloud, torch and other machine learning libraries were used to process and train the models.

**Discussion and Result:**

To build the predictive models, a raw and unlabeled Twitter dataset was used. The steps listed below were followed to arrive to my goals:

* **Dataset labeling:** The acquired Twitter dataset remained in its original unprocessed state. To introduce a sentiment column, the TextBlob algorithm was deployed for classifying Tweets into positive and negative categories. This choice was informed by TextBlob's integrated sentiment analysis capability. The sentiment property produces a structured tuple, Sentiment (polarity, subjectivity). Polarity, a floating-point value in the [-1.0, 1.0] range, signifies sentiment strength. Subjectivity, ranging from 0.0 (high objectivity) to 1.0 (high subjectivity), enables TextBlob's handling of unlabeled data (TextBlob, n.d). This initial classification of unlabeled data set the foundation for subsequent steps.
* **Data Cleaning:** Various data exploration techniques were employed to identify inconsistencies, null values, and duplicates. A total of 1508 duplications were identified and subsequently removed, resulting in 27385 rows and 2 columns, down from the original 28892 rows.
* **Exploratory Data Analysis:** WordCloud was employed to analyze positive and negative words in the dataset, specifically in the Tweet and Sentiment columns. The Tweet column is the first, and the Sentiment column is the second. This approach facilitates a comprehension of the most frequent words before initiating data preprocessing. In Figure 1.0 and 2.0, the discovered words for the Tweet column are illustrated.

A close-up of words

Description automatically generated

Figure 1.0 – Negative words in the tweet column

Some of the negative words observed are crash, killed, worse, afraid, problem, wrong, least, and worst.

A close-up of words

Description automatically generated

Figure 2.0 – Positive words in the tweet column

Some of the positive words are safety, live, innovation, great, new, want, able, ready, Thank, safe, smart, and love.

* **Data Balance:** Sentiment column data balance was assessed, revealing a marginal 3.86% difference between classes. Considering the slight disparity, additional data balancing algorithms like SMOTE or oversampling were deemed unnecessary. Two models were constructed using both balanced and unbalanced data scenarios to ensure result reliability. The detailed outcomes will be discussed in the section on Supervised Learning Predictive Models. In Figure 3.0, the negative (0) class comprises 51.93%, and the positive (1) class encompasses 48.07% of the data.

A graph of a bar chart

Description automatically generated with medium confidence

Figure 3.0 percentage of data make-up of the Sentiment classes.

**Data preprocessing:**

* The following techniques were applied to clean up the dataset for model training and evaluations.
  + HTML Tags Removal: unstructured text like Tweeter dataset contains a lot of noise and HTML tags are one of those components that introduce noise and doesn’t add many values towards text analysis (Yadav, 2020). To ensure that our model is rid of such noise, HTML tags were removed.
  + Fix Contractions: The presence of contractions poses redundancy challenges in Natural Language Processing (NLP) models. Contractions are not inherently recognized as abbreviations by computers and NLP models. As a result, they augment the dimensionality of the document-term matrix. For instance, 'we are' and 'we're' may necessitate separate columns. This escalation in dimensionality leads to heightened computational complexity, thereby increasing processing costs (Lukei, 2019). To avoid this in my work, the contraction was fixed.
  + Tokenization: I employed this cleaning method to partition the text into more manageable segments, facilitating machine comprehension of human language. Tokenization, as described by Chakravarthy (2020), involves the division of raw text into units like words or sentences, referred to as tokens. These tokens serve as the basis for training my Natural Language Processing (NLP) models.
  + Digits Removal: Numerical digits possess the potential to disrupt the functioning of natural language processing algorithms, meticulously crafted for the interpretation and manipulation of textual data (Silva, 2023). To preserve the optimal performance of our model, a dedicated cleaning approach was diligently applied to excise these numeric elements.
  + Remove Non-Ascii character: This cleaning technique was used to clean up dataset to ensure that unnecessary characters like symbols are removed which helps to ensure cleaner and better performance of my model.
  + Lemmatization: "Lemmatization transforms words to their base form based on context, e.g., 'running' to 'run’ and is more context-aware than stemming (Otten, 2023). This technique was used.

Other cleaning techniques used ranged from punctuation removal, lowercasing, stopword removals, regular expression, and vectorization which is a conversion of texts to numbers to ensure a clean dataset for the model building.

**Model Building:**

**Supervised Learning:** Different models were trained using RandomForest and GradientBossting algorithms with their outputs evaluated. This section will be discussed based on these two algorithms. They are discussed as shown below:

* **Random Forest with Bag of Words (BOW):** The initial model underwent training using unbalanced data and was subsequently trained with data balanced via SMOTE algorithms. Surprisingly, the results for both balanced and unbalanced data remained consistent.

In Figure 4.0, the classification report and confusion matrix reveal compelling statistics. Notably, the model attained robust precision scores of 0.82 for the negative class and 0.87 for the positive class, indicating a strong ability to discern respective sentiments. Furthermore, the model demonstrated commendable recall scores of 0.89 for negative and 0.79 for positive, accurately capturing the actual sentiments. Balanced F1 scores of 0.86 for negative and 0.83 for positive underscore the model's balanced precision and recall. The confusion matrix showed that model correctly classified 3872 as True-Negative, 3101 as True-Positive but misclassified 449 texts as False-Positive and 839 as False Negative.

With an overall accuracy of 84%, this model showcases promising performance in a challenging sentiment analysis task, suggesting its competitiveness with benchmark models in similar contexts which are available in repositories like Kaggle, OpenMl, Papers with code and UCI Machine Learning Repository.

A screenshot of a graph

Description automatically generated

Figure 4.0 Classification report and Confusion matrix of RandomForest With BOW.

Worcloud was used to visualize top 50 words of the RandomForest model on Bag of words as can be seen in Figure 5.0 below. Some of the top words observed are good, thanks, ready, latest, new, top, safe, and many others.

A close up of words

Description automatically generated

Figure 5.0 Top 50 featured wordcloud

I also have other RandomForest models which were trained with BOW with the max\_depth tuned to 4, Skip-gram and Continuous Bag of Words. Figure 6.0 below depicts other results obtained.

A table with different colors and text

Description automatically generated

Figure 6.0 Classification reports of various algorithms using Random Forest

For the RF\_BOW with a max depth of 4, the negative class shows high recall (0.97) but lower precision (0.68), indicating it captures most actual negative instances but misclassifies some. Conversely, the positive class has high precision (0.95) but lower recall (0.51), suggesting it accurately predicts positives but misses several actual positive instances. In contrast, both Random Forest models using Word2Vec embeddings exhibit balanced precision and recall for both classes, with F1-scores around 0.70. While RF\_Word2Vec (Skipgram) demonstrates slightly improved metrics for both classes, all models achieve a balanced accuracy of 0.75, showcasing consistent overall performance across these algorithms in distinguishing between positive and negative sentiments.

* **GradientBoosting Algorithm:** This algorithm was used to train the predictive model using Bag of Words, Skip-gram, and Continuous Bag of words. Figure 7.0 shows all the results obtained.

**A table with text and words

Description automatically generated**

Figure 7.0 Classification reports of GradientBoosting algorithms.

The analysis of these algorithms showcases the performance in classifying negative and positive classes. In the Gradient Boosting model, GB-BOW, the negative class displays a well-balanced precision and recall, achieving a high F1-score of 0.86, indicating strong overall performance. However, the positive class demonstrates a trade-off between precision and recall, resulting in an F1-score of 0.81. Similarly, both GB\_Word2Vec models, Skipgram and Continuous Bag of Words (CBOW), present balanced precision and recall for both classes. The F1-score for Skipgram is 0.75 for the negative class and 0.71 for positive class while that of CBOW is 0.73 for negative class and 0.68 for positive class. The GB-BOW exhibits superior metrics for the negative class, all models achieve an accuracy of 0.84, showcasing consistent overall performance in correctly classifying negative and positive sentiments, with the GB-BOW algorithm standing out for its balanced precision and recall in the negative class.

**Deep Learning Models**: Three algorithms were used on the same dataset to train a predict model. The algorithms are Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), and Bidirectional Encoder Representations from Transformers (BERT). The results of these models are as discussed below:

* **Long Short-Term Memory (LSTM**): Different LSTM model were trained, tweaked, and tuned before we got the beset performing model. Figure 8.0 show the epochs and evaluation performance of this model.

**A screenshot of a computer code

Description automatically generated**

Figure 8.0 Epoch output of the trained LSTM model.

The model trained over 5 epochs exhibited noticeable improvement. Training loss decreased from 0.6098 to 0.1947, signifying enhanced fit to the data. Training accuracy rose from 76.93% to 93.61%, indicating learning. Notably, epoch 3 achieved the best validation loss of 0.3696, a marked improvement from 0.3733. Validation accuracy peaked at 86.76% in epoch 3, moderately surpassing 85.87% in epoch 1. With continued enhancement in validation metrics across epochs, the model showcased consistent progress, hinting at improved generalization.

* **Bidirectional Long Short-Term Memory (BiLSTM**): The BiLSTM model trained over 10 epochs displayed evident improvements. Training loss reduced from 0.4357 to 0.0489, showcasing enhanced predictive accuracy. Training accuracy surged from 0.7990 to 0.9814, signifying robust data classification. However, validation loss fluctuated, ending at 0.6687, hinting at potential overfitting. Validation accuracy remained stable at 0.85-0.87, reflecting competent performance on unseen data. While the model exhibited significant learning in training, fluctuating validation loss suggests a need for regularization techniques to bolster generalization.
* **Bidirectional Encoder Representations from Transformers (BERT):** Utilizing a pretrained BERT model on textual data yielded robust classification performance. Illustrated in Figure 9.0, the model presented high precision (0.93 for class 0, 0.89 for class 1) and recall (0.90 for class 0, 0.93 for class 1), resulting in balanced F1-scores of around 0.91 for both classes. The overall accuracy of 0.91 underscores its strong predictive capability. The confusion matrix showcased accurate identification of instances, with slightly higher accuracy in class 0. Balanced precision, recall, and high accuracy collectively highlight the model's effectiveness in distinguishing between classes, emphasizing its reliability in classification tasks.

**A screenshot of a graph

Description automatically generated**

Figure 9.0 Confusion matrix and classification report of the pre-trained BERT model.

**Statistical Model Evaluation and Comparison**

A comprehensive analysis was conducted on 10 predictive models, revealing robust performance based on diverse evaluation metrics. Their accuracy spanned from 0.71 to 0.98 across all classes, aligning with established machine learning thresholds cited in various publications, including Barkved (2022), Hendricks (n.d), and Allwright (2022), as referenced in related work. However, caution was advised, emphasizing the need for additional metrics beyond accuracy.

Notably, among classical models, Gradient Boosting (GB) with Bag of Words (BoW) achieved the highest accuracy of 84% and excelled in negative sentiment recall at 0.95, while Random Forest (RF) with BoW displayed balanced precision and recall. Conversely, Word2Vec embeddings seemed less impactful for classical models.

The BERT base model exhibited the highest overall accuracy of 91% with remarkable precision and recall (>0.89) for both classes, outperforming classical models, underscoring the potency of transfer learning from pre-trained deep models like BERT.

The LSTM and BiLSTM models indicated high accuracies, BiLSTM revealed signs of overfitting, with LSTM displaying superior generalization. The validation loss for BiLSTM also did not improve over the 10 trained epochs. Ultimately, BERT emerged as the preferred model, excelling in accuracy, contextual understanding, and minimal class discrepancies, showcasing the efficacy of deep transfer learning for nuanced textual analysis. The balance between performance and interpretability emphasizes BERT's significance over classical ML models, while careful model interpretation and hyperparameter tuning remain vital considerations for model finalization.

A screenshot of a spreadsheet

Description automatically generated

Figure 10.0 Table containing the Evaluation Metrics of 10 Models Trained.

**Comparison with Related work in the same field**: Upon reviewing related work within our field, it's evident that several models demonstrated commendable performance, though suitability for production varied according to our objectives. For instance, Sadiq and Khan's study (2018) on analyzing self-driving cars on Twitter reported precision, recall, F1-score, and accuracy at 0.652, 0.585, 0.490, and 62.24%, respectively. Another study by Kohl et al. (2018) explored the acceptance of emerging technologies in self-driving cars, achieving 0.925 accuracy. Meanwhile, Ahmad et al. (2017) in their work on sentiment analysis of tweets using SVM, reported precision, recall, F1-score, and accuracy at 0.558, 0.559, 0.572, and 59.9%. Our classical machine learning models outperformed these related works.

Regarding RNN models, a study by Duta and Das (2020) on deep sentiment analysis with LSTM achieved 85.5% accuracy, reinforcing the performance of LSTM and BiLSTM models with 90% and 98% accuracy, respectively.

Comparing our BERT model to Trivyza's (2018) work on autonomous vehicles sentiment analysis, our model excelled, reporting an accuracy of 91%, a substantial improvement over the related work's BERT models.

**Outcomes:**

The work and all the analysis have shown that the two objectives set to achieve has been fulfilled. The outcomes and how the fulfil the objectives are as follows:

* After training a diverse set of 10 models, including Classical machine learning and Deep learning approaches, we've successfully developed a deployable BERT predictive model for analysing internet users' opinions on self-driving cars.
* Our comprehensive examination of text categorization methods, encompassing supervised learning (Random Forest, Gradient Boosting) and deep learning (LSTM, BiLSTM, BERT), highlights the effectiveness of all these algorithms for constructing predictive models in opinion mining. Our top-priority models are BERT (91% accuracy, high precision, and recall), GB with BoW (84% accuracy), RF with BoW (84% accuracy), LSTM (91% validation accuracy), and RF with Word2Vec Skipgram (75% accuracy).

**Recommendation:**

The analysis forms a strong foundation for autonomous vehicle opinions but offers avenues for improvement. BERT's tuning, BioBERT embedding, interpretability tools like SHAP and LIME, and classical ML techniques could enhance performance and trust. Addressing overfitting in LSTM models and refining models through tuning, engineering, and interpretability will advance nuanced public opinion understanding.

**Conclusion:**

This comprehensive analysis establishes an effective framework for discerning public sentiment on autonomous vehicles using sophisticated NLP techniques. The work successfully develops a deployable BERT model with remarkable 91% accuracy that excels in sentiment classification. The examination of diverse algorithms highlights the capabilities of deep transfer learning through BERT combined with intrinsically interpretable classical ML models. While extensive tuning remains vital across approaches, the analysis paves the way for impactful real-world application of advanced NLP to derive nuanced insights from public opinions on the critical domain of self-driving automobiles. Overall, the study provides a robust foundation for integrating state-of-the-art techniques to enable insightful opinion mining.