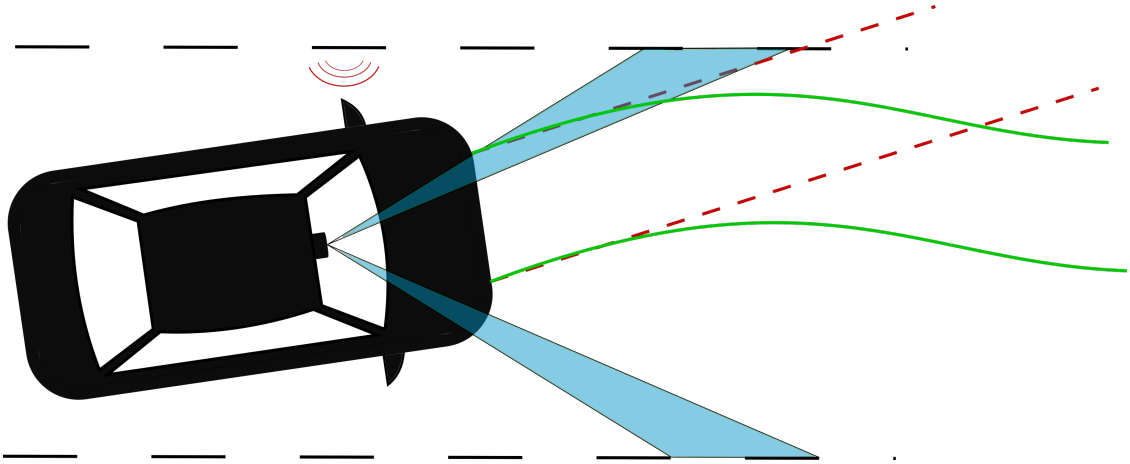
CHALMERS
UNIVERSITY OF TECHNOLOGY



A neural network-based lane-keep assist (LKA) function

A sensitivity analysis related to prediction accuracy and false positives

Master's thesis in Mobility Engineering

CHINMAY SHARMA
LAKSHMANAN SUBRAMANIAN

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES
DIVISON OF VEHICLE SAFETY

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2023
www.chalmers.se

MASTER'S THESIS IN MOBILITY ENGINEERING

A neural network-based lane-keep assist (LKA) function

A sensitivity analysis related to prediction accuracy and false positives

CHINMAY SHARMA
LAKSHMANAN SUBRAMANIAN

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mechanics and Maritime sciences
Division of Vehicle Safety
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2023

A neural network-based lane-keep assist (LKA) function
A sensitivity analysis related to prediction accuracy and false positives
Chinmay Sharma Lakshmanan Subramanian

Master's Thesis 2023
Department of Mechanics and Maritime Sciences
Division of Vehicle Safety
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone +46 31 772 1000

Cover: A car sensing the lane and making the decision to steer

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2023

A neural network-based lane-keep assist (LKA) function
A sensitivity analysis related to prediction accuracy and false positives
Master's thesis in Mobility Engineering
Chinmay Sharma Lakshmanan Subramanian
Department of Mechanics and Maritime Sciences
Division of Vehicle Safety
Chalmers University of Technology

Abstract

Road accidents are one of the leading causes of deaths worldwide (*Road traffic injuries* 2022). Around 1.3 million fatalities were reported due to road traffic crashes, and the injuries and economic losses are even higher. The cause of the majority of these car accidents can be attributed to human errors while driving (*research note: 2016 fatal motor vehicle crashes: overview - transportation* 2017), emphasizing the importance of reliable safety systems in modern automobiles. Lane Keep Assist (LKA) is one such system commonly referred to as an Advanced Driver Assistance System (ADAS). The LKA system plays a critical role in ADAS-equipped vehicles by assisting vehicles in maintaining their lane positions. This research explores various factors that affect the performance of a neural network-based LKA system. Real-world driving data, including ego vehicle states and environmental information captured through a forward-looking camera, has been collected and been made available for this study. The collected data then underwent processing and normalization as part of this study, to facilitate subsequent machine learning and analysis.

To enhance the dataset's variability and improve the performance of the machine learning models, various data augmentation techniques were employed. The augmented data, along with an appropriate sample size, was then used to train different machine learning models. A main objective was to determine the optimal combination of data sampling, data augmentation, and machine learning algorithms.

The evaluation of the models is based on multiple metrics, with a primary focus on intervention prediction accuracy. This metric measures the system's ability to accurately predict the need for LKA intervention based on the input signals provided. Additionally, the Area Under Curve of the Receiver Operating Characteristics curve (AUC-ROC) is used as a secondary evaluation metric. Furthermore, tools such as the confusion matrix are utilized to obtain a visual representation of the system's performance.

The findings of this study provide valuable insights into the influence of a variety of modeling parameters and methodological choices on the performance of neural network-based LKA systems.

Keywords: Road accidents, Advanced Driver Assistance System (ADAS), Lane Keep Assist (LKA), False Positives, Neural Network

Contents

Preface

This study is an attempt to create a Lane Keep Assist (LKA) system using Neural Network. The thesis has been proposed by Qualcomm and was carried out from January 2023 to June 2023 at their office in Gothenburg.

We offer our sincere gratitude to Jonas Bärgman for examining the project and providing invaluable guidance throughout the project, including in report writing. We appreciate him spending his precious time regularly to ensure the swift continuation of the thesis.Additionally, we are grateful for the supervision and support of Cheerudeep Chintha, whose expertise and direction were essential in navigating the thesis from start to end. We also extend our appreciation to Adrian Welter from LSS team, who helped us comprehend the existing system during the initial phases of this research.
We extend our gratitude to Aykut Argun, a valuable member of Qualcomm's Pre-development team, for providing exceptional support in the field of Machine Learning and offering invaluable suggestions for novel methodologies. Additionally, we would like to express our appreciation to Emma Hilmersson, the manager of the LSS team, for her assistance during our onboarding process and for facilitating our access to the office workspace, which greatly contributed to our overall progress.
Finally, we would like to express our heartfelt appreciation to all the employees at Qualcomm in Gothenburg whose support and cooperation made our time at the company enjoyable and rewarding.
Chinmay Sharma Lakshmanan Subramanian, Gothenburg, June 2023

1 Introduction

1.1 Background

Road accidents have consistently remained a significant concern for our society. According to a report by (WHO 2018), road accidents are the eighth leading cause of death in the world. The number of injuries is close to 50 million annually, some of them requiring serious medical attention (Ritchie, Spooner, and Roser 2018). The consequences of these accidents are not limited only to impacting human lives. Fatal and non-fatal injuries are estimated to cost the world nearly 1.8 trillion dollars from 2015 to 2030 (*NHTSA estimates for first nine months of 2022 suggest roadway fatalities beginning to level off after two years of dramatic increases* 2023).

As concluded in a study conducted by the National Highway Transportation Safety Administration (NHTSA), human error accounts for 94-96% of road vehicle accidents (*research note: 2016 fatal motor vehicle crashes: overview - transportation* 2017). This includes causes such as distraction, drowsiness, reckless driving, and intoxication. Despite federal laws in many countries mandating safe driving practices, instances of distraction and drowsiness are still observed frequently on long drives on highways, which could lead to safety-critical situations such as unintended lane departure.

Lane departure is one of the most common safety-critical road behaviors that lead to crashes, especially when it comes to resulting fatalities (Dean and Riexinger 2022). Advanced Driver assistance systems (ADAS) have been developed to aid drivers in preventing many such situations, including unintentional lane departure. A Lane Departure Warning (LDW) system can prevent such situations by warning the user through audio and visual cues when the user crosses the lane markers. A Lane Keep Assist (LKA) system offers similar functionality but instead triggers an intervention by applying a gentle steering input to keep the vehicle within the lane when the user does not respond to the LDW warning.

ADAS systems in general have found good acceptance from drivers (ConsumerReports 2019). Though LKA has been implemented with good effectiveness in vehicles (Sternlund et al. 2017), it is limited by weather conditions (snow or rain causing an error in sensor perception of the road and environment) and drivers' noncompliance i.e., the driver turning the feature off, citing "too many false alarms" (*Erie Insurance* 2020). It is therefore imperative to make the system less prone to false interventions to improve driver satisfaction and give the system more opportunity to save the users from accidents caused by unintended lane departures.

Several attempts have been made to optimally design the LKA system. These approaches can broadly be classified into rule-based approaches and learning-based approaches. Rule-based approaches involve modeling the dynamics of the system (or 'plant') and using control methodology to optimize system parameters. (Marino et al. 2009) proposed an active steering controller that uses nested Proportional integral derivative (PID) control

with only two inputs: yaw rate from the gyroscope and lateral offset from the vision system. They simulated their controller in CarSim (*CarSim* n.d.) simulation environment where their model showed improved performance over the pre-existing controller in CarSim.

While rule-based approaches offer high interpretability, they suffer from a lack of adaptability of novel and uncertain driving scenarios. This limitation can be addressed by using a large amount of driving data and with a learning-based approach. This also ensures that complex non-linearities could be included in the solution. Tan, Chen, and Wang (2017) proposed an architecture to predict lane departure using Deep Fourier Neural Network (DFSS). They used Monte-Carlo simulation to generate different vehicle states (this included variables like lateral speed, acceleration, yaw angle, etc) and calculated time to Lane change for each. This was used for training the DFSS. The authors found out that this method reduced the false-positive rate significantly, while also ensuring driver safety by limiting lateral acceleration. Deep learning also enables incorporating driver behavior as one of the influencing parameters which would lead to a more user-friendly system design. Beglerovic et al. (2018) used Deep Learning to classify various driving scenarios into those relevant to LKA. In some cases, the problem can instead be reversed, and the system performance can help predict driver behavior. McDonald et al. (2013) monitored the steering wheel angle to identify patterns that may indicate driver drowsiness.

1.2 Aim and Objectives

This thesis focuses on the analysis of driving data, consisting of approximately 3600 annotated segments with four labels: True positive, False positive, True negative, and False negative. The labeled data is then processed to extract information about the specific time intervals and signal states associated with each event. This serves as a foundation for the machine learning model to comprehend the environment and establish correlations with the desired output. Various machine learning and data processing techniques are explored, and the performance of the model is assessed to determine the impact of different parameters on the output's sensitivity.
The aim is to explore a neural network-based approach to model an LKA system. The objective is to assess the impact of a set of factors on the performance of a neural-network-based LKA system, to help developers make more informed decisions in the design of such systems. This is a learning-based algorithm that is trained on data available from various driving scenarios of a car equipped with the conventional LKA system. The research can be formulated into the following research questions:

1. *Is it possible to achieve good prediction accuracy for LKA intervention using Neural Networks?*
2. *Are there ways through which one can work with limited annotated data and still achieve good accuracy?*
3. *What is the effect of different parameters such as sequence length, data augmentation, and different machine learning methods on the output model performance, and which method is best to adopt?*

The thesis is organized as follows. The theory section, later in this chapter, covers the theoretical background of the work.The methods section explains the methodology used to prepare the Machine Learning model for the given problem. The results section presents the various result metrics for the model performance. In the discussion section, the authors' inference, conclusion, as well as limitations and proposed future work are presented.

1.3 Theory

1.3.1 Neural Networks

Neural Networks emerged as a mathematical model inspired by the functionality of Neurons in the brain. (Palm 1986) used logical calculus to define the inputs and outputs of these mathematical neurons. A major breakthrough came in 1958 with the introduction to the perceptron by (Rosenblatt 1958) which was the first and simplest neural network.

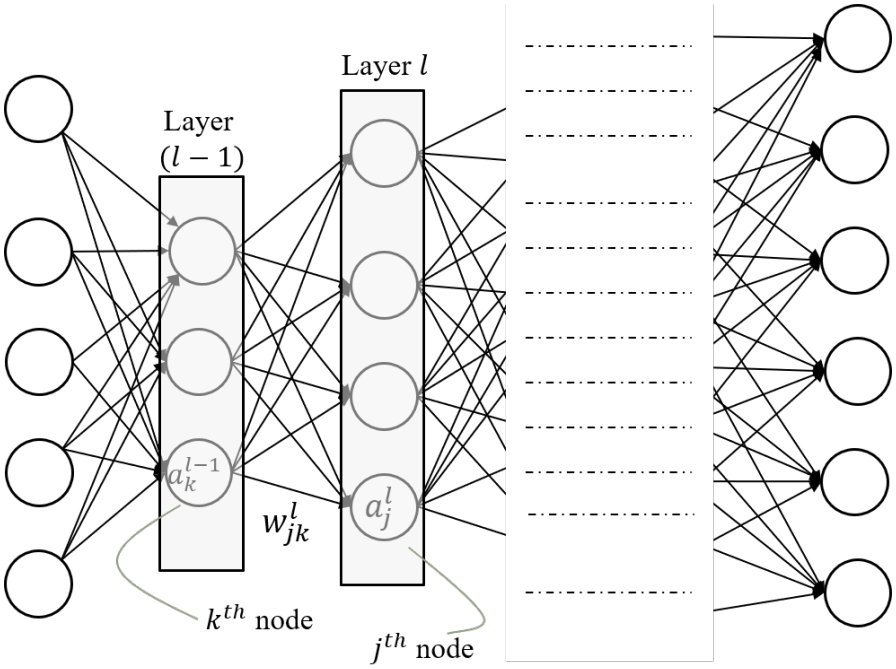


Figure 1.1: Neural Network Architecture

Mathematically speaking, each node (see Figure 1.1) is dependent on the nodes before it, in the following way:

$$z_j^l = \sum_{k=1}^{N_{l-1}} w_{jk}^l a_k^{l-1} + b_j^l$$

$$a_j^l = f(z_j^l)$$

The first layer is the *input layer*, the rightmost layer is called *Output Layer* and all other layers in between are called *hidden layers*. *f(.)* is called the activation function.
The architecture shown above is typical of Feed Forward Neural Network (FFNN). Here

while the information is passed forward (feed forward) and backward (back propagation) between layers, it is never passed *within* the layer. That's where the Recurrent Neural Networks (RNNs) come in.

1.3.2 Recurrent Neural Networks and LSTM

The Feed-forward Neural Networks struggle with scenarios where the input nodes may be dependent on each other, for instance, time series data, language sentences, musical notes etc. RNNs address this issue by passing the current state of the node (called *hidden state*)as an input for the next iteration of the feed forward operation. This is indicated in Figure 1.2 below with curved lines around each node in hidden layers.

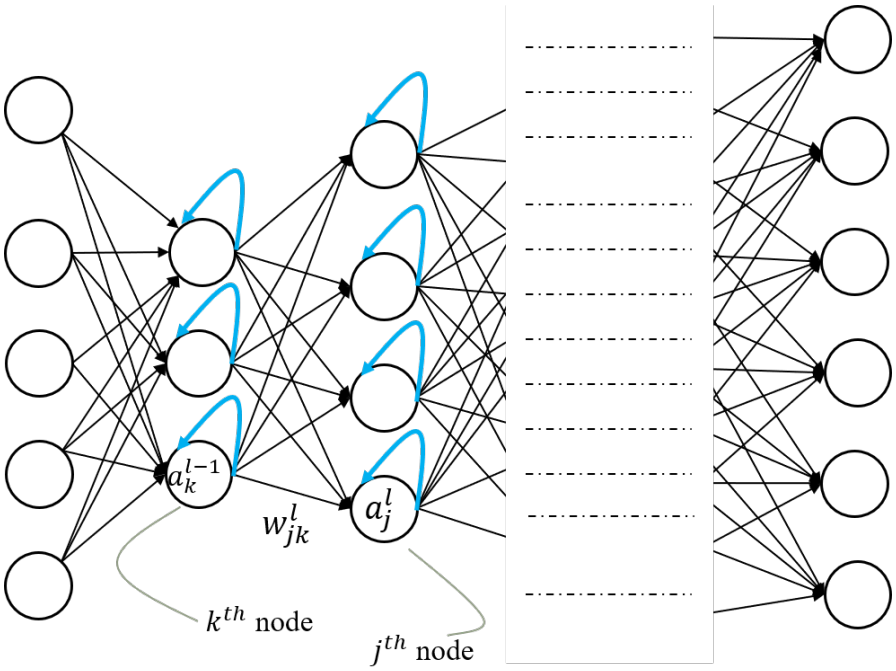


Figure 1.2: Recurrent Neural Network Architecture

Long Term Short Memory (LSTM) is a type of RNN that has proven to be quite efficient when it comes to sequential data. Typically, machine learning models use gradient descent to achieve an optimal solution. Gradients are partial derivatives of the loss function (optimization objective) in terms of model parameters. A model trains itself iteratively such that the combination of weights that produce the maximum gradient descent is the combination for maximum improvement in performance compared to the previous iteration. In the case of sequential data, the model parameters for RNNs are dependent over a large span of time. Usually, this leads to very small gradients or very large gradients for updating the prediction towards the optimal value, resulting in the network either taking excessively long to converge (called "vanishing gradient problem") or overshooting the optimal solution (called "exploding gradient problem"). LSTM addresses this by using a memory cell to decide which information to keep or forget based on the variation of gradients (Hochreiter and Schmidhuber 1997). This way excessively high or low gradients are dropped and learning proceeds normally.

1.3.2.1 Pre-training in Machine Learning

Pre-training is an approach where a model is first trained on large amounts of unlabeled data to help learn the general patterns in the data. It is then fine-tuned using a relatively small quantity of labeled data.
Auto-encoders, a type of Neural Network that works primarily by transforming input data into a different representation (called encoding) and using that to recreate the input (called decoding) are often used in the pre-training of time series data. (Yang et al. 2022) and (Hinton and R. R. Salakhutdinov 2006) provides a detailed review of Autoencoders' typical architecture, the training process and recent applications. (Sagheer and Kotb 2019) used a layer-wise pre-training with Autoencoders to generate initialization weights for the Deep LSTM architecture. They used this combined model, referred to as LSTM-based stacked Autoencoder (LSTM-SAE) for time series forecasting and presented improved results compared to the deep LSTM model.

1.3.3 Data Handling and over-fitting

Data is central to machine learning model development and evaluation and thus available data for Machine learning is often split into two parts: the part used for training and learning and another used for testing. Usually, the split is 80% training and 20% test but it also depends on the size of the data available and the complexity of the model (Xu and Goodacre 2018). While the test data can be any subset of available data, Training data should be representative of the problem the ML model is tasked with, and should generalize the problem well. In the case where this isn't met, the ML model may run into the problem of over-fitting.

1.3.3.1 Over-fitting

Every machine learning model have some model parameters which are iteratively updated as the model learns the patterns and relationships in the data. For a neural network, these are the weights that connect one node to another. The number of model parameters is decided by the number of layers and number of nodes in each layer. In many cases, it may happen that the available data is too small or lacking enough samples to cover all possible patterns in the real-time data. Alternatively, the model architecture is too complex/deep i.e., it has too many nodes and layers for it to train. In this case, the model may end up memorizing the data provided for training. As a result, while the model may achieve very high performance on training data, it might show poor performance with test data. This is often referred to as the model 'overfitting' the data (Ying 2019).
To catch and visualize over-fitting early on during the training, K-fold cross-validation is often used. This consists of splitting the data into K-parts such that during every iteration the model is trained on (K-1) parts and tested on the remaining part, which is referred to as the validation data for that given iteration. For example, for K=5, the data is divided into 5 equal parts in every epoch, 4 of which are used for training and 1 for validation (refer Figure 1.3). The model accuracy for the training data is monitored alongside the validation data. Since the validation data is unseen data for that given iteration, normally the validation accuracy is slightly worse than the training accuracy. If the model begins to over-fit the training data, the training loss/accuracy may still improve but the validation

loss/accuracy becomes stagnant.

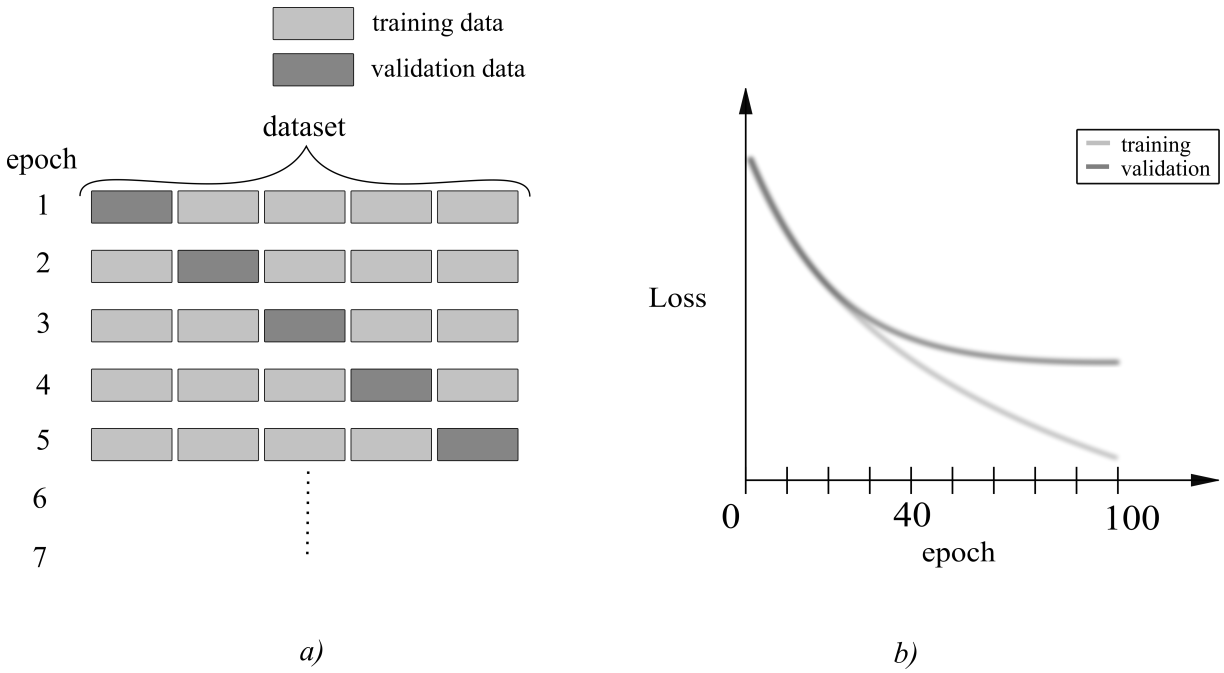


Figure 1.3: a) K-fold cross validation for K=5, and b) Loss variation vs epoch for training and validation data-set. Over-fitting initiates at 40 epochs

In figure 1.3, we see both the validation and training loss decreases as the model trains epoch by epoch. After around 40 epochs, the model doesn't learn any new patterns in the data, but rather memorizes the patterns specific to the training data.
There are many ways to address over-fitting in machine learning models, including regularization, dropout, data augmentation (which is discussed later), ensembling, feature selection, etc. See (Nusrat and Jang 2018) for an overview of methods to address over-fitting.

1.3.4 Data Balancing

Another important consideration while preparing the data for ML model development is data balancing. Real-world data can often be imbalanced in terms of the amount of data/number of samples available for each class used in the training. An example would be real-world driving data collection to study accident prevention. A data collection activity that mimics real-world driving would be biased heavily toward samples with normal driving, having very few recorded instances of the rare events. As a result, the ML model which is trained on such data may also show a similar bias (Haibo and Garcia 2009).
In order to avoid this behavior, the input data to the model must be balanced. There are various approaches to achieve data balancing. Some common approaches are listed below

- **Random under-sampling**: Here the samples in the majority are dropped to match the minority count. This may also lead to loss of information and under-fitting.
- **Random oversampling**: Here the samples in the minority are duplicated to match the majority count. As this may lead to repetitive data, the model may run into the risk of over-fitting.

- **Ensembling**: Ensembling involves combining various sub-models trained on balanced subsets of the data. These subsets could be balanced using a combination of randomly under-sampled and over-sampled data. This helps the model combine the learning from these sub-models into a single model.
- **Cost-sensitive Learning**: In this method, a custom loss function is created that can assign higher penalties to misclassification in minority classes.

The authors in (Haibo and Garcia 2009) have described these methods in detail.

1.3.5 Machine Learning Model Evaluation

Machine learning models are used to make predictions from a given set of input data and thus their performance is dependent on the data that has been used to generate the model. Evaluation of such models can help in accessing whether the model can be used for unseen data and further, whether it can be generalized to suit a wider range of applications. There are various ways to evaluate model performance, but in the context of this thesis, the following methods are identified to be important.

1.3.5.1 Confusion Matrix

When a machine learning model is tasked with classification, there is a possibility that the model may give an output that is different from what is expected. Many such cases are possible. For binary classification tasks, it usually takes the form of a 2X2 matrix This matrix, referred to as a Confusion Matrix for the given model, helps in evaluating the performance of a classification model by summarising predicted and actual class labels of the data set (Figure1.4).

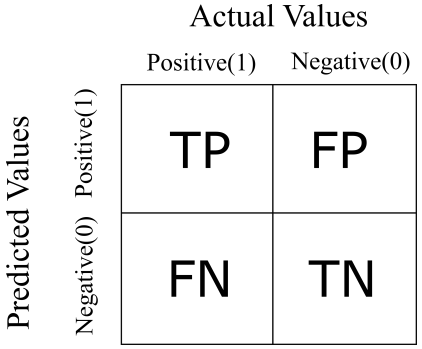


Figure 1.4: Confusion Matrix

To construct Confusion Matrix for a machine learning model, the model's predicted output is compared with the actual output and classified into the following four categories (note: this applies to a binary classifier. For multi-class Classifier, the matrix could be different but can be simplified into a 2X2 matrix)

- **True Positive (TP):** where the predicted output matches the actual output and both are positive (1).
- **True Negative (TN):** where the predicted output matches the actual output and both are negative (0).

- **False Positive (FP):** where the predicted output is positive (1) while the actual output is negative (0).
- **False Negative (FN):** where the predicted output is negative (0) while the actual output is positive (1).

A good model should maximize 'True' predictions and minimize 'False' predictions. The easiest way to quantify this is through accuracy, which means the total count of samples in the prediction that exactly matches the actual sample. In other words:

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)}$$

A more comprehensive understanding of the model can be achieved through the True Positive Rate (TPR) and False Positive Rate (FPR) metrics. True Positive rate indicates the total number of **true (1)** output predicted correctly out of the actual true output i.e,

$$TPR = \frac{TP}{(TP+FN)}$$

Similarly, FPR provides this information for the total number of **false (0)** output predicted correctly. It is defined as follows

$$FPR = \frac{FP}{(FP+TN)}$$

However, in certain applications minimizing False Negatives may be more important than False Positives, such as in the medical diagnosis of chronic diseases like breast cancer (Bleyer and Welch 2013) whereas in other applications, such as intrusion detection in computer systems (Abouabdalla et al. 2009), minimizing False Positives may be more important. In order to find the best model for any given application, ROC curves are often used.

1.3.5.2 ROC Curve

A machine learning model never outputs an absolute value in classification. It rather outputs a probability of the prediction falling under one of the prediction classes. For a binary classifier where the classes are 0 and 1, the model would output a value between 0 and 1 (e.g., 0.6846, 0.3447, etc). In order to convert these values to absolute classes, a threshold value can be chosen such that values lesser than this threshold are classified as 0, while those equal to or greater than the threshold are identified as 1. This threshold is called as **discrimination threshold**.

A receiver operating characteristics (ROC) curve is a graphical representation of model performance as this discrimination threshold is varied. ROC curves have been extensively used in medicine to evaluate the performance of new proposed diagnostic tests in terms of their ability to predict positive cases against negative cases. An example would be measuring the effectiveness of fluid pressure in a patient's eyes to diagnose Glaucoma. By varying the threshold pressure, one may see varying rates of positive and false diagnoses and a suitable value can then be chosen (Swets, Dawes, and Monahan 2000). ROC curves have been in use in Machine Learning since (Spackman 1989) used them to evaluate and compare various algorithms.

In cases where the benefit gained by a correct diagnosis/prediction is equivalent to the loss born by an incorrect diagnosis/prediction, the performance of the model can be quantified by the area under the ROC curve (AUC).

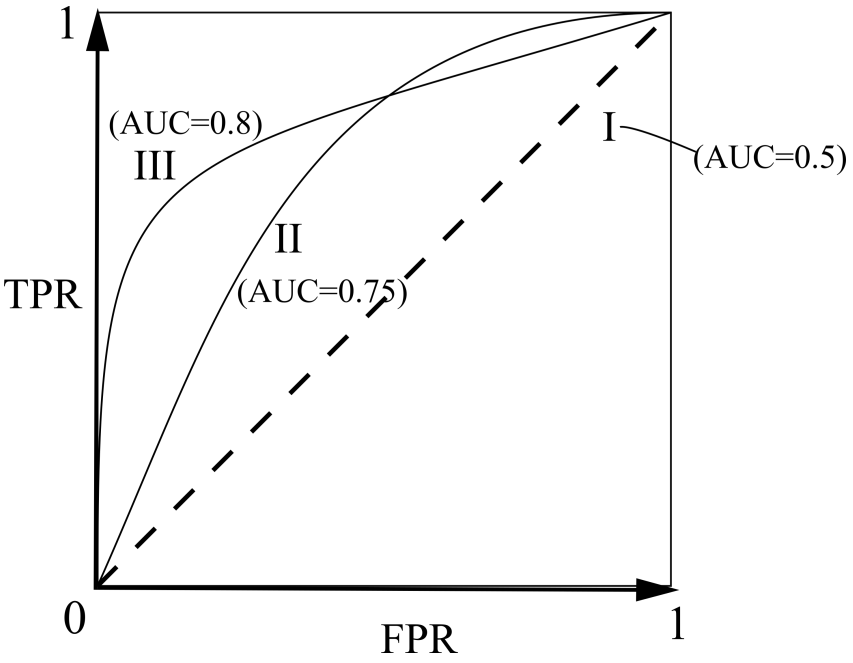


Figure 1.5: The ROC curves for three classifiers

The ROC curve for three different classifiers is shown in Figure1.5. Model I has an AUC of 0.5 computed by the area of the triangle. Here the change in the discrimination threshold does not affect the prediction, which is equivalent to using a flip of a coin. Model II is better with an AUC of 0.75, but worse than Model III which has an AUC of 0.8. However, model II would perform better on TPs than III since for the same FPR, II provides a better TPR than III. In the case where the benefit caused by a True Positive is larger than the drawback caused by a true negative, model II is a better choice for prediction. In all the other cases, model III is a better choice.

2 Method

This section describes the methodology employed to develop and assess a Neural Network architecture for an LKA system. It begins with a brief description of the collected data and pre-processing steps to transform the data into a form suitable for the ML model to train on. Next, the supervised learning and unsupervised learning methods (along with some pre-processing steps specific to each of the methods) are described. Finally, an overview of the analysis of model performance is presented leading to the results in the next section.

2.1 Data Pre-processing

The available driving data consists of drives across different regions in Europe, America, and Asia collected using a fleet of vehicles driven by professional drivers. The vehicles were equipped with a front-looking camera and other sensors to monitor the vehicle kinematics. All ADAS/AD systems inbuilt into the vehicles were disabled for the entire duration.
A drive is typically a few hours long. Each of these drives is further divided into **segments** that are 1-min long. The data is collected at 40 Hz, resulting in 2400 frames in each segment. Each of these segments can be further split into smaller **samples** based on a chosen time step. For example, a sample with 40 frames would correspond to a time step of 1 second. The collected data containing all the signals is fed into the prototype software and the software makes a decision whether to trigger or not. This process is called re-simulation and the output data of the intervention decision is called the re-simulated data from here on.

2.1.1 Labelled data extraction

For a given build version of sensor and system software, we have several **interventions events** from the system during the re-simulation, which are events where the LKA system is triggered. These events could be anywhere between 10 to 100 frames long. The data received from the forward-looking camera is processed by the perception system of the prototype software. An example of an perception-processed frame is shown in figure 2.1. The events are then manually annotated as TP, FP, TN and FN based on analysing the perception processed image. These labels are used for the evaluation of intervention in the re-simulated data. This is similar to the labels decribed in the confusion matrix in 1.3.5.1, except that it is applied to the re-simulated data instead of machine learning model. For this application:

- A True Positive (TP) is where the LKA system intervened as intended
- A False Positive (FP) is where the system intervened but it wasn't intended by the user.
- A False Negative (FN) is where the system should have intervened as per the user's intention, but it didn't.
- A True Negative (TN) is where neither the user intended for system intervention, nor did the system intervene.

The green lines in the figures shows lane markers as detected or perceived by the perception system. It can be seen in this specific case in figure 2.1 that the perception system detected the left lane marker to be offset towards the right relative to the actual lane marker position. This could be the reason for the false positive in this case. Similarly for all the events the perception processed image is analyzed and suitable labels are assigned.

Figure 2.1: Sample instance of a false positive: the system identified the left lane marker to be closer than it actually is

As far as this thesis is concerned two of the labels (TP, FP) are already available for all the events (around 4000). From this point onward, the term "labelled data" will be used to refer to this data-set that includes the labels (TP, FP) for all the events. There are several more events available where the annotations corresponding to any intervention event are missing. These are referred to as the "unlabelled data" in this thesis.
To add to the labelled data, an estimate for the scenario for a TN is made. Recall that a TN is where the system didn't intervene, nor was an intervention expected. This is equivalent to a typical driving scenario i.e., the car driving sufficiently far from the lane markers. Hence, from the unlabelled data, a subset where:

1. The distance between the center of the ego vehicle and lane markers is greater than 1.5 meters, and
2. There was no intervention

would be representative of a TN scenario.
There is no available data set which is annotated as FN by annotators. This means that the system was not able to classify and train on the FN scenarios.

2.1.2 Signal Selection and Analysis

The existing LKA system utilizes a subset of signals from an Electronic Control unit (ECU) that receives and processes all the data from the sensors. These signals can either take a value from a pre-defined set of values (e.g., Turn indicator, intervention direction, presence of snow, lane quality etc.) or a range of values (ego velocity, lane clothoid coefficients etc.). An LKA intervention is when some or all of these signals assume the values specified in the logic. For instance, LKA intervention is triggered when there

is no snow on the road, Lane Quality is sufficiently high (> 0.8, to give an example), ego velocity is above 40 km/h and lateral distance from the lane marker is below 0.5 meters. Please note that the values provided here are arbitrary and intended solely to help the reader comprehend that the LKA (Lane Keeping Assist) system operates only when certain conditions are satisfied.
In this work, twenty such signals have been considered for the Neural Network based model, which captures the majority of lane information. These signals are:

1. Eight coefficients, four each for left lane and right lane, of the clothoid curve that describes the shape of the lane in front of the ego vehicle a_l,a_r,b_l,b_r,c_l,c_r,d_l,d_r
2. lane selection confidence C_l,C_r
3. longitudinal distance from ego to the end of the first clothoid curve for each lane ld_l,ld_r
4. transition distance(for control transfer from driver to LKA and vice versa) for each lane td_l,td_r
5. ego vehicle data such as velocity, v_e, and acceleration, a_e
6. ego vehicle lane heading with respect to lane markers, θ

A LSTM model (section 1.3.2) is used for learning the relationship between the input signals and the output signal for intervention. The model expects an input to be a 3-D matrix of the dimension ***(number of samples, sequence length, no of features)***, where *number of samples* and *number of features* are equivalent to the drive samples and number of signals (which is 20) respectively, while *sequence length* is the number of frames considered in one sample or the length of the sequence fed into the model. Corresponding to this input, the model would return an output of shape ***(number of samples, sequence length, N)*** where N would be decided by the number of cells/nodes in the output layer. Since we require the model to predict only 1 signal value I.e., the intervention, N would be 1.

2.1.3 Normalization

Varying the range of different input signals may significantly affect the prediction accuracy of some machine learning methods, specifically Neural Networks. Thus, some sort of data normalization is necessary. Many of the input signals have both positive and negative values and thus it is required to normalize in such a way that sign is preserved. This can be achieved by using the maximum value out of the absolute value of the signal (i.e., *max(abs(signal values))* as the normalizing factor. The normalized signals, as a result, would lie in (-1,1). Normalization needs to be done before sampling since different samples may have different range of values.

2.2 Building the prediction Model

This section covers the steps taken to build the prediction model. This includes the various data augmentation techniques to increase the sample size for training and methods to utilize the unlabeled data for prediction.

2.2.1 Supervised Method

The simplest and most robust way to build an intervention prediction model from the processed data is through supervised learning. In supervised learning, the model is supplied with both the input signals and the output signals for training. This means that the model only can train on labeled data.

2.2.1.1 Sampling strategy

This section describes how samples are extracted from drive segments. Figure 2.2 shows one of the 20 selected signals (the process for all the other signals is identical) from a segment. The segment is in the form of a time series signal where an intervention event is observed within a certain time duration as shown. In the figure, the highlighted portion denotes the intervention event as described in section 2.1. We want to use the network to learn:

- what caused it to trigger i.e., what caused the beginning of an intervention event, and
- what caused it to turn off i.e., what caused the end of the intervention event

In order to train the network on these scenarios, we need to capture frames where the *transition* to the start of the intervention event occurred and transition from the intervention event to normal driving occurred(i.e., the end of the intervention event).

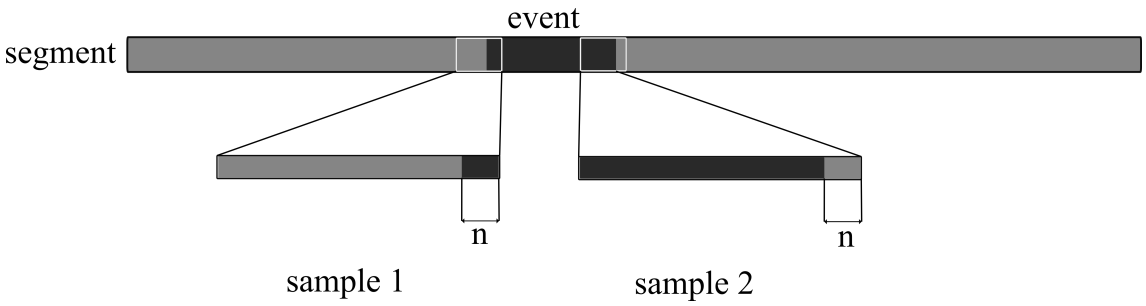


Figure 2.2: sample extraction from a drive segment

The data is thus captured in the following way: for the first sample ('sample 1' in Figure 2.2) the sampling is done so that the last ***n*** frames contains the events. For the second sample('sample 2' in Figure 2.2), the sampling is done so that all frames except the last ***n*** frames contain the event. This way we get 2 samples per event. Note that ***n*** is considered the same for both samples to reduce computational complexity. It is however an adjustable parameter.

2.2.1.2 Data Augmentation

Given that only 4000 segments were available for a deep neural network to train on, it would be prone to overfitting (see 1.3.3.1). In such cases, the existing data can be modified to generate additional data for training. This is called Data augmentation. In this thesis, the following types of augmentations have been used:

1. **Time-shift Augmentation:** This involves shifting the samples backwards in time by a few frames while retaining the sequence length. (Figure 2.3)

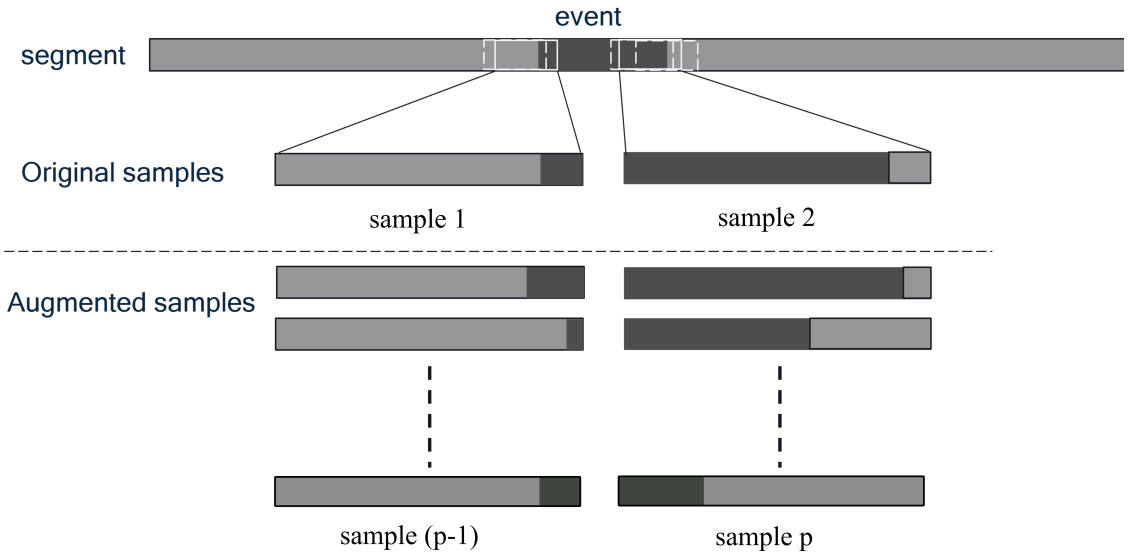


Figure 2.3: Time-shift augmentation on extracted signals

In this case, a total of 10 (p in the figure) augmented samples, 5 each close to intervention start and intervention end, were generated by shifting the samples randomly within 10 frames to the left (i.e., before in time). This increases the number of samples while also providing additional information on signal variation around the intervention time.

2. **"Flip-scene" Augmentation:** This involves modifying the scenario to simulate a mirrored case, which means the signals associated with the car in the lateral direction would be flipped, while those in the longitudinal direction remain the same (see Figure 2.4).

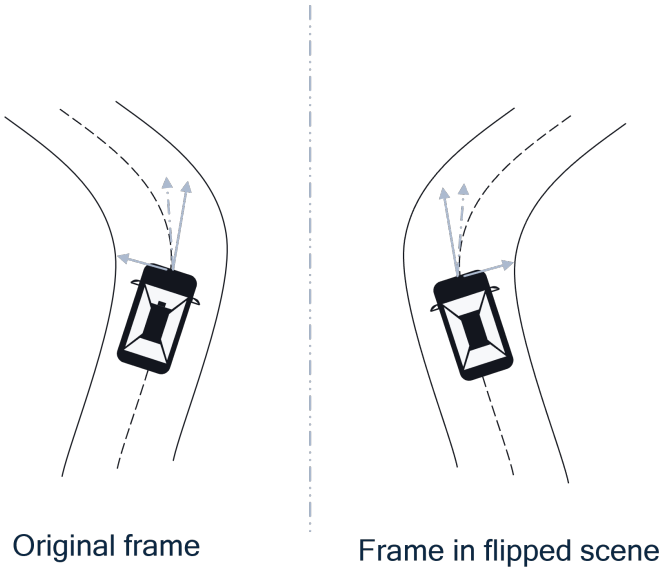


Figure 2.4: Flip-scene Augmentation: The input signals in longitudinal direction e.g., longitudinal velocity, would be same while those in lateral direction e.g., lateral velocity, would change its sign

The augmented samples are finally organized into an array and fed to the LSTM network for training.In summary, the supervised method can be described in the following steps (Figure 2.5):

A. Extract the chosen signals of interest from the datalogger and normalize them.
B. Sample the signals according to the sampling strategy in section 2.2.2.1.
C. Augment the samples using one or both of the techniques described in section 2.2.1.2.
D. Re-organize the signals in the form *(number of samples, sequence length, no of features)*.
E. Train the machine learning model using the inputs and outputs.

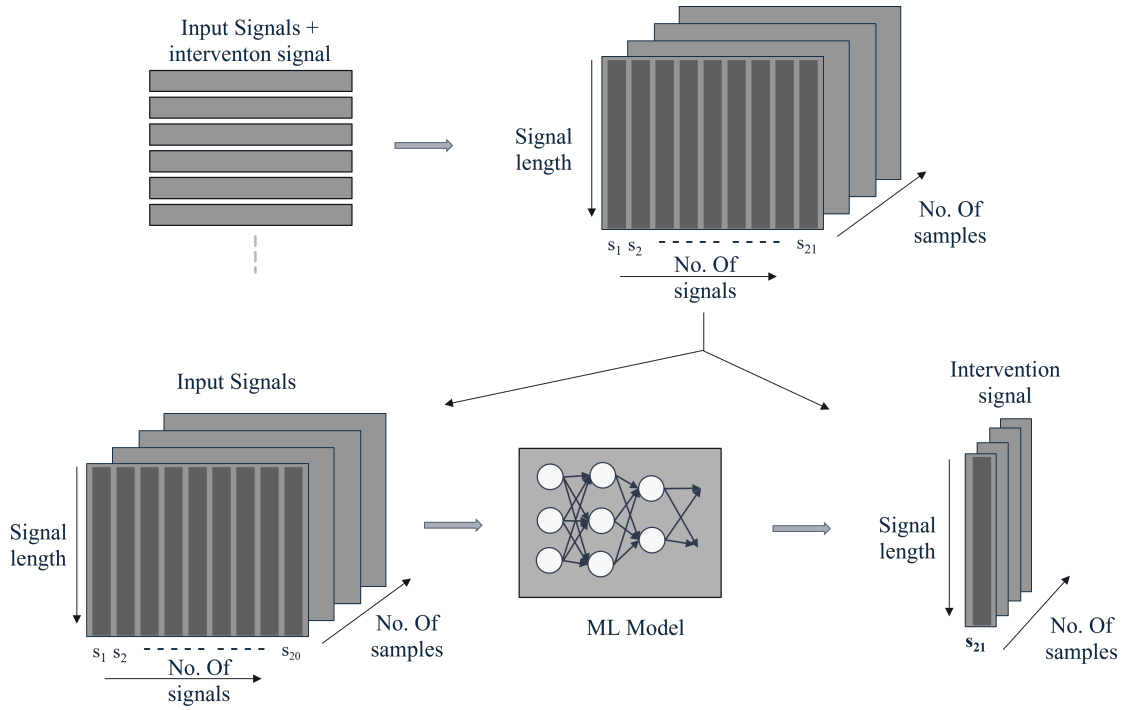


Figure 2.5: Supervised Method summarised

2.2.2 Unsupervised method: Training with unlabeled data

The annotations in the available data for model development is limited leading to limitations in supervised model accuracy. However, a lot of data is available where the events aren't classified into TP, FP, TN, and FN. In this case, the model can be 'pre-trained' over unlabeled data by training it over the existing signals (but then unrelated to the TP, FP, TN and FN). The trained model can then be saved and used for the supervised traning. That is, when the model is trained on the labeled data (supervised), the old model is loaded again with the weights frozen (i.e., they are not changed during the model training) with new layers added on top, which trains with labeled data. This can provide an improvement in prediction even with a low quantity of labeled data (Sagheer and Kotb 2019).

2.2.2.1 Sampling Strategy

Note that the events are not labeled in the data used for unsupervised learning. Thus the samples are generated by slicing the segment over uniform intervals (Figure 2.6). This mean that the unsupervised model is trained on the entire segment and not just the samples where the LKA system intervened.

Figure 2.6: sample extraction from a drive segment in unsupervised model

2.2.2.2 Pre-training

The following are two ways to pre-train the model used in this study to predict intervention from LKA:

1. **Signal masking:** After the input signals are normalized and sampled, a small part of each sample is chosen to be *masked*. As we know after normalization, all signals are limited in (-1,1) (see section 2.1.3), and thus in the *masked* part of the signal, the original values are replaced by values outside (-1,1). Here the masked signal values are replaced with -2 (used purely for illustration). This becomes the input.

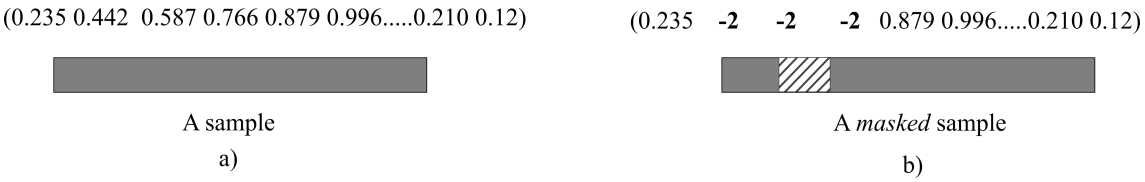


Figure 2.7: signal masking

A copy of unmasked data is stored as output. The model used here is LSTM too but with more nodes to learn more patterns in the data. The entire process is pictorially summarised in figure

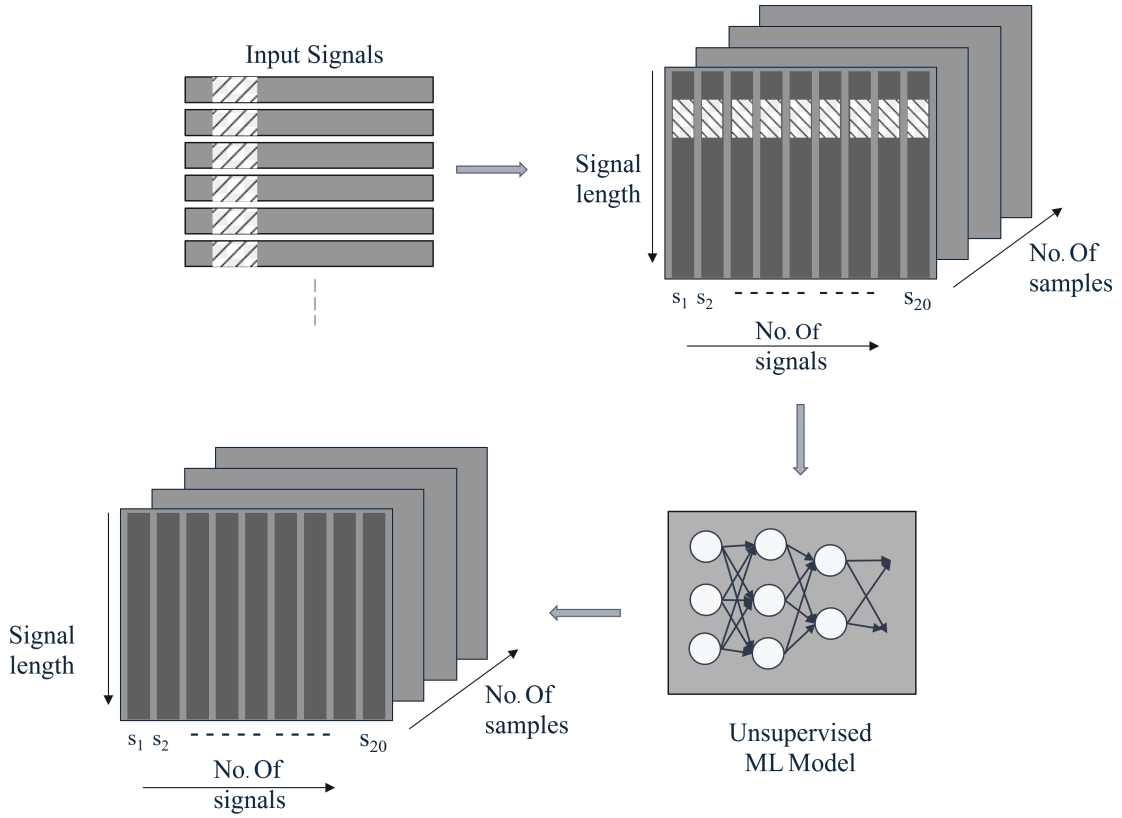


Figure 2.8: Unsupervised method with masking

2. **training with unlabeled intervention:** In the unlabeled segment, the output for intervention is present, but it isn't clear whether the intervention was TP or FP. In this case, all the interventions are considered TP and trained i.e., the existing system is considered the actual output on which the model will be trained. In other words, this method can be used to *mimic* the existing system and present its performance.

As explained earlier in this section, the unsupervised model is now used with weights already trained on the "type of data" it will use also for the supervised training. That is, the supervised training is done "on top" of the unsupervised model. A pictorial approximation is shown below.

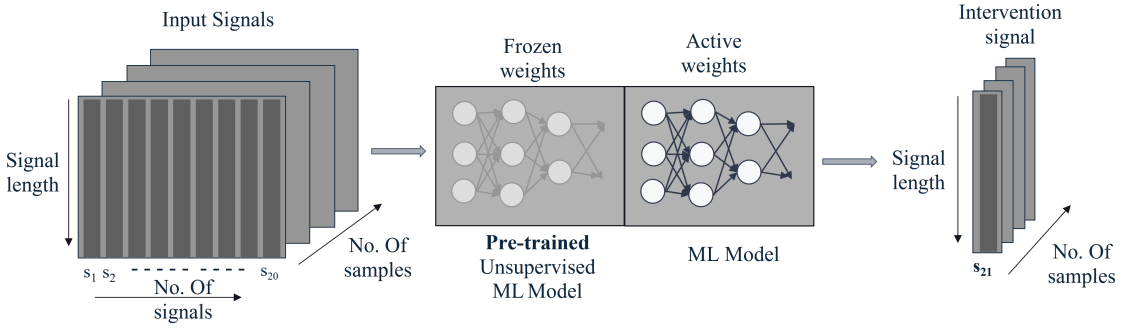


Figure 2.9: Supervised model with pre-training

For more information about this approach and successful applications of pre-training see (Srivastava, Mansimov, and R. Salakhutdinov 2016) and (Sagheer and Kotb 2019).

2.3 Analysis

In the section that follows, different machine learning and data processing methods are tried out and the model performance is evaluated to understand the sensitivity of each of these parameters on the output performance. The different choices and methods assessed in this thesis are:

- Sequence length
- Different data augmentation techniques :
  - Time shift augmentation.
  - Flip-scene augmentation.
  - Flip-scene augmentation with Time shift augmentation.
- Different machine training methods:
  - Only supervised training.
  - Only unsupervised training.
  - Unsupervised training followed by supervised training.

  The metrics that are used for evaluating system performance are:
  - Accuracy (refer 1.3.5.1)
  - FPR and TPR (refer 1.3.5.1)
  - ROC and AUC-ROC

3 Results

3.1 Effect of different machine learning and data processing methods

Different machine learning and data processing methods are tried out and the model performance is evaluated to understand the sensitivity of each of these parameters on the output performance.

3.1.1 Sequence length

The model is trained on samples with different number of frames. The ML performance is measured in terms of the accuracy of the predicted data and plotted in y-axis against different sequence lengths in x-axis as shown in Figure 3.1. Looking from the left to right in the figure 3.1,and examining the sequence length as it increases from left to right, a clear change in accuracy is evident around 60 frames, exhibiting a distinctive "knee" shape. It becomes apparent that beyond this point, there is no further improvement in model accuracy.Since there is no improvement in model accuracy for sequence lengths more than 60 frames it is suitable to choose 60 frames as the sequence length for further processing.

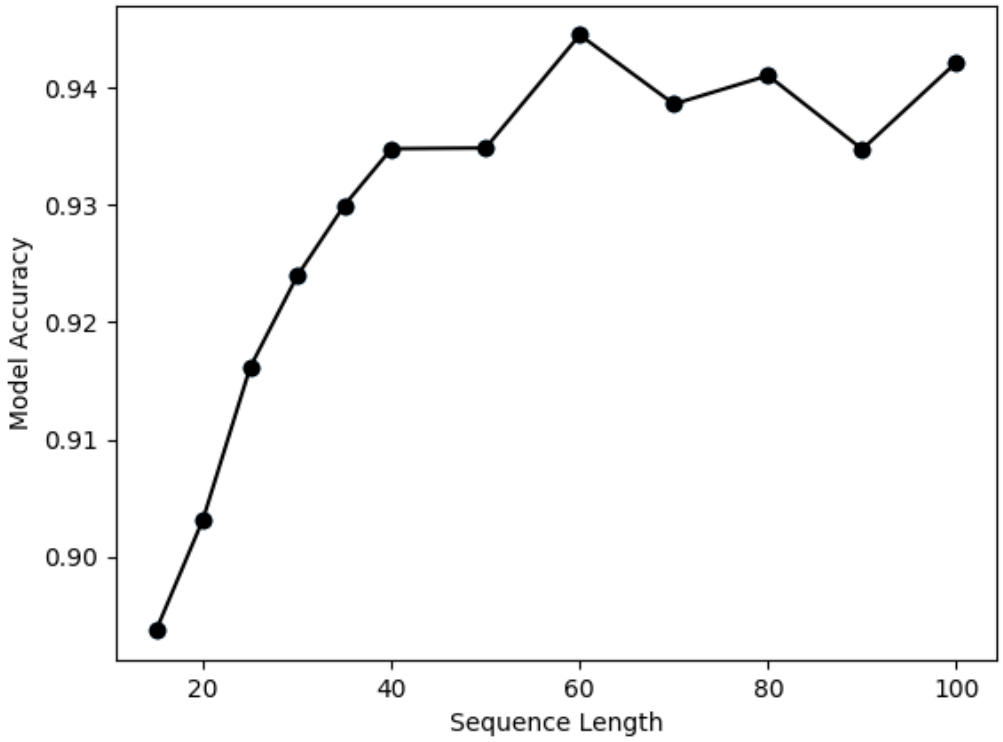


Figure 3.1: Effect of sequence length on the model performance

3.2 Effect of data augmentation

Figure 3.2 illustrates the prediction accuracy of the machine learning model on the test data, specifically for a sequence length of 60 frames. The y-axis represents the accuracy, and the figure displays different combinations of data augmentation methods and machine learning methods that were implemented.
The prediction accuracy is observed to be the highest for all the machine learning methods when time shift augmentation is performed. Hence, time shift augmentation method is chosen as the augmentation method for further processing.

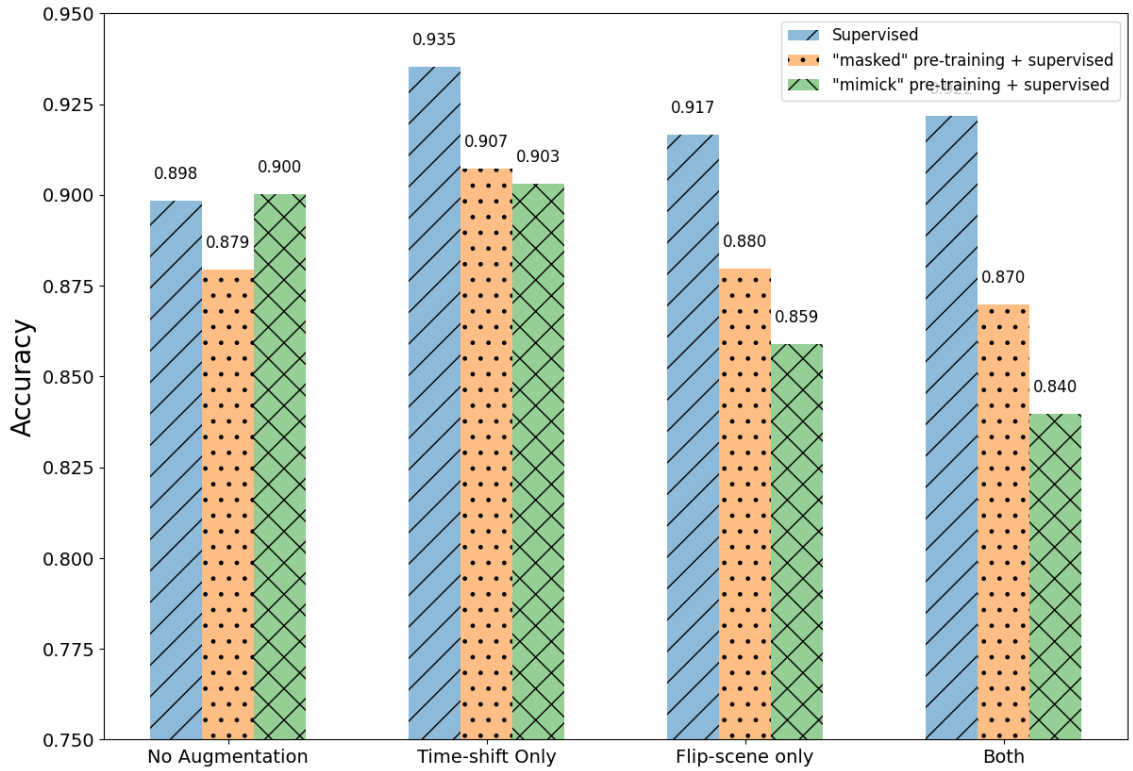


Figure 3.2: Effect of data augmentation techniques on the model performance

3.3 Overall system performance metrics

For the sequence length of 60 frames and augmentation technique of time shifting, the three different machine learning models (supervised learning, unsupervised learning, pre-training and then supervised learning) are assessed with a test data-set, and their performance is visualized and compared using the following metrics:

1. Area under curve of ROC (AUC-ROC)
2. Confusion matrix
3. False positive rate (FPR)
4. True positive rate (TPR)

3.3.1 AUC-ROC

The Area Under the Curve (AUC) in the Receiver Operating Characteristic (ROC) curve is a measure of the machine learning model's ability to classify LKA (Lane Keeping Assistance) intervention decisions, specifically between triggering (1) or not triggering (0). A higher AUC value indicates better classification capability. By considering all probability threshold values from 0 to 1, the ROC-AUC helps determine the best ML model for overall classification performance in accurately identifying when LKA intervention should be triggered or not. Figure 3.3 shows the ROC curve for the different machine learning models (supervised model, 'mask' pretraining+supervised model 'mimic' pretraining + supervised model) tested out with an optimal sequence length of 60 frames and the best augmentation technique of time shifting. As seen in the Figure 3.3 the supervised machine learning model achieves the highest AUC-ROC value of 0.98, indicating strong classification performance. Following that, the "pretraining + supervised" model obtains the next highest ROC-AUC value of 0.97. In comparison, the unsupervised model exhibits the lowest ROC-AUC value among all models, measuring 0.95.The blue dotted line represents a base model, a random classifier with no predictive ability. Details on ROC and AUC are described in 1.3.5.2

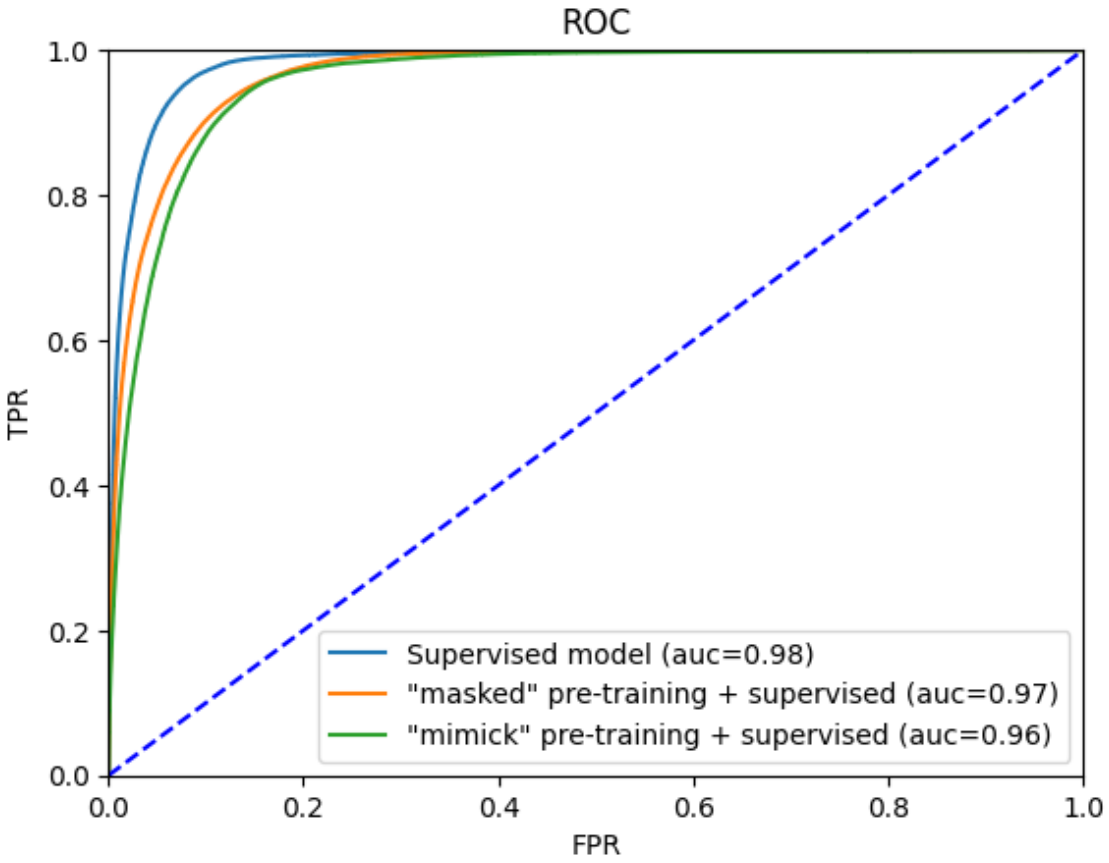


Figure 3.3: ROC curves for different models

3.3.2 FPR, TPR, Confusion Matrix

The True Positive Rate (TPR) and False Positive Rate (FPR) of the three ML-models (supervised, unsupervised, pre-training + supervised) for the sequence length of 60 frames and augmentation technique of time shifting are tabulated below.

Model	TPR	FPR
Supervised Model	0.884	0.051
(Pre-training + Supervised) Model	0.815	0.061
Unsupervised Model	0.789	0.068

Table 3.1: TPR and FPR values for the selected models

The best model, which is the supervised ML model, with sequence length of 60 frames and time shifted augmentation is tested with the test data and its prediction checked against the ground truth. The output of this comparison is shown in the confusion matrix 1.3.5.1 shows the number of instances of all the four possible outcomes: True positives, False positives, True negatives and False negatives. The results indicate nearly 75.11% predictions as TNs and 14.71% as TPs, giving a total accuracy of 93%. The TNs constitute the majority of the data, followed by TPs, TNs and lastly the FPs. More details on confusion matrix is provided in the section 1.3.5.1)

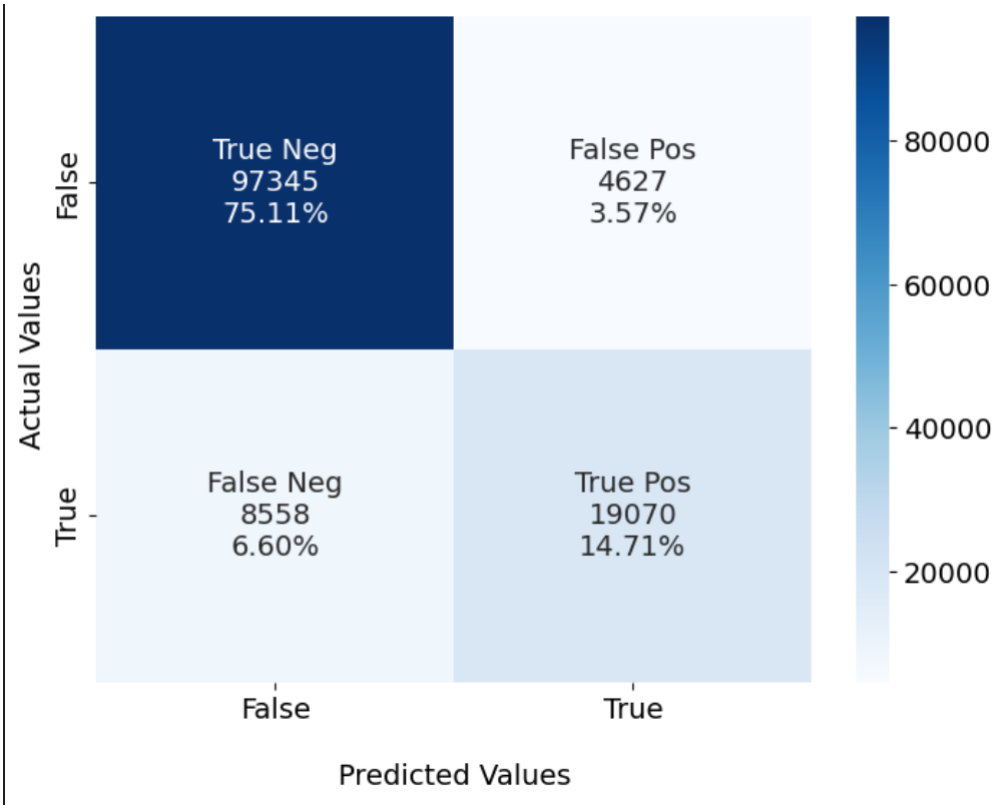


Figure 3.4: Obtained confusion matrix from the best model (only supervised)

4 Discussion

This thesis aims to explore various machine learning method components for a decision-making system of a LKA (Lane Keeping Assist) system. Different data processing techniques (such as varied sample size or sequence length, time shift augmentation of the data, flip scene augmentation of data) and machine learning models (such as supervised learning, unsupervised learning, pre-training with unlabeled data followed by supervised learning with labeled data) were examined.
The objective of this study was to identify the best method, among these combinations, based on the accuracy of intervention decision predictions. In other words, the goal was to determine how accurately each method predicted whether to trigger an intervention or not for a given set of 20 input signals in test data.
Initially, the basic model (a supervised ML model with no augmentation performed) was tested with different sequence lengths, ranging from 15 frames to 100 frames, to determine the optimal sequence length.
Subsequently, for the selected optimal sequence length, various machine learning models and data augmentation techniques were assessed. The performance of these combinations was assessed by measuring the prediction accuracy when tested with prepared test data. The objective was to identify the best method among these combinations based on the accuracy of intervention decision predictions on the prepared test data.

4.1 Contribution

In this section, we will outline the contributions we, as authors, hope that this thesis can make toward understanding the application of Neural Networks for modeling an LKA system. Our primary objective was to address the research questions we outlined at the beginning of this thesis. We will provide explanations of our inferences based on the results of our work to answer these research questions and thereby attempt to contribute to the existing knowledge in the field.

1. *Is it possible to achieve good prediction accuracy for LKA intervention using Neural Networks?*
   As observed in figure 3.1, LSTM-based Neural Networks have been able to provide high prediction accuracy with a limited number of input signals. It is however observed that the model has a high count for FPs and FNs (Figure 3.4). This could be attributed to a lack of sufficient data for training on FPs and FNs that can be explained as follows: for each of the annotations (TP, FP, TN, and FN), It can be said that there is a specific combination of input signals leading to a scenario corresponding to that annotation. The samples available from the training data mostly contains TP and TN scenario, with few FP scenarios and no FN scenarios. A model trained on this data would have limited classification capability for FP and FN scenarios. Such imbalance often leads to the model classifying more test scenarios into the majority class (i.e., the class/scenario dominantly present in the training data). This is explained in (Haibo and Garcia 2009). In our case, since the model has lim-

ited training on what to do in an FP scenario (where the expected outcome would be to not trigger an intervention), it may choose to trigger intervention (TP being the majority of the scenarios), resulting in a False intervention (FP). Similarly, in an FN scenario, where the right course of action is to trigger intervention, the model, having been majorly exposed to the TN scenario, would rather choose to not trigger, resulting in a False no-intervention (FN). If the data is balanced between all four scenarios, the FP and FN count could be lower.

The LKA feature is designed for enhanced driving comfort, indicating that it doesn't hold primary safety responsibility. The driver remains primarily accountable for vehicle control and lateral safety. This feature aims to further enhance the driver's lateral control experience.

The occurrence of an FP would cause driver annoyance and conflict with the core goal of providing driver comfort. Therefore, for this specific type of LKA feature, minimizing FP instances is the objective. While the overall aim is to reduce all false predictions, which include FPs and FNs, there's a greater emphasis on minimizing FPs for this particular LKA variant.

Additionally, the LKA system ensures driver engagement in lateral vehicle control through continuous checks and alerts for hands-on-wheel presence."Hands-on-wheel" is a system designed to ensure the driver's consistent engagement in driving activities by verifying the presence of their hands on the steering wheel through various mechanisms. These mechanisms include checking steering torque, steering angle, and touch sensing on the steering wheel. The paper by (Moreillon 2017) delves into hands-on-steering wheel detection and highlights its significance in ADAS systems of Level 2 and beyond. Due to the presence of such mechanisms, even if an opportunity is missed (FN), the risk is relatively mitigated as the driver remains available for vehicle control. Moreover, the LDW system consistently provides supplementary assistance, remaining vigilant in alerting the driver to respond, even in cases where no intervention prompt is triggered by an LKA system tailored to prioritize minimal FPs over low FNs. In contrast to the Automated Emergency Braking (AEB) feature, where a missed opportunity often leads to a collision, a missed opportunity in LKA does not inherently result in a collision event. The potential impact of a missed opportunity varies based on the circumstances, such as the external environment beyond the lane, such as the presence of obstacles like trees or the presence or absence of oncoming vehicles.

2. *Are there ways through which one can work with limited annotated data and still achieve good accuracy?*
   We could achieve an overall accuracy of about 90% in intervention prediction. That is, we predicted X based on Y. As we can see in **??** we could marginally increase the intervention prediction accuracy using data augmentation technique. But, as can be observed in Figure 3.2, this increase is marginal, from 0.898 (blue bar in 'No Augmentation' block) without any augmentation to 0.935 (blue bar in 'Time-shift Only' block) with time-shift augmentation. Hence in our study, it was not possible to increase the accuracy substantially using data augmentation techniques. We would need more data to be able to increase the accuracy substantially. As

inferred from the paper by (Dahl n.d.), a data set containing 10000 data points (labeled events) has previously been sufficient to achieve a good result. We currently are using only 3600 data points in our model. Additionally, as pointed out earlier, a balanced representation of all four scenarios is crucial to effectively train the model and enhance its performance in correctly identifying these specific events. However, the specifics on what distribution of TP, FP, TN, and FN in the resulting confusion matrix would qualify to be called good LKA performance is a part of the core intellectual property of the companies developing such systems and is hence not shared.

3. *What is the effect of different parameters such as sequence length, data augmentation, and different machine learning methods on the output model performance and which method is best to adopt?*
   Each parameter mentioned has a substantial impact on the prediction performance of the model. After evaluating different combinations, it was found that utilizing a supervised machine learning model with a sequence length of 60 frames, along with time-shifted augmentation, yields the highest prediction accuracy on the test data.

4.2 Effect of sequence length

From figure 3.1, the model's prediction accuracy on test data tends to increase as the sequence length is extended; however, this also leads to a reduction in the amount of sampled data. Consequently, there exists an optimal point where the model achieves the best overall prediction performance. In this study, it was determined that a sequence length of 60 frames yields the most accurate predictions for the specific problem at hand. Considering the sensor frequency of 40 Hz, 60 frames are equivalent to a time horizon of 1.5 seconds. This finding is supported by Dahl in (Dahl n.d.), who has noted this number to be around 1.25 seconds.
When the sequence length is increased beyond this optimal point, the number of available samples for training diminishes. This limitation arises due to the data extraction method employed, which requires a sufficient number of consecutive frames within the same segment surrounding the event of interest (LKA intervention).

4.3 Effect of Augmentation

Time-shift augmentation: The results in Figure 3.2 demonstrate that the use of time augmentation contributes to a slight improvement in accuracy. In theory, time shift augmentation should enhance intervention prediction accuracy by introducing temporal variations to augment the existing labeled data. These variations help the ML model become more resilient to temporal patterns, thereby enhancing its capability to identify distinct features or patterns in the data, as discussed in the paper by Zheng et al. (Zheng et al. 2014). The results shown in Figure 3.2 align with this expectation, illustrating an improved prediction accuracy when time shift augmentation is applied.
Flip-scene augmentation: The flip-scene augmentation has a negative effect on performance, which means the prediction accuracy of the ML model on test data decreases com-

pared to the base model when flip-scene augmentation is performed (the base model is the model with supervised learning without any data augmentation technique performed).
This could be caused by multiple reasons such as over-fitting, the creation of unrealistic scenes on flipping, and generalization. These issues are discussed in the papers (Keshavarzi et al. 2021) and (Sai Abhishek 2022)

4.4 Effect of different machine learning models

1. **Effect of masked pre-training:** Ideally, pre-training using unlabeled data combined with supervised learning of labeled data should improve accuracy compared to mere supervised learning, as explained in the paper by (Srivastava, Mansimov, and R. Salakhutdinov 2016). This is because, in an ideal case, pre-training helps the model understand the context of the environment, such as the sequence continuity of each of the 20 signals and the interrelation between these signals. However, as seen in Figure 3.3 and Figure 3.2, the model with masked pre-training exhibits the lowest intervention prediction accuracy compared to the mere supervised model.
   A possible reason for this discrepancy could be the difference in context between pre-training and supervised learning. During pre-training, the ML model predicts the missing parts (masked entries) of the input signals. In supervised learning, however, the model predicts the output signal (intervention side) based on the input signals (the 20 signals). This issue is known as domain mismatch, and further information about this can be found in the paper by (Ganin and Lempitsky 2015)
   Effect of Mimic Pre-training: Ideally, pre-training should improve accuracy, as explained in the paper by (Srivastava, Mansimov, and R. Salakhutdinov 2016) However, as observed in Figure 3.3 and Figure 3.2, the model with mimic pre-training exhibits lower intervention prediction accuracy compared to the mere supervised learning model, although it performs better than the model with masked pre-training.
   One possible reason for this lower performance could be over-regularization or overfitting. Since the unlabeled data significantly outnumbers the labeled data, the model as a whole may become overfitted to the unlabeled data. As a result, the learning from the supervised data might not be fully or appropriately utilized for predicting the intervention decision. For more information about this problem of overfitting during pre-training and its potential solutions, please see to the paper by (Pereyra et al. 2017)

4.5 Limitations and Future Work

The work is based on several assumptions. Firstly, we do not have any information related to the drivers. False positive is often subjective to the driver, hence the annotations would be more relevant if they are made based on the driver's information or feedback.In other words, after a data collection drive is done, the driver can be asked for instances where the system intervened and if it was an acceptable intervention - at least in larger field operational tests. Naturally, this would require that the data collection drives are conducted with the ADAS system enabled. Additionally information such as driver glances

from a driver monitoring system could also be added as an additional input signal to the ML model to predict an intervention output that is more adaptive to the driver. In both approaches, utilizing driver data would have facilitated the development of a machine learning model that has the potential to be more widely accepted by the driver in terms of the Lane Keeping Assist (LKA) feature.The paper (Davoli et al. 2020) discusses use of driver behaviour data for driver state estimation, driving intent understanding for ADAS feature development.
Secondly, the thesis has focused only on optimizing the performance using one type of neural network (LSTM). It is possible that other Machine Learning methods can produce even more accurate results, or reduce the computational effort.(add John Dahl reference and methods he used)
Finally, we had the labels only for True positive and False positive events. Since we didn't have any labels specifically indicating True negative or False negative data, we used an alternative approach to generate simplified True Negative events as described in 1.3.5.1. The implication of this, i.e., the lack of labeled data for TNs and FNs, would be that the model would have limited performance, similar to the impact of unbalanced data with respect to existing labels (TPs and FPs). A poor performance in predicting FNs would mean that system would be compromised in terms of safety. This is in contrast to that in FPs, which would mean poor acceptance by drivers.

4.6 Conclusion

This thesis aimed to model the threat assessment and decision-making of an LKA system through learning-based methods, specifically using a Neural Network. An LSTM network was trained as an LKA system based on 20 signals related to lane markers, road geometry, and ego vehicle kinematics.In this thesis, we investigate a range of data processing techniques and machine learning models, including both supervised learning models and pre-training models. The pre-training models involved training the model with unlabelled data and then transferring this learned knowledge to a supervised learning model that learned from labeled data. The objective was to identify the most accurate method for predicting intervention decisions based on 20 input signals. We initially tested a basic model without augmentation using different sequence lengths, then evaluated different machine learning models and augmentation techniques for the selected optimal sequence length. The goal was to determine the best method for intervention decision predictions on the prepared test data.
We determined that the optimal method, which combines a specific machine learning model and data augmentation technique, achieved a prediction accuracy of 93.5% for intervention decisions on the test data. This accuracy implies that out of the 100 cases provided, the ML model accurately predicted the correct outcome in 93 cases, compared to the the ground truth (manually annotated data).
Further, this ideal model consists of samples with sequence length of 60 frames (which is close to 1.5 seconds), used only time-shift type of augmentation, and uses only supervised learning without any pre-training. This can be further improved with annotations on FPs and FNs to generate a sufficient and balanced training set and data on driver glance behavior and reevaluated with other Machine Learning methods.

Bibliography

Abouabdalla, Omar et al. (2009). "False positive reduction in Intrusion Detection System: A survey". In: *2009 2nd IEEE International Conference on Broadband Network amp; Multimedia Technology*. DOI: 10.1109/icbnmt.2009.5348536.

Beglerovic, Halil et al. (2018). "Deep learning applied to scenario classification for Lane-keep-assist systems". In: *Applied Sciences* 8.12, p. 2590. DOI: 10.3390/app8122590.

Bleyer, Archie and H. Gilbert Welch (2013). "Effect of three decades of screening mammography on breast-cancer incidence". In: *Obstetrical amp; Gynecological Survey* 68.6, pp. 440–442. DOI: 10.1097/ogx.0b013e3182978e4a.

CarSim (n.d.). URL: https://www.carsim.com/products/carsim/.

ConsumerReports (Nov. 2019). URL: https://data.consumerreports.org/reports/consumer-perceptions-of-adas/.

Dahl, John (n.d.). "Machine learning based prediction models and threat detection for Lane Keeping Assistance". PhD thesis.

Davoli, Luca et al. (2020). "On Driver Behavior Recognition for Increased Safety: A Roadmap". In: *Safety* 6.4. ISSN: 2313-576X. DOI: 10.3390/safety6040055. URL: https://www.mdpi.com/2313-576X/6/4/55.

Dean, Morgan E. and Luke E. Riexinger (2022). "Estimating the real-world benefits of Lane Departure Warning and Lane Keeping Assist". In: *SAE Technical Paper Series*. DOI: 10.4271/2022-01-0816.

Erie Insurance (June 2020). URL: https://www.erieinsurance.com/news-room/press-releases/2020/car-safety-survey.

Ganin, Yaroslav and Victor Lempitsky (2015). *Unsupervised Domain Adaptation by Backpropagation*. arXiv: 1409.7495 [stat.ML].

Haibo, He and E.A. Garcia (2009). "Learning from Imbalanced Data". In: *IEEE Transactions on Knowledge and Data Engineering* 21.9, pp. 1263–1284. DOI: 10.1109/tkde.2008.239.

Hinton, G. E. and R. R. Salakhutdinov (2006). "Reducing the dimensionality of data with Neural Networks". In: *Science* 313.5786, pp. 504–507. DOI: 10.1126/science.1127647.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural Computation* 9.8, pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

Keshavarzi, Mohammad et al. (2021). *Contextual Scene Augmentation and Synthesis via GSACNet*. arXiv: 2103.15369 [cs.CV].

Marino, Riccardo et al. (2009). "A nested PID steering control for lane keeping in vision based Autonomous Vehicles". In: *2009 American Control Conference*. DOI: 10.1109/acc.2009.5160343.

McDonald, Anthony D. et al. (2013). "Steering in a random forest". In: *Human Factors: The Journal of the Human Factors and Ergonomics Society* 56.5, pp. 986–998. DOI: 10.1177/0018720813515272.

Moreillon, Maxime (2017). "HIGHLY AUTOMATED DRIVING – Detection of the driver's hand on and off the steering wheel for ADAS and autonomous driving". In: *7th International Munich Chassis Symposium 2016*. Ed. by Prof. Dr. Peter E. Pfeffer. Wiesbaden: Springer Fachmedien Wiesbaden, pp. 505–525. ISBN: 978-3-658-14219-3.

NHTSA estimates for first nine months of 2022 suggest roadway fatalities beginning to level off after two years of dramatic increases (Jan. 2023). URL: https://www.nhtsa.gov/press-releases/nhtsa-estimates-traffic-deaths-2022-third-quarter.

Nusrat, Ismoilov and Sung-Bong Jang (2018). "A comparison of regularization techniques in deep neural networks". In: *Symmetry* 10.11, p. 648. DOI: 10.3390/sym10110648.

Palm, G. (1986). "Warren McCulloch and Walter Pitts: A logical calculus of the ideas immanent in nervous activity". In: *Brain Theory*, pp. 229–230. DOI: 10.1007/978-3-642-70911-1_14.

Pereyra, Gabriel et al. (2017). *Regularizing Neural Networks by Penalizing Confident Output Distributions*. arXiv: 1701.06548 [cs.NE].

research note: 2016 fatal motor vehicle crashes: overview - transportation (Oct. 2017). URL: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries.

Ritchie, Hannah, Fiona Spooner, and Max Roser (Feb. 2018). *Causes of death*. URL: https://ourworldindata.org/causes-of-death.

Road traffic injuries (June 2022). URL: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries.

Rosenblatt, F. (1958). "The Perceptron: A probabilistic model for information storage and organization in the brain." In: *Psychological Review* 65.6, pp. 386–408. DOI: 10.1037/h0042519.

Sagheer, Alaa and Mostafa Kotb (2019). "Unsupervised pre-training of a deep LSTM-based stacked Autoencoder for multivariate time series forecasting problems". In: *Scientific Reports* 9.1. DOI: 10.1038/s41598-019-55320-6.

Sai Abhishek, Allena Venkata (May 2022). "Augmentation Techniques Analysis with Removal of Class Imbalance Using PyTorch for Intel Scene Dataset". In: 8.

Spackman, Kent A. (1989). "Signal detection theory: Valuable tools for evaluating inductive learning". In: *Proceedings of the Sixth International Workshop on Machine Learning*, pp. 160–163. DOI: 10.1016/b978-1-55860-036-2.50047-3.

Srivastava, Nitish, Elman Mansimov, and Ruslan Salakhutdinov (2016). *Unsupervised Learning of Video Representations using LSTMs*. arXiv: 1502.04681 [cs.LG].

Sternlund, Simon et al. (2017). "The effectiveness of Lane Departure Warning Systems—a reduction in real-world passenger car injury crashes". In: *Traffic Injury Prevention* 18.2, pp. 225–229. DOI: 10.1080/15389588.2016.1230672.

Swets, John A., Robyn M. Dawes, and John Monahan (2000). "Better decisions through science". In: *Scientific American* 283.4, pp. 82–87. DOI: 10.1038/scientificamerican1000-82.

Tan, Dongkui, Wuwei Chen, and Hongbo Wang (Jan. 2017). "On the use of Monte-Carlo simulation and Deep Fourier neural network in Lane departure warning". In: *IEEE Intelligent Transportation Systems Magazine* 9.4, pp. 76–90. DOI: 10.1109/mits.2017.2743204.

WHO (2018). *Global status report on road safety 2018*. World Health Organization.

Xu, Yun and Royston Goodacre (2018). "On splitting training and validation set: A comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning". In: *Journal of Analysis and Testing* 2.3, pp. 249–262. DOI: 10.1007/s41664-018-0068-2.

Yang, Zheng et al. (2022). "Autoencoder-based representation learning and its application in Intelligent fault diagnosis: A review". In: *Measurement* 189, p. 110460. DOI: 10.1016/j.measurement.2021.110460.

Ying, Xue (2019). "An overview of overfitting and its solutions". In: *Journal of Physics: Conference Series* 1168, p. 022022. DOI: 10.1088/1742-6596/1168/2/022022.

Zheng, Y. et al. (Jan. 2014). "Time series classification using multi-channels deep convolutional neural networks". In: *WAIM 2014. LNCS* 8485, pp. 298–310.

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY