Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Research into Offensive Language Interpretation and Artificial Intelligence using Neural Networks

by
Lee James Rodger

Supervisor: Dr Tim Bashford
Second Marker: Gordon Dickers

Project submitted as part of the requirements for the award of
MSc Software Engineering and Web Development

January 2020

DECLARATION

I, LEE JAMES RODGER declare that I am the sole author of this Project; that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

Signed : LEE RODGER

Date : 27/01/2020

FORM OF CONSENT

I LEE JAMES RODGER hereby consent that my Project, submitted in candidature for the *MSc.Software Engineering and Web Development* degree, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations.

Signed : LEE RODGER

Date : 27/01/2020

RESEARCH INTO OFFENSIVE LANGUAGE INTERPRETATION AND ARTIFICIAL INTELLIGENCE USING NEURAL NETWORKS

A Viable Solution for Detecting Offensive Vehicle Registrations

Abstract

Where possible, we strive to find computing solutions to manual everyday tasks that we as humans have to carry out. Vehicle registrations can contain cryptic messages that we are able to decipher and interpret as words in our everyday language. The research carried out in this document aims to introduce machine learning solutions to an area of industry which relies on manual intervention and human perception Whilst looking at machine learning, it is important that we uncover the interpretation of offensive language amongst people of different ages.

Lee Rodger

1809707

Table of Contents

Table of Figures 5

Table of Tables, Graphs and Data 5

1 – Introduction *6*

1.1 – Project Aim 6

1.2 – Objectives 7

1.3 – Null Hypothesis 7

2 – Evolution of Language *8*

2.1 – Offensive Language 8

2.2 – Offensive Language in the Modern World 8

2.3 – Types of Offensive Language 8

2.4 – Geographical Impact 8

3 – Categorisation and Classification *10*

3.1 – Categorising Offensive Language 11

4 – Machine Learning *12*

4.1 – Supervised Learning 12

4.2 – Unsupervised Learning 13

4.3 – Neural Networks 13

4.3.1 – Real World Applications of Neural Networks 14

4.4 – Applying AI Ideas 14

4.5 – Loss Functions 14

4.5.1 – Mean Squared Error 15

4.5.2 – Binary Cross-entropy 15

4.5.3 – Categorical Cross-entropy 16

5 – Neural Network Solutions *17*

5.1 – Software Package Solutions 17

5.1.2 – Neural Designer 17

5.1.3 – Darknet 17

5.1.4 – TensorFlow 18

5.2 – Chosen Implementation 18

5.3 – Dataset construction 18

6 – Research Methods *19*

6.1 – Strategy and Approach 19

6.2 – Risk Analysis 19

6.3 – Data Evaluation 20

7 – Software Development Methods *21*

8 – Research Survey Results and Analysis *22*

8.1 – Initial Results 22

8.2 – Data Analysis **23**
8.2.1 – P Value 26
8.3 – User Study Conclusions **26**
9 – Software Implementation ***27***
9.1 – Constructing the Initial Classifier **27**
9.2 – Creating the Train/Test Data **27**
9.2.1 – PDF Data Abstraction 28
9.2.2 – Creating Non-offensive Vehicle Registrations 28
9.2.3 – Generating JSON 29
9.2.4 – Creating a Mixed Data CSV for TensorFlow 29
9.3 – Using TensorFlow for Binary Classification **30**
10 – Software Implementation Results ***31***
11 – Summary and Conclusion ***32***
References ***33***

Table of Figures

Figure 1 - Taxonomies used in basic object research - taken from Roach and Lloyd [9]. 10
Figure 2 - Offensive language categories using taxonomies. Categories used by Ipsos [4]. ...11
Figure 3 - An example of a decision tree taken from Rokach and Maimon [3] 12
Figure 4 - An example of data after a k-means clustering algorithm has been applied. Taken from Dey [2]. 13
Figure 5 - An example of a mean squared error loss function. Taken from Godoy [14]. 15
Figure 6 - An example of binary cross-entropy, taken from Godoy [14]].....15
Figure 7 - An example of categorical cross-entropy, taken from Godoy [14]. 16
Figure 8 - Software Development Lifecycle taken from Synotive - https://www.synotive.com/blog/software-development-client-questionnaire 21
Figure 9 - RegEx used to find vehicle registrations in strings 28
Figure 10 - A Switch statement used for giving a random vehicle registration character given the current character index 28
Figure 11 - Java code to print out data in a JSON format 29
Figure 12 - Code snippet producing the CSV for mixed registration data 29
Figure 13 - Python code snippet which loads the IMDb movie review data and initialises a train/test split using the TensorFlow Datasets library. 30
Figure 14 - Python code snippet setting the optimiser algorithm, lost function and result metrics. 30

Table of Tables, Graphs and Data

Table 1 - Initial results from the user study to detect if there is a trend between age and offesnive language interpretation in vehicle registrations.....22
Table 2 - A bar chart visualising the offensive vehicle registrations survey data.....23
Table 3 - Number of positive responses for offensive language per age bracket for the vehicle registration PR11 CKK 24
Table 4 - Number of positive responses for offensive language per age bracket for the vehicle registration FA16 CHB.....24
Table 5 - Number of positive responses for offensive language per age bracket for the vehicle registration MU19 DER 25
Table 6 - Loss and Accuracy results from initial use of TensorFlow. 31

1 – Introduction

In modern day computing, artificial intelligence plays a key role in many people's day to day lives. We can find many examples of AI from autonomous vehicles, to electric toothbrushes as demonstrated in an article by Vincent James [1]. We are progressing into a world where we use the power of computing to solve many of our day-to-day problems as well as revolutionise the industry with new technologies.

The aim of his project is to research and investigate machine learning through neural networks and apply these ideas to a system for detecting and discarding offensive vehicle registration plates. This area of the DVLA still relies heavily on human intervention to decide upon what variations of alpha-numeric characters are valid for use, which becomes as issue each time a customer requests a customised vehicle registration.

By the end of the study, we aim to find that a system such as the one described can indeed be beneficial to the industry. It is hoped that by finding a successful outcome, we can start to implement these ideas into many other areas within industry to alleviate the risk of human error in many possible tasks.

Firstly, we must establish what exactly we mean by an offensive term. Language has evolved, and still us evolving each day. The way in which we determine certain words and phrases can range drastically depending on the language, culture and geographic region. Since the way we interpret language evolves, we need to develop a system which can evolve with us, in this case, a neural network.

It is important not only that we have data which can be used to train and test the neural network, but it is far more beneficial if the data is accurate based on real human interpretation. From a user study, we can determine which registration combinations could be deemed offensive and use these the test and train the neural network.

We will be able to call the implementation a success if the system is able to correctly identify registration plates deemed already to be offensive. Over time, the system should be able to adapt and learn new patterns and become better at identifying such plates.

1.1 – Project Aim

This project overall aims develop a system for detecting vehicle registration plates, which can be deemed offensive. This takes into account situations where customers attempt to use numbers to replace letters, creating a word/phrase which we as humans can interpret to have a different meaning. The system should incorporate machine learning techniques and utilise a neural network in order to analyse the contents of the plate. By definition of a neural network, the application should be able to generate more accurate results throughout extended use.

Prior to this, we aim to understand the way in which people interpret potentially offensive language. It is important that no registration which could be offensive is used, therefore we must conduct research using different demographics as a way of getting a broader idea of what we can and can't allow when it comes to personalised vehicle registrations.

1.2 – Objectives

The aim has been outlined in order to determine the success of the project. The area of machine learning and artificial intelligence is extremely broad, therefore, to avoid losing focus on the ultimate goal, a number of objectives have been set to work towards the final goal.

The four main objectives of this project are as follows:

- Conduct a user study to create test data, containing offensive and non-offensive vehicle registrations, based on human interpretation.
- Create training data containing many offensive and non-offensive variations.
- Develop a binary-classification neural network.
- Conduct a number of tests with both offensive and non-offensive data to determine the accuracy of the network.

To determine whether or not a neural network implementation of an offensive detection system is viable, we need an application by which we can begin to test these ideas. Firstly, an application which can generate both offensive and clean (non-offensive) data should be created.

After this, we can begin to implement a neural network to see the results the application gives, given a large amount of data. The network should be able to analyse each registration we give it as test data and produce a binary result (offensive or not).

Once implemented, we should be able to see an accuracy rating for whichever algorithm is used for the implementation. It is important to test the network using a number of different algorithms to determine which gives the best results overall. We should hope to see an accuracy rating of 80% or higher to deem the network as working to a good enough standard.

1.3 – Null Hypothesis

As part of this research, we aim to look into the interpretation of offensive language in the context of vehicle registrations. Our null hypothesis for this research is as follows:

Age has no effect on the interpretation of offensive words/phrases contained within vehicle registrations.

2 – Evolution of Language

Language and how we use it plays an important component in understanding how we can categorise terms as offensive. The way we use language rapidly evolves over time and therefore the way in which we determine how offensive certain words are relies heavily on the current use and interpretation of certain words as well as where they are being used geographically. The origins of language are also an important area to look into as this gives us insight into where certain terms became offensive and how they have evolved over time. According to Mufwene [2], just as we as humans have evolved, our use of language has moved with us over time.

2.1 – Offensive Language

Throughout the course of mankind, we have evolved to deem certain terms as offensive and therefore we culturally make rules about how certain language can be used. Hagen acknowledges that swearing is "generally regarded as offensive" and that we as humans choose to use offensive language for a number of different reasons [3]. It is suggested that swearing has effects on the neurological and physical systems within humans and a lot of offensive language use occurs depending on the current cultural context. Some people are raised with stricter rules around the use of swear words, whilst others are raised under more relaxed views. Members from both groups will likely both deem the same words to be regarded as 'offensive', however, the level of offence to which members of either group would take from the use of a word would differ severely.

2.2 – Offensive Language in the Modern World

Published in a guide by Ipsos [4], there are a number of highly offensive terms which are deemed unsuitable for almost all circumstances, especially in a professional work environment. This also explains however that context is important, and considerations should be made for individuals from different cultural backgrounds, where certain terms which may be completely harmless in some regions are deemed to be some of the most highly offensive terms someone could use. It is important when assessing terms which are offensive to consider terms from all corners of the globe, as to avoid accidently offending someone due to a cultural misunderstanding.

A system which could theoretically evolve its understanding of offensive language long with changes in language itself would be ideal for a world which moves as rapidly as ours. Artificial intelligence has the capacity to allow such a technology to exist and find a place in our world where this technology could be heavily utilised.

2.3 – Types of Offensive Language

We can place different offensive terms into a number of categories. The category usually relates to the context in which a word is being used or gives a context for when a word may be deemed offensive. To do this, it is first important to understand how we categorise data and what attributes allow us to place certain data into different categories.

2.4 – Geographical Impact

Certain terms may only become offensive when put into a specific context, such as geographical. The place a person lives, or their nationality may have a drastic impact on

whether a term is deemed offensive or not. For example, the term 'Nazi' refers to a certain political regime i.e. that of Germany under the rule of Hitler. Although the word alone is not offensive, when used under a certain context, it can cause great offense. 'Nazi' can be deemed a racist term, especially when used to describe someone from Germany, and therefore would need to be blacklisted from any system which allows for personalisation, so as to not cause offense to anybody.

In an article by James Harbeck [5], he discusses the differences within offensive language in different parts of the world. He says, *"You may utter the name of the male or female organ when irritated in China or Russia; in Italy…the word for the female organ is usually the more forbidden one".* This indicates how different attitudes exist towards language centred around sexual organs. In Italy, it is considered far more 'offensive' whereas this is not typically even a consideration within Russia or China. It is important to be aware of cultural differences when assessing offensive language.

3 – Categorisation and Classification

By the nature of the goals we would like to achieve throughout this research, it is important to have an insight into how we classify data. We will largely be working with simple groups, in this case, registration numbers which are not offensive, and those which are. But it is the content of these registrations which decides upon how we classify this data.

Firstly, we need to understand which characteristics of a registration number deem it to be offensive. Roach and Lloyd discuss the concept of taxonomies within basic object research [6]. Categories can have multiple levels of categorisation. Below is an example taken from their paper on cognition and categorisation [6].

Superordinate	*Basic Level*	*Subordinate*
Furniture	Chair	Kitchen chair
		Living-room chair
	Table	Kitchen table
		Dining-room table
	Lamp	Floor lamp
		Desk lamp
tree	Oak	White oak
		Red oak
	Maple	Silver maple
		Sugar maple
	Birch	River birch
		White birch

Figure 1 - Taxonomies used in basic object research - taken from Roach and Lloyd [9].

From this example, we can see how basic information can be split down into multiple levels, in this case, three. The top level, 'superordinate', defines the highest level and most generic description for a category. Below that we can see the 'basic level' which contains more detail but still represent a category each. Finally, we have the lowest, 'subordinate' level which is the highest amount of detail we hold for the given categories.

This idea can also be represented using a tree structure, which clearly illustrates the levels in the hierarchy, demonstrating a similar idea to inheritance. Each category contains more detailed information the further down the tree is traversed, giving us a structure, which is very easy to interpret.

3.1 – Categorising Offensive Language

By using the methodologies discussed above, we can begin to break down offensive language into relevant categories to give context to why or when a word is deemed offensive.

The report by Ipsos [4] breaks down swear words into a multitude of different categories. We can begin by splitting words into either discriminatory or non-discriminatory contexts. These two categories cover all of the offensive language which exists within the world today. However, a word which appears in the discriminatory category may only be offensive in a more specific context and therefore more specific categories would need to be defined.

Superordinate	**Basic Level**	**Subordinate**
Offensive Words/Phrases	Non-discriminatory	General swear words/body parts
		Sexual references
		Offensive gestures
	Discriminatory	Older people
		Religion
		Sexual orientation and gender identity
		Mental health and physical ability
		Race and Ethnicity

Figure 2 - Offensive language categories using taxonomies. Categories used by Ipsos [4].

By breaking down offensive language into appropriate categories, it becomes clear that there is a vast amount of considerations to be made when it comes to detecting offensive words/phrases. This is an important consideration to make as we want to eliminate the risk of offending anybody by missing out some forms of offensive language when it comes to looking at which terms need to be eliminated, it must be ensured that all areas are covered.

According to Ipsos [4], context is of key importance when it comes to interpreting offensive language, and therefore we must take into consideration all of the applications in which a word may be used. As previously discussed, certain terms which aren't in themselves offensive, could be used in an offensive way.

4 – Machine Learning

Dey describes machine learning as a means to teach machines how to handle data more effectively [1]. In his paper, he describes a number of different machine learning techniques, each of which can be attributed to a learning style. The two main categories of learning are labelled 'supervised' and 'unsupervised', however there are a vat variety of learning categories which all hold valuable techniques for learning with different types of data.

4.1 – Supervised Learning

This approach to machine learning always requires human intervention at some level. The algorithms must be given datasets for 'train' and 'test' data. The train dataset contains an output variable which needs to be predicted [1]. A common implementation of a supervised learning algorithm is a decision tree. These trees work by grouping attributes and then sorting each one based on its value.

Rokach and Maimon describe decision trees as "a classifier expressed as a recursive pattern of the instance space" [7]. Data flows through the decision tree to reach an eventual end-node, whereby a 'decision' has been made based on the data.

An example of such tree can be seen below in *Figure 1*. This tree looks at the data given and checks it against three scenarios. The first step checks the age of the subject within the data, if greater than 30, the traversal of the tree ends there. If not, the gender is then checked. If female, the final check is carried out, 'Last R'. This is a simple model of a supervised form of machine learning which constantly needs to be fed data and is built upon decisions made by humans beforehand. The tree essentially sorts the data it is given and produces a result based upon pre-determined attribute values.

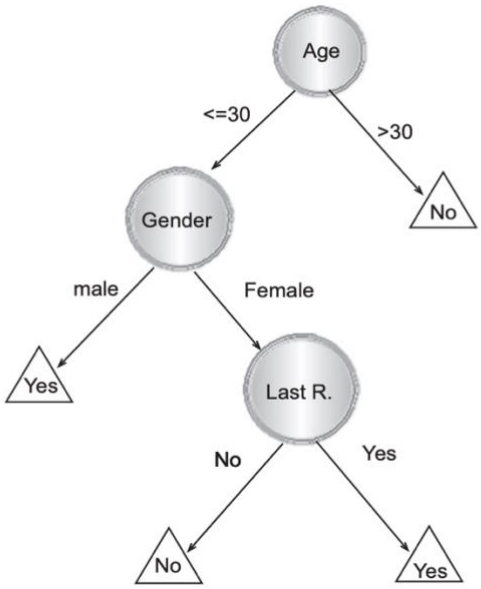

Figure 3 - An example of a decision tree taken from Rokach and Maimon [3]

4.2 – Unsupervised Learning

Unsupervised learning algorithms learn new features about data as it is introduced. Comparisons are made between new and old data and it continues to learn as more and more data is introduced.

An example of this algorithm can be seen within an implementation known as 'K-Means Clustering'. This technique automatically creates groups or 'clusters', and items which share similar characteristics are grouped together. Dey tells us that the name 'k-means' is given because the number of clusters which will be determined is unknown at the beginning [1]. Below is an example of a k-means clustering result, whereby we can see that the data has been classified based on similar characteristics *[Figure 2]*.

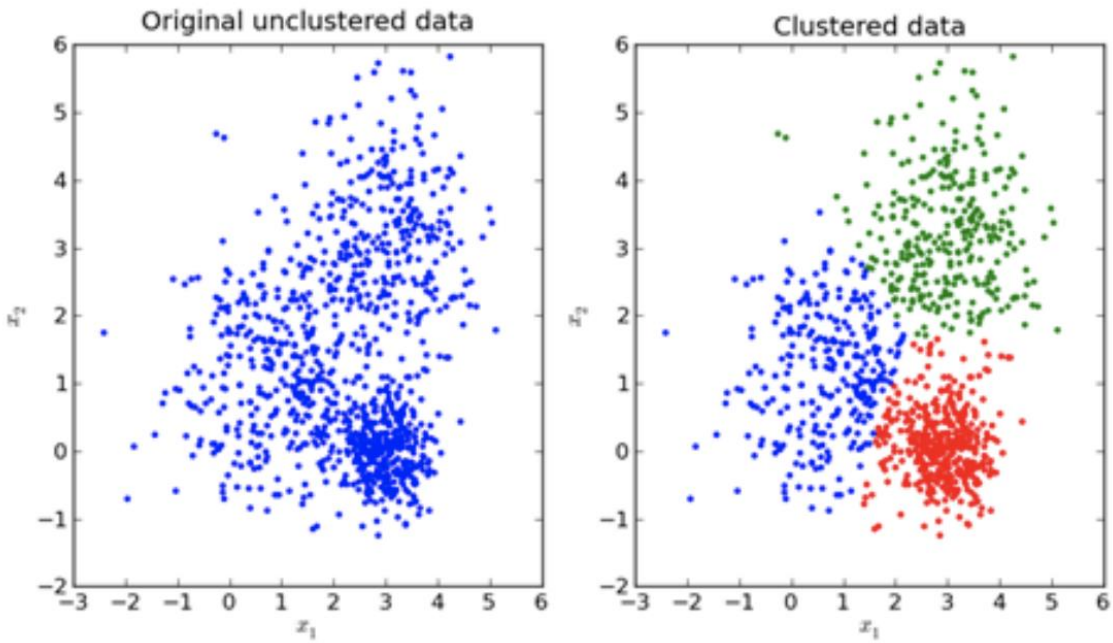

Figure 4 - An example of data after a k-means clustering algorithm has been applied. Taken from Dey [2].

4.3 – Neural Networks

The primary focus of this research is around the implementation of neural networks. Gurney describes these networks as "an interconnected assembly of simple processing elements, units and nodes, whose functionality is loosely based on the animal neuron" [8]. Neural networks are particularly useful in cases whereby it may be a struggle to classify data. We can see neural networks being used in fields such as facial and speech recognition, whereby over continued use, the application using this network will become better at making decisions and smarter at understanding different dialects. These algorithms allow machines to learn independently, and therefore become better over time without there necessarily being a need to implement changes to the system.

Zwass tells us that neural networks are typically trained under supervised learning [9]. A network requires a level of training before it is ready to begin working alone on data it is given. Firstly, it must be told what it is looking for in the data so it can begin to categorise

what it sees. For example, a speech recognition neural network would need to be trained against a number of different accents so that it can begin to learn words being spoken in different ways. A paper published my Microsoft details the extensive use of deep neural networks with regards to speech recognition [10]. The network analyses voice data for use with speech recognition within applications and devices.

Neural networks have the ability to be as complicated as necessary when it comes to sorting out the data they are given. A more complex neural network will require a larger amount of compute time, so it is important to find the balance between accuracy and speed. It would be unwise to create a very complex network for an application which does not require a high level of accuracy.

Unfortunately for developers, according to Rojas, neural networks require a large amount of computation time [11]. This can be problematic for developers who want to create quick-to-use applications that have the power of artificial intelligence which neural networks provide. However, the benefits outweigh the computational expense in most cases.

4.3.1 – Real World Applications of Neural Networks

In today's world, we are always exposed to this type of learning without even knowing it. Many homes are now adopting 'smart' devices such as Google Home and Amazon Alexa. These devices collect voice data continually to adapt to its specific users as well as learn different styles in which users may speak.

The Alexa neural network plays a big focus on personalised data. In a paper by Kim, Kim, Kumar and Sarikaya of Amazon [12], it is explained that Alexa utilises a "Personalized attention method". This is particularly interesting with regards to the research being carried out for this project, as a large focus on that on neural networks are personalisation.

4.4 – Applying AI Ideas

By looking at these modern-day examples, we can see just how complex the tasks typically are for neural networks, and we can see how computationally expensive they can be. This may often make people question the application of a neural network within simple software applications, but this study aims to prove that there is a need for these complex systems in small tasks. On a grander scale, this application may be used hundreds of thousands of times a year and having a system which can speed up the rate of producing these plates will aid heavily in the long run.

4.5 – Loss Functions

According to Verma [13], A loss function helps us to optimise parameters used in a neural network. We can calculate the loss each time we train and test a network, allowing us to optimise parameters over time. This will in turn help to improve the network overall. There are a number of different loss functions which we can use to help us in training a network.

4.5.1 – Mean Squared Error

Typically used in regression tasks, mean squared error is a common choice of loss function. It calculates loss by taking the mean of the squared differences between target and predicted values [13]. It is common to use mean squared error when the output of a neural network is a real number, such as when estimating house prices.

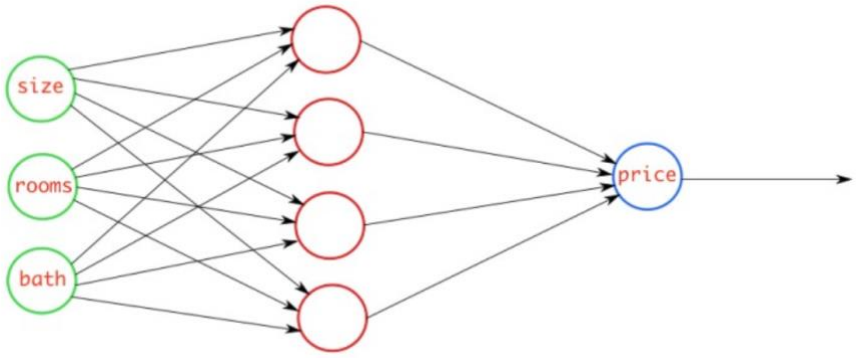

Figure 5 - An example of a mean squared error loss function. Taken from Godoy [14].

4.5.2 – Binary Cross-entropy

According to Daniel Godoy, when performing binary classification using a neural network, it is quite common to use Binary Cross-entropy as the loss function [14]. This method is ideal for networks which are attempting to perform binary classification of data. The output Is passed through a 'sigmoid activation function' [13] where the output is converted into a value between 0 and 1. These values represent a Boolean output, but allow for a level of 'certainty'. For example, if 0 represents 'no' and 1 represents 'yes', an output of 0.3 would be classified as 'no', but not with absolute certainty. A value of 0.5 and above would be given a classification of 'yes'.

In the below example *[Figure 6]*, we can see how three categories of temperature, cloud coverage and humidity are passed into a network to determine if the conditions mean it will rain or not. We obtain a value between 0 and 1 so that we can make this determination.

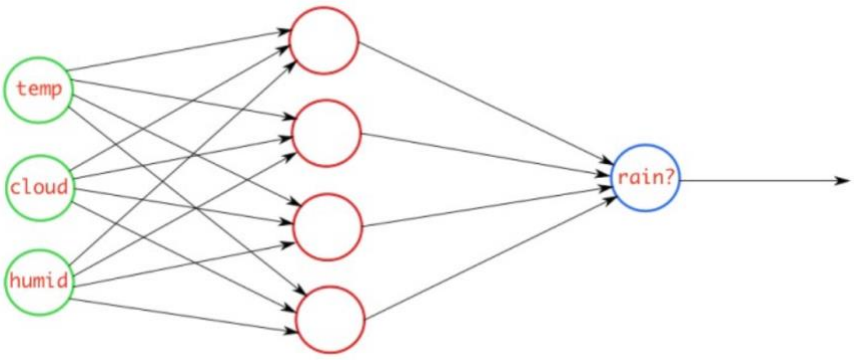

Figure 6 - An example of binary cross-entropy, taken from Godoy [14]].

4.5.3 – Categorical Cross-entropy

Godoy also discusses the idea behind categorical cross-entropy. This loss function is used when we have a *"multi-class classification task"* [14]. When using this methodology, it is important that there are as many output nodes as there are classes. This concept works well for image recognition tasks where we are trying to match an image to a given category.
For example, if we are trying to classify an image, the image will be given a score based on the probably for which it may belong each category. We can then establish that the image belongs to the class for which it scored the highest.

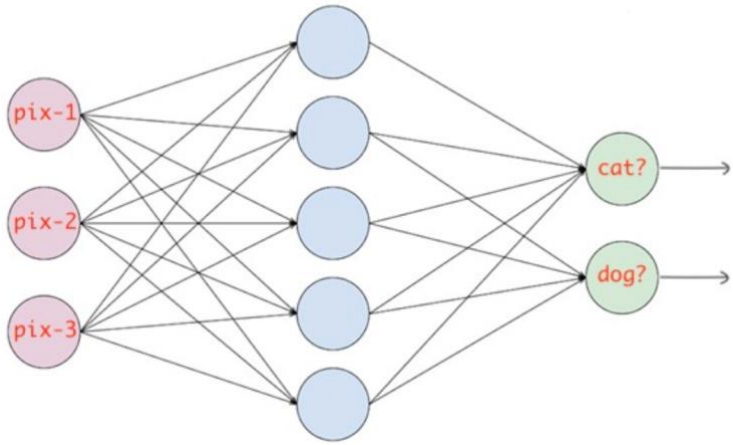

Figure 7 - An example of categorical cross-entropy, taken from Godoy [14].

5 – Neural Network Solutions

In order to determine how viable a neural network is as a solution to the problem at hand, we must first determine the way in which we would like to implement such a system. There is a vast quantity of software available which can help to build a network of our own. It is important that we consider all of the available options before we choose one, or even begin to work towards a new implementation from scratch.

5.1 – Software Package Solutions

To begin to implement a neural network, it is important to assess a number of different options to ensure that the best possible solution is the one which is used. It is advantageous to use software which already exists to begin the implementation as this will save time and effort which may be better used elsewhere, especially if a suitable candidate already exists. Below is an analysis of a number of different software used for neural network implementation.

5.1.2 – Neural Designer

Neural designer is a popular choice within neural network implementation. According to the company, *"Neural Designer is a machine learning platform that helps you discover relationships, recognise patterns, predict trends and find associations from your data without programming"* [15].

For use in this project, Neural Designer offers a fast implementation for a neural network, with little to no need of any programming knowledge. This would be ideal for beginners; however, this tends to offer less functionality to the user and less customisability as to how the network operates.

Neural Designer offers a number of tools which can allow for very fast results without the need to know necessarily how the network actually works. It would be difficult in this case to understand why anomalous results (if present) have appeared, as there is little understanding of how the backend functions.

5.1.3 – Darknet

Darknet is an open source package for creating neural networks, written in C and CUDA. This means that is allows from both CPU and GPU computation. Having this ability means that we can implement a fast, neural network which can be trained with an extensive amount of data in less time than it would take using just a CPU.

Darknet offers the ability to have a deeper understanding of how the network works as it does not hide everything behind a typical user interface. However, it does mean that more time will need to be spent on understanding how to use the software. This though is outweighed by the possibilities it has. Being open source, it also means that it is free to use which is beneficial to the research also.

Typically, Darknet is used for image processing on a network, although this does not limit its capabilities. This software package is a good contender for implemented the neural network needed for this project.

5.1.4 – TensorFlow

TensorFlow calls itself, *"An end-to-end platform that makes it easy for you to build and deploy [machine learning] models"* [16]. It's branded as a great solution for neural network implementation, ideal for beginners and experts. TensorFlow is primarily implemented using the Python programming language, making for shorter and fast code.

A number of companies use TensorFlow for research and development including Google, Intel, Coca-Cola and Airbus [17]. Airbus in particular use TensorFlow for looking at the Earth's surface and monitoring changes for urban planning and *"mapping damage and landscape changes caused by natural catastrophe"* [17].

5.2 – Chosen Implementation

For this study, TensorFlow is the chosen platform by which the neural network will be implemented, using the Python programming language. TensorFlow has been chosen as it is ideal for all levels of experience and there are a vast amount of learning resources and tutorials available. This should allow for a simple implementation without having to construct the complex inner workings of a neural network from the ground up.

TensorFlow in particular has multiple example for binary classification [18], which is also the desired result for the purposes of this research. Therefore, implemented the same type of system for categorising vehicle registrations into offensive and non-offensive should be a relatively hassle-free process.

5.3 – Dataset construction

The data for the vehicle registrations will be relatively basic. Each element of the dataset will contain a string of characters, representing a vehicle registration and a Boolean flag, indicating if the registration has been predetermined to be offensive or not. It is important to have an even mix of many offensive and non-offensive registrations so that the network can be trained to the best level of accuracy possible.

The offensive registrations will be generated from a list of banned registration combinations held at the DVLA. This list contains combinations which create offensive words/phrases. This data is listed within a PDF file so we will need to build a parser to extract the required data.

6 – Research Methods

When conducting research for this project, it is important that we aim to achieve the most reliable results possible, to determine the validity of the outcome. We hope to see that the use of neural networks is viable for detecting if a registration plate is offensive or not. If this does indeed work, we may be able to implement these ideas into many other areas.

6.1 – Strategy and Approach

When conducting research for this project in the form of a user study, it must be ensured that the users are not led to be biased opinion before they begin. By this point, the initial neural network will have been created, and able to analyse any number of vehicle registrations it is given.

Each participant will be given a short questionnaire containing a list of possible vehicle registrations. Some will consist of a potentially interpretable offensive set of characters, and others will not. The participants will indicate as to whether or not they believe a particular registration to read as offensive.

Initially, we would hope to see that the group of participants generally give the same results. This would give an accurate analysis of which combinations are offensive and which are not. It should also be noted how many users have taken part in the survey to give more validity to the human result.

If we do obtain mixed results, we would hope to see a trend along with the type of user which has taken part. Therefore, the survey will ask the user to indicate what age bracket they fall into. The categories will be made up on 18-30, 31-50, 51-64 and 51+. Certain words/phrases which may be deemed offensive may be terms which are new or no longer used regularly. This means that certain age demographics may have heard of a particular phrase, whereas others haven't. The age categories will start with a minimum age of 18 due to the potentially offensive language included in the study.

By the end of this stage, we should have a list of vehicle registrations whereby we can say with certainty, contain an offensive set of characters, based upon human interpretation.

6.2 – Risk Analysis

The survey aims to give a list of vehicle registrations which are definitely offensive or not, although some combinations may cause issues. For example, they may be sets of characters which create words that are deemed offensive only in certain cultures. For the purposes of the study, it would be ideal to have well known offensive terms, although it is very important to consider people from different demographics. Therefore, if one participant deems a registration to appear offensive, we should either mark it as offensive or remove it from the testing pool.

There are no immediate risks for the users involved in the study due to the simplicity of the experiment and lack of hazardous equipment. However, it is not to be ignored that there are a number of risks with reliability of data when it comes to this type of research. Some people may simply not find certain terms offensive whereas others will. It should be

explained to the participants that they make their decision based on what they think they or others may say, in order to not only get results based on what only they personally find offensive.

6.3 – Data Evaluation

Once the surveys have been completed, the results will be plotted on a bar chart to see clearly which combinations were perceived to contain an offensive set of characters. We hope to see clearly which registrations were interpreted as offensive. This will help us to create the testing data, whereby we can apply the correct offensive/non-offensive flag.

We will also look into trends within the given age brackets to understand how certain words/phrases are interpreted by people of different ages. This will give us insight into how broad and understanding we need to be when removing offensive vehicle registrations. It is important that all demographics are considered when we look into what is and isn't offensive, so that we do not unintentionally offend somebody.

From the list of banned registration combinations already held by the DVLA, we can create training data by using many offensive and clean combinations. Since we already have approximately 5000 banned combinations from the DVLA data, we should create approximately the same amount of clean data. This can be achieved by simply creating random registrations and checking that they do not appear in the already banned list.

By this stage, we can continue testing the data using the neural network. After the network has been trained, we hope to see a fairly high level of accuracy (greater than 80%) from the test data evaluation. This number is based on the accuracy given from the TensorFlow binary classification example data [18].

7 – Software Development Methods

The software will be designed and implemented using an agile methodology [19]. The software product will go through a number of iterations, adding further levels of functionality over time, but always having a level of usability at each iteration. This development model allows for a working product to be produced in a less amount of time compared to traditional approaches.

Throughout development however, the software development lifecycle will be taken into consideration when implementing new features.

Figure 8 - Software Development Lifecycle taken from Synotive - https://www.synotive.com/blog/software-development-client-questionnaire

By following this, it will be far easier to develop new features and implement them over time through a number of iterations. Each stage of the lifecycle shows how a software application is created which the cycle starting again at each new feature being implemented.

Even though the software will be developed by a single individual, a scrum methodology will be adhered to where each feature will be developed using short sprints to make sure that time is being managed effectively. This is to ensure that by the time of the user study, the software is fully ready, and the neural network is working correctly.

8 – Research Survey Results and Analysis

As previously discussed, a survey was conducted in order to understand people's interpretation of offensive language within different age ranges. This survey is a key component in helping us disprove our null hypothesis:

Age has no effect on the interpretation of offensive words/phrases contained within vehicle registrations.

It is important that we gain participation from a balanced number of users from each age bracket to keep the results as reliable as possible.

For the study, 80 people participated, with 20 in each of the four age brackets. Beforehand, the users were asked which age bracket they fall into so that the results could remain as balanced as possible. This allowed us to analyse the results from four individual age brackets to prove that age does have an impact on people's ability to spot offensive words/phrases within vehicle registrations. This means that we would expect to see a balanced interpretation of words/phrases hidden within vehicle registrations regardless of the age of the participant.

8.1 – Initial Results

From initially collecting the results, we can see already that there are indeed few responses by older participants which indicate that certain vehicle registrations should be removed from circulation, based on the words/phrases interpreted from them.

Registration	**18-30**	**31-50**	**51-64**	**65+**
CE15 RJF	0	0	0	0
MU19 DER	20	20	16	15
FP54 PPJ	1	0	0	0
LP13 UUJ	0	0	0	0
BA55 TRD	19	20	20	18
SL65 UTS	20	19	17	15
NO67 NOB	19	20	15	15
FA16 CHB	3	1	1	0
PR11 CKK	20	20	18	17
GU15 FBR	2	0	0	*0*

Table 1 - Initial results from the user study to detect if there is a trend between age and offesnive language interpretation in vehicle registrations.

Once this data has been plotted onto a bar chart, we can indeed visualise the trend. We can see clearly that participants in older age groups seem to be less perceptive of offensive language held within certain vehicle registrations.

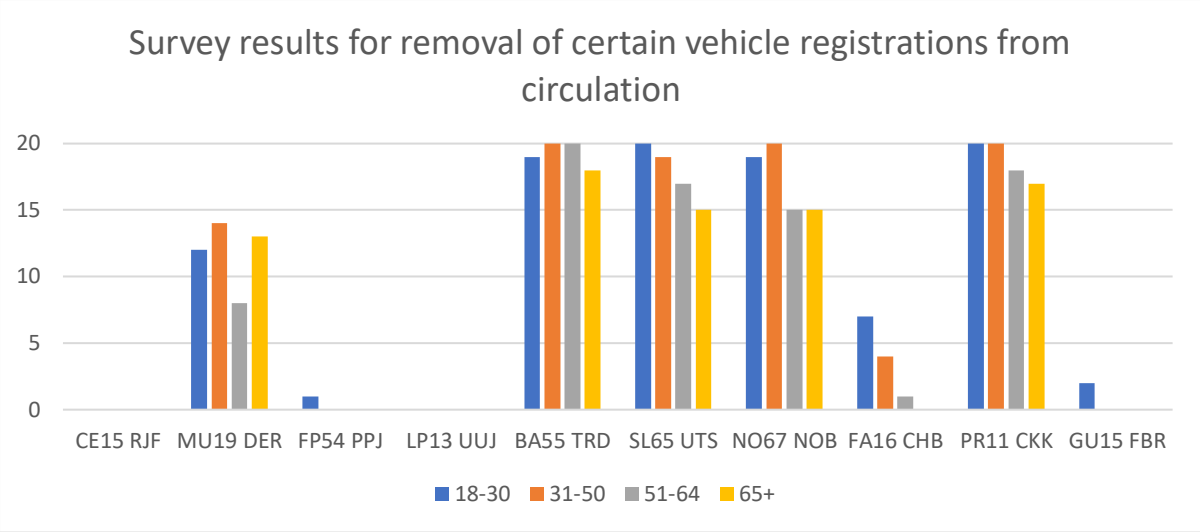

Table 2 - A bar chart visualising the offensive vehicle registrations survey data

Whilst the decline is not immediately significant, it is clear that there is indeed a trend which disproves our initial null hypothesis. We can say with certainty however that all of the registrations which contained potentially offensive language were correctly identified by each age group, with very few participants either not finding an offensive word or deriving one from a non-offensive character combination.

8.2 – Data Analysis

It can be easily said that the data collected here helps us to understand trends in age and offensive language interpretation. However, there are a number of factors which may have helped in creating the data which we have collected.

Firstly, we must look at the language which has been used in the study. 4 out of 5 registrations which were previously deemed offensive contain words which are typically used as an insult from one person to another. For example, if we look at the ninth registration in the survey, *PR11 CKK*, it is clear to most English speakers that the word "prick" is easily derivable from this set of characters. This was correctly identified by the whole of the first two age groups, and the majority of the second two. We can see that as age increases however, there is a notable decline in the interpretation of an offensive word in this vehicle registration.

There are two main theories as to why this decline in offensive language interpretation has taken place. Firstly, it could be the case that older participants found it more difficult to interpret offensive language our of characters within a vehicle registration. Secondly, it could be simply down to the meaning of the word and how it has changed over time.

It is only in the last century that the word "prick" has become a slang term used as an insult, usually referred to as a vulgar term for male genitalia [20]. Aside from this, the term is often used to describe a pointed end, or the act of puncturing something. It is quite possible that the term is not perceived as offensive to more of the older generations, and therefore would indicate that the word is being associate more commonly over time as an evolving offensive term.

When we look at a line-graph for the vehicle registration in question, we can see that as age increases, less participants were able to detect an offensive word/phrase hidden within.

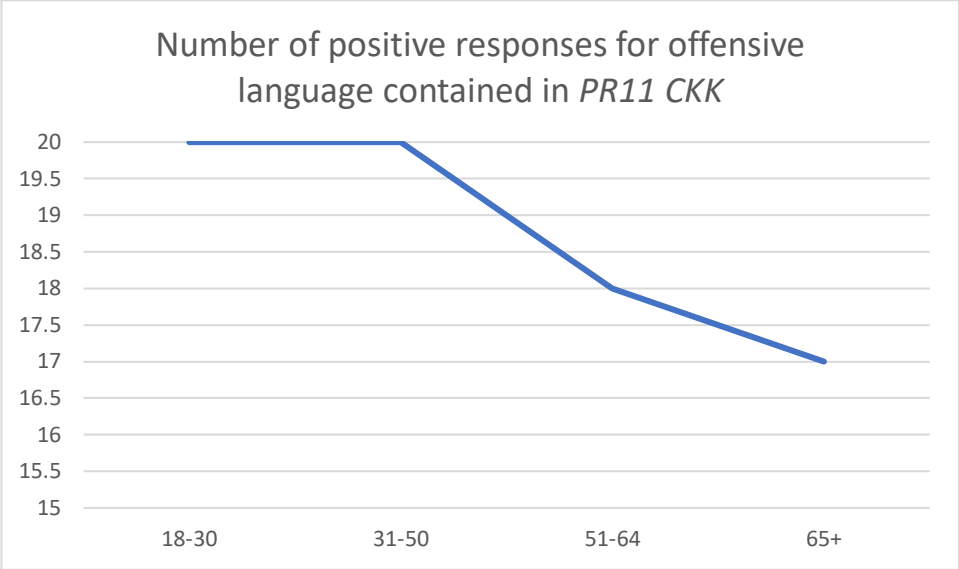

Table 3 - Number of positive responses for offensive language per age bracket for the vehicle registration PR11 CKK

This trend also seems to be prevalent in vehicle registrations which were previously deemed to not contain offensive terms. If we look at the survey results for the registration *FA16 CHB*, we can see that there were some responses which indicated that some participants found it to contain offensive language. From the survey, there is a part where participants can indicate what words they believe are contained within the registration. A number of the participants who found it to be offensive believed that the characters contained the word "fat" and/or "chub", which may be used in an offensive way for people who are overweight.

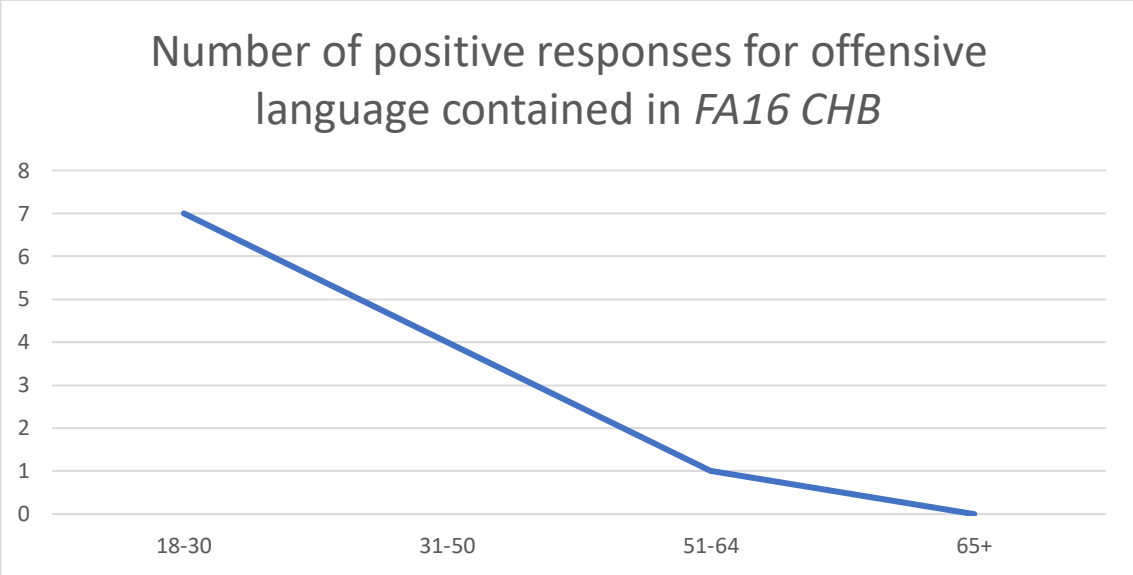

Table 4 - Number of positive responses for offensive language per age bracket for the vehicle registration FA16 CHB.

This data is interesting or a number of reasons. Firstly, neither 'FA** ***' or '**** CHB' are banned characters combinations already identified by the DVLA. However, it is clear that some participants feel as though these combinations do indeed contain words which are unfavourable.
It is also interesting as we are seeing the same trend as we saw for the data which did contain words previously deemed to be offensive. We can likely attribute this to the more recent rise in weight awareness in news and media, and an increase in general rise in unhealthy individuals over the past few decades.

By looking at data from parliamentary obesity statistics, Carl Baker indicates that obesity levels have increased of 15% to 29% between 1993 and 2016 [21]. In just this small period of time alone, we can see a dramatic rise in weight issues within the UK, which would typically also introduce a stronger level of awareness for people all over the UK. The trend in our data indicates that younger people take more offense to terms such as this than the older generations do, which supports the idea that the interpretation of this registration would be deemed more offensive by younger participants. This further disproves our null hypothesis from the beginning of the study.

Although we can see a clear trend in our data for the most-part, we can also identify that we have one clear outlier in our results. This outlier sits with the registration *MU19 DER.* All participants who felt that the plate should be removed from circulation correctly identified that the word 'murder' could be interpreted from it. This result is particularly interesting as the term itself is not considered offensive, but instead gives some people a feeling that it is not suitable, based on what it means.

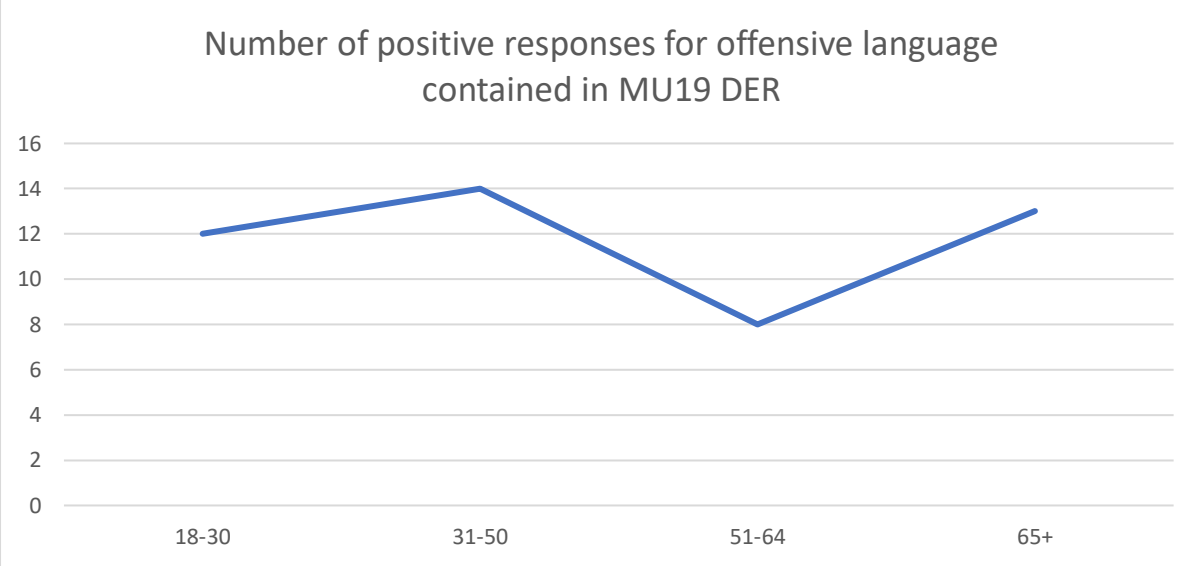

Table 5 - Number of positive responses for offensive language per age bracket for the vehicle registration MU19 DER

Our results indicate that individuals ages between 31 and 50 agreed that the registration should be removed more than any other age group, which is what makes this data inconsistent with the rest. Individuals in this case may not necessarily have found the word to be offensive but found that it should be removed on a different basis. This means that not only should we be aiming to remove offensive words from circulation but look at

removing inappropriate terms also. Terms such as these may not offend the general public but may instead upset people. It is important to consider people's background and history before assuming that a word/phrase will not cause any upset for them.

8.2.1 – P Value

When looking for statistical significance, we can use a P value as a means of determining the probability of our results being able to disprove our null hypothesis. As discussed by Dahiru [22], the P value looks at the data and compares this with the expected results, calculated using an average of the collected data. If the value lies at 0.05 (5%) or below, then we can say with strong confidence that our research disproves our null hypothesis.

When looking at the data which we have collected, we can use statistical analysis techniques to look into how well we can disprove our null hypothesis. Once calculated on our data, we obtain a P value of 0.01139189, which is approximately 1.1%. This strongly indicates that we have disproved our null hypothesis with the data which can been collected.

8.3 – User Study Conclusions

From the results we have gained, and the statistical analysis performed, we can say with a strong level of confidence that the research findings disprove our null hypothesis. We can see through data visualisation, an obvious trend which goes against our initial null hypothesis. The calculated P value also leads us to believe that our results show us that age does indeed have an impact on interpretation of offensive language.

9 – Software Implementation

As discussed previously, the neural network implementation will be carried out using TensorFlow with the Python language. This is due to the vast amount of resources available which we can reference for our own implementation. As established, we will be performing binary classification so that we can categorise vehicle registrations into either 'offensive' or 'non-offensive'. In order to do this, we will be using a the 'binary cross-entropy' loss function to give us a result whereby we can categories a registration into one of two categories.

9.1 – Constructing the Initial Classifier

To begin, we will be basing our classifier off an example by TensorFlow. This example looks at analysing movie reviews and categorising them into either positive or negative reviews [18]. This is a similar process by which we want to categories our vehicle registrations, therefore, it would be a good example to build our network off.

The data used in this example is taken from IMDb [23] and contains a balanced mix of positive and negative movie reviews. The positive/negative tag for each review has been placed there manually based on human interpretation, therefore we can use this data to train and test the binary classifier network.

We split the data into our train and test data by choosing how much of either we want. The typical split is 70% train and 30% test. The more training data we have, then the more accurate the neural will be in theory. Although, we then have less data to use when it comes to testing the network. Therefore, 70/30 is the typical split.

9.2 – Creating the Train/Test Data

In order to train a neural network using vehicle registration data, we need a number of elements. Firstly, we need a large collection of vehicle registrations. These registrations need to be a fairly balanced mix of offensive and non-offensive combinations, based upon data already analysed by humans. We need already classified registrations so that we are able to train the network so it can begin to understand what combinations make up a potentially offensive word/phrase.

The data we are initially using comes directly from the DVLA. The agency holds a list of vehicle registrations which are removed from the new batches which are released every 6 months. It also contained all banned letter/numbers combinations which can never be used on any series or custom vehicle registration. From this list, we can begin to construct our offensive registration data. We know that this list is accurate, as it has been built up over time by a group of people from different demographics to cover all possible areas of potentially offensive language. This gives us a good starting point.

The data as it stands exists on a PDF document, outlining each series and the already removed registration combinations. Therefore, we will need to produce one or more small programs to extract the data we require and then build up valid data which we can use to train our network.

9.2.1 – PDF Data Abstraction

Initially, the PDF of vehicle registrations was converted to a CSV format. Each element of the PDF is read using space delimiting. Each of the elements are abstracted and stored in a CSV using a third-party application.

Once we have the CS, we can create a small Java program to extract all of the registrations. Java was the chosen language for this task due to an already solid understanding of the language and prior experience with using it to handle files. The CSV includes titles and headings which we want to remove from our data. We can do this by using a regular expression (RegEx) to look for all character combinations which are seven characters in length and contain only integers and upper-case letters. This leaves us with only the vehicle registrations which we can store into another CSV.

```
Pattern p = Pattern.compile("([^a-z]{7})");
```

Figure 9 - RegEx used to find vehicle registrations in strings

We can adjust our regular expression to look for patterns which are seven characters in length and contain no lower-case letters, as this yields the same results using less code. This is demonstrated above.

9.2.2 – Creating Non-offensive Vehicle Registrations

By this stage, we now have a CSV file containing all of the offensive vehicle registration combinations created by the DVLA, alongside a Boolean flag indicating that all of the current registrations are deemed offensive. We need to develop a way of creating non-offensive vehicle registrations so that we can construct a mixed dataset of possible vehicle registrations. We can implement a simple random vehicle registration generator which conforms to the standard UK format. Upon creating each registration, we check if it already exists in the banned list and if it does not, we add this to the non-offensive list, and give it a flag indicating that it is not deemed offensive.

```
switch(pos) {
    case 0: return randomCapital;
    case 1: return randomCapital;
    case 2: return (char)(rand.nextInt( bound: 10) + 48);
    case 3: return (char)(rand.nextInt( bound: 10) + 48);
    case 4: return randomCapital;
    case 5: return randomCapital;
    case 6: return randomCapital;
    default: return '?';
}
```

Figure 10 - A Switch statement used for giving a random vehicle registration character given the current character index

This method however does present an issue whereby a combination is creating which is deemed offensive by individuals but is not already present on the offensive list. This is a small edge-case as the majority of offensive combinations have already been considered, and an anomaly such as this would not present itself often enough to cause a large issue. Therefore, we can for now accept that some registrations may be incorrectly marked, if human perception was also taken into consideration.

We can now create two lists of vehicle registrations, containing offensive and non-offensive vehicle registration combinations which we can use for training and testing our neural network.

9.2.3 – Generating JSON

To allow for use in a wider range of applications and languages, it is a good idea to also represent our data as JSON (JavaScript Object Notation). This data can be sent easily through web applications and would allow for better integration into larger applications, as well as giving us another way to read in our data.

We can read in our CSV data and create a JSON file using a small Java method, as demonstrated blow.

```
for(int i = 0; i < regnums.size(); i++) {
    out.println("\t{");
    out.println("\t\t\"reg\": " + "\"" + regnums.get(i) + "\",");
    out.println("\t\t\"offensive\": " + "\"" + 1 + "\"");
    out.print("\t}");

    if(i < regnums.size() - 1) {
        out.println(",");
    } else {
        out.println();
    }
}
```

Figure 11 - Java code to print out data in a JSON format

9.2.4 – Creating a Mixed Data CSV for TensorFlow

For use in TensorFlow, we will be reading the data in a CSV format, which also gives us great versatility for viewing the data is different applications. To create a simple CSV containing all of our data, we firstly need to write a program which will read our JSON files. To do this, we will utilise the 'org.json' library from the Maven repository [24]. This allows us to easily parse JSON and extract the data we need.

```
for(int i = 0; i < offensiveRegArray.length; i++) {
    out.println(offensiveRegArray[i] + ",1,");
    if(i < cleanRegArray.length) {
        out.println(cleanRegArray[i] + ",0,");
    }
}
```

Figure 12 - Code snippet producing the CSV for mixed registration data

By this stage, our data is ready for use in TensorFlow, whereby we can begin to train/test our neural network.

9.3 – Using TensorFlow for Binary Classification

As previously discussed, the work on our binary classifier will be based off the work by François Chollet [18], who using the TensorFlow libraries, demonstrates how we would create a neural network for binary classification using binary cross-entropy as the loss function.

The example in question is great for looking at how TensorFlow works and how we can use it to build a neural network for our data. This example looks at analysing movie reviews and categorising them as either positive or negative.

Firstly, we collect the data and allocate a train/test split value. In the case of this example, we have chosen a 70% train and 30% test split. This is the usual split ratio which most networks begin with, but we can alter this and see the impact which this has on the network.

```
train_data, validation_data, test_data = tfds.load(
    name="imdb_reviews",
    split=('train[:70%]', 'train[70%:]', 'test'),
    as_supervised=True)
```

Figure 13 - Python code snippet which loads the IMDb movie review data and initialises a train/test split using the TensorFlow Datasets library.

TensorFlow also allows us to set a number of parameters, giving us great flexibility with how the network operates. The library incorporates a number of optimisation algorithms [25], which are used extensively within neural networks. In the case of this example, we will be using the 'adam' optimiser. We can also state our loss function, which as previously discussed, will be 'binary cross-entropy' as this is ideal for data which we want to perform a binary classification on. Lastly, we define our results metric. In this case, we will be looking at accuracy, as this will give us an idea of how well our neural network operates and how reliable it is for use.

```
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])
```

Figure 14 - Python code snippet setting the optimiser algorithm, lost function and result metrics.

10 – Software Implementation Results

Upon running the TensorFlow code for the movie reviews example, we can see that we get a loss value of 0.322 (32.2%) and an accuracy rating of 0.861 (86.1%).

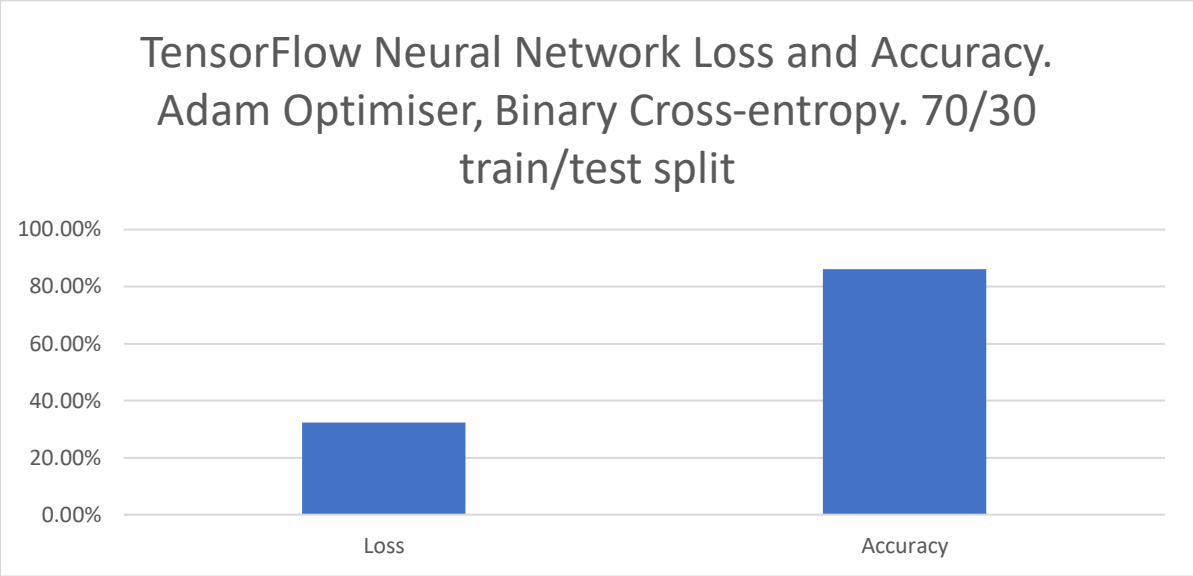

Table 6 - Loss and Accuracy results from initial use of TensorFlow.

This implementation is considered fairly naïve which is why we have an accuracy reading below 90%. However, the results from this implementation are still very credible and show us that binary classification is indeed a good task for a neural network to perform. With some further tweaking and more advanced methods, we would hope to see an accuracy rating closer to 95%, giving the network even more credibility as a good method for classification. We would also ideally like to see a lower loss result. The number represents the level of error on our network's analysis, which we would want to get as close to 0 as possible. It is virtually impossible for us to achieve zero loss, but the lower the value, the more credible the network for its ability to perform binary classification.

We can improve our network by experimenting with a number of parameters. We could look at altering the train/test split, and even the data itself. The more data we have, the better our network's chances of being able to perform more accurate classification. We could also look at altering our optimiser algorithm and loss function. There are a range of optimisation algorithms available for implementation, these could yield different result and over time could allow us to improve the network as a whole. Although our loss function is designed for use in binary classification, we may also see improvement by utilising others.

11 – Summary and Conclusion

Our initial point of research began with our null hypothesis around the interpretation of offensive language. We stated that, *"Age has no effect on the interpretation of offensive words/phrases contained within vehicle registrations"*. However, through the research carried out, we can say with a strong level of certainty and confidence, that age does indeed have an impact on the way in which humans interpret offensive language. Not only may it impact how we perceive hidden words and phrases within vehicle registrations, but the meaning behind certain words changes over time. It is important to understand that like us, language evolves, bringing about different meaning and understanding from the words we use.

Our data shows us that participants in different age brackets likely took different meaning from certain terms and also the 'level of offense' which they cause changes over time. This makes the banning of certain terms an interesting and ever on-going task, as we can't outright declare which words are fine and which are not. Therefore, the use of a neural network with a task such as this allows us to train programs so that they can learn and adapt with us over time.

We have shown that an implementation of a neural network can in today's world be fairly trivial, given the right tools. We have identified that there is a place of artificial intelligence in the realms of binary text classification, and we can utilise these technologies to overcome outdated and often slow, manual and inaccurate processes of human interpretation and intervention.

References

[1] A. Dey, "Machine Learning Algorithms: A Review," *International Journal of Computer Science and Information Technologies,* vol. 7, no. 3, pp. 1174-1179, 2016.

[2] S. S. Mufwene, The Origins and the Evolution of Language, Chicago: University of Chicago, 2013.

[3] S. H. Hagen, "Swearwords and attitude change: A sociolinguistic study," Universtiy of Bergen, Bergen, 2013.

[4] Ipsos, "Attitudes to potentially offensuive language and gestures on TV abnd radio," Ofcom, 2016.

[5] J. Harbeck, "Mind your language! Swearing around the world," BBC, 6 March 2015. [Online]. Available: http://www.bbc.com/culture/story/20150306-how-to-swear-around-the-world. [Accessed 12 09 2019].

[6] E. Roach and B. B. Lloyd, "Cognition and Categorization," Lawrence Erlbaum Associates, Hillsdale, 1978.

[7] L. Rokach and O. Maimon, Data Mining and Knowledge Discovery Handbook (Second Edition), New York: Springer Science and Business Media, LLC, 2010.

[8] K. Gurney, An Introduction to Neural Networks, London: UCL Press, 1997.

[9] V. Zwass, "Neural Network," Britannica, [Online]. Available: https://www.britannica.com/technology/neural-network. [Accessed 31 May 2019].

[10] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath and B. Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition," *IEEE Signal Processing Magazine,* pp. 82-97, November 2012.

[11] R. Rojas, "Neural Networks," Springer-Verlag, Berlin, 1996.

[12] Y.-B. Kim, D. Kim, A. Kumar and R. Sarikaya, "Efficient Large-Scale Neural Domain Classification with Personalized Attention," Amazon.

[13] S. Verma, "Understanding different Loss Functions for Neural Networks," 20 June 2019. [Online]. Available: https://towardsdatascience.com/understanding-different-loss-functions-for-neural-networks-dd1ed0274718. [Accessed 12 January 2020].

[14] D. Godoy, "Understanding binary cross-entropy / log loss: a visual explanation," 21 November 2018. [Online]. Available: https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a. [Accessed 6 January 2020].

[15] Artificial Intelligence Techniques, Ltd., "Neural Designer," Artificial Intelligence Techniques, Ltd., 2019. [Online]. Available: https://www.neuraldesigner.com/. [Accessed 18 September 2019].

[16] TensorFlow, "Why TensorFlow," TensorFlow, [Online]. Available: https://www.tensorflow.org/about. [Accessed 21 December 2019].

[17] TensorFlow, "Why TensorFlow - All case studies and mentions," TensorFlow, [Online]. Available: https://www.tensorflow.org/about/case-studies/. [Accessed 21 December 2019].

[18] F. Chollet, “Text classification with TensorFlow Hub: Movie reviews,” TensorFlow, 2017. [Online]. Available: https://www.tensorflow.org/tutorials/keras/text_classification_with_hub. [Accessed 21 12 2019].

[19] Agile Alliance, “Agile 101,” Agile Alliance, 2019. [Online]. Available: https://www.agilealliance.org/agile101/. [Accessed 31 May 2019].

[20] Merriam-Webster, “prick noun,” Merriam-Webster, [Online]. Available: https://www.merriam-webster.com/dictionary/prick. [Accessed 27 January 2020].

[21] C. Baker, “Obesity Statistics,” House of Commons Library, London, 2019.

[22] T. Dahiru, “P – VALUE, A TRUE TEST OF STATISTICAL SIGNIFICANCE? A CAUTIONARY NOTE,” Association of Resident Doctors, UCH, Ibadan, Ibadan, 2008.

[23] IMDb, “IMDb,” IMDb, 2020. [Online]. Available: https://www.imdb.com/. [Accessed 24 January 2020].

[24] Maven, “org.json,” Maven, 2020. [Online]. Available: https://mvnrepository.com/artifact/org.json/json. [Accessed 28 January 2020].

[25] TensorFlow, “TensorFlow Corse r2.1 API,” TensorFlow, 28 January 2020. [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/keras/optimizers#modules. [Accessed 29 January 2020].

[26] J. Vincent, “The State of AI in 2019,” The Verge, 28 January 2019. [Online]. Available: https://www.theverge.com/2019/1/28/18197520/ai-artificial-intelligence-machine-learning-computational-science. [Accessed 25 May 2019].

[27] A. Joshi, S. Kale, S. Chandel and D. K. Pal, “Likert Scale: Explored and Explained,” *British Journal of Applied Science & Technology,* vol. 7, no. 4, pp. 396-403, 2015.