A Comparative Study in Text-Based Emotion Recognition through Multilayer Perceptrons and Support Vector Machines

Ayliah Fani

ayliah.fani@city.ac.uk

1. Introduction

Emotion is a part of the human experience, spurred by one's circumstances, mood, or interpersonal relationships. The Cambridge English dictionary defines emotion as "a strong feeling such as love or anger, or strong feelings in general". Recently, emotion detection (ED) has become of particular interest to researchers due to a wide range of applications and inferences gained from data generated from such textual analysis. Text-based emotion detection has found footing in business, education, psychology, and various other fields where understanding and interpreting emotion holds particular interest [1].

The field of sentiment analysis (SA) is concerned with studying language through methodologies of natural language processing (NLP), textual analysis, and computational linguistics. SA's aim is the designation of polarities- positive, neutral, or negative, allowing for opinions, ideas, and thoughts to be analysed. Emotion detection is a subfield of SA that delves further past general polarity assignment by extracting emotions such as happy, sad, angry, etcetera [2]. Textual ED intends to understand the underlying feeling of the author by examining their texts [1].

This paper critically evaluates two algorithms- the support vector machine (SVM) and multilayer perceptron (MLP), comparing and contrasting their proficiency for text-based emotion recognition of six different emotions.

2. Data Set

The dataset used for this analysis was acquired from Kaggle's *Emotion Dataset for NLP.* This dataset comprised of 20 000 statements along with each corresponding conveyed emotion, totalling 40,000 instances. Six emotions were present: sadness, joy, fear, anger, love, and surprise.

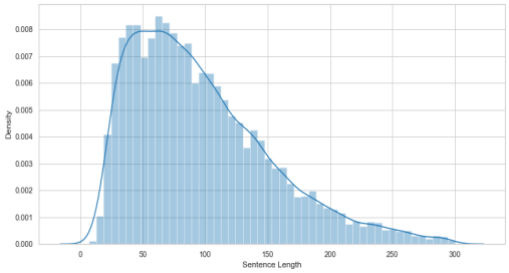

Figure 1 Distribution of sentence lengths

2.1 Exploratory Data Analysis

The data consists of personal statements of one sentence length as input text and an associated emotion as a categorical target. The dataset was of high quality and did not appear to have erroneous values or missing data in attributes being considered. Emotions were represented in the dataset as text. Therefore they were encoded to be represented numerically (sadness: 0, joy: 1, fear: 2, anger: 3, love: 4, surprise: 5). The average sentence length for each emotion was in close proximity to each other. Sentence length of all emotions had a mean of 96 words, as seen in Figure 1.

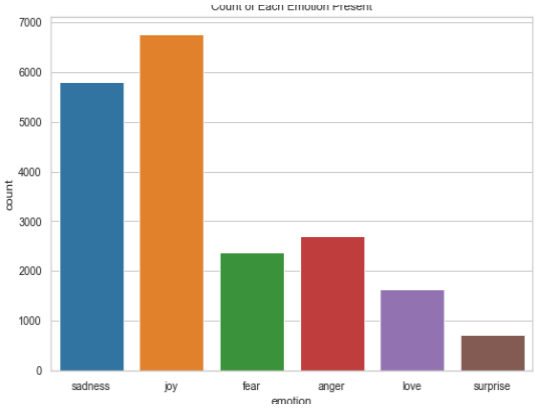

Figure 2 Emotion representation

From Figure 2 it is apparent that there is a significant class imbalance. To counteract this issue, stratified sampling was used to eliminate sampling bias in the training and testing datasets. Additional resampling methods such as under and oversampling were considered. Liu [3] and Padurariu [4] found that in the case of support vector machines used for text classification, the case where resampling does not occur tends to produce the highest precision. Liu also states Synthetic Minority Over-sampling Technique (SMOTE) and its variants usage with SVM classifiers produce high precision but with low recall and rarely produce precision higher than its non-resampled counterparts [3]. Particularly, SMOTE has a marginal effect on classifiers trained with highly dimensional data [5]. Due to this and computational cost sensitivity, it was determined that resampling techniques would not be implemented in this study.

2.2 Text Processing

The statements were pre-processed with several NLP techniques, including lowercasing, punctuation removal, stop word removal, and lemmatisation.

Following, term frequency-inverse document frequency (TF-IDF) vectorisation was done to the statements to create a bag-of-words model comprising a specified vocabulary. This model simplifies word representation disregarding word order and grammar but keeping multiplicity. TF-IDF statistically evaluates the frequency of a word in a document and its relevance in a collection or corpus [6]. The vocabulary size was determined via a logistic regression model for unbiased evaluation of feature selection. It was determined that the best number of features to retain was 2000. Over that value, performance increases were marginal and would lead to higher computational costs that were not viable for this study.

Additionally, a Chi-squared test was done to determine the most relevant features for each emotion. Table 2 shows the top 10 unigrams for each emotion. The final dataset contained 20 000 statements and 2000 features.

NUMBER OF FEATURES	ACCURACY
500	0.61
800	0.7925
1000	0.82675
2000	0.8795
3000	0.8775
5000	0.87725

Table 1: Accuracy of the base model with n-Features

Table 2: Top 10 unigrams for each emotion

Emotion	Unigrams
Sadness	Humiliate, shitty, ashamed, homesick, dull, discourage, miserable, exhaust, gloomy, punish
Joy	Festive, resolve, successful, rich, useful, divine, energetic, confident, talented, content
Fear	Reluctant, shaky, unsure, scare, anxious, nervous, apprehensive, insecure, vulnerable, terrify
Anger	Piss, dangerous, insult, dissatisfy, violent, cranky, bitchy, irritable, resentful, greedy
Love	Delicate, gentle, tender, supportive, fond, naughty, loyal, horny, nostalgic, sympathetic
Surprise	Strange, weird, stun, surprise, shock, daze, funny, curious, amaze, impress

3. Summary of Algorithms

3.1 Support Vector Machine

A support vector machine (SVM) is a supervised machine learning algorithm used for classification and regression analysis. An SVM aims to find an optimal hyperplane in an n-dimensional space- where n is the number of features that maximises the margin distance between two classes (Figure 3). The pros and cons of an SVM are outlined in Table 3.

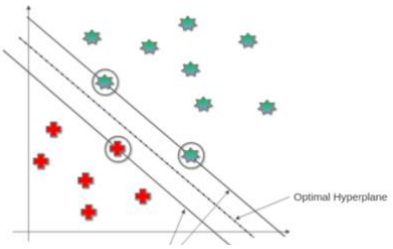

Figure 3 Optimal hyperplane and margin. SVMs in circles define the margin of largest separation between classes [8]

The SVM classifier is fundamentally a binary classifier, although various strategies can be implemented to remedy this issue. One such method is the *one-versus-rest* approach. This method constructs a single SVM per class, where the data of that class is used as positive examples, and data from the remaining classes are used as negative examples [7]. Alternatively, the *one-versus-one* approach trains different binary SVMs on all possible pairs of classes. The SVMs then classifies test samples according to which class has the highest number of 'votes' [7].

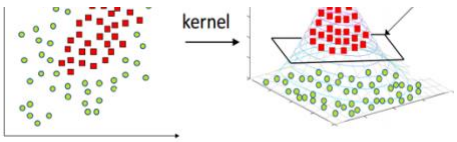

Figure 4 Kernel Trick Visualised

Another point of consideration for SVM implementation is the linear separability of data. Fundamentally, SVMs deal with linearly separable data. For instances of non-linearly separable data, a transformation of data into a higher dimensional feature space is done, allowing the classes to become linearly separable via the kernel trick. The kernel trick allows for fitting a decision boundary in n-dimensional space to separate classes and make predictions, as represented in Figure 4.

Table 3: Pros and Cons of Support Vector Machines

PROS	CONS
• Show better results than ANNs on most popular benchmark algorithms [8] • Single solution characterised by global minima as opposed to multiple local minima in NNs [8] [7] • Effective in high dimensional space • Memory efficient since it only uses a subset of training points in the decision function • Various Kernel functions allow for versatility.	• High training time for large datasets • Does not handle noise well • If the number of features is greater than the number of samples, Kernel function choice is crucial to avoid over-fitting • No direct probabilistic explanation for the classification [7]. It is calculated through an expensive 5-fold CV. • no built-in handling of multiclass problems

3.2 Multilayer Perceptron

The multilayer perceptron (MLP) is a class of artificial neural network (ANN) trained via the supervised learning technique of backpropagation [8][9]. MLPs comprise of unidirectionally connected neurons in layers: an input layer, n-hidden layers, and an output layer, where connections of neurons are only permitted between neighbouring layers [10]. Figure 5 shows the basic architecture of an MLP. The input layer is comprised of n-neurons, where n is the number of features. The input from the first layer is then fed forward to the hidden layer(s), where a weighted linear summation and nonlinear activation function is applied [9]. The activation function of a node establishes a node's output given an input or set of inputs. These activation functions are then fed into the output layer.

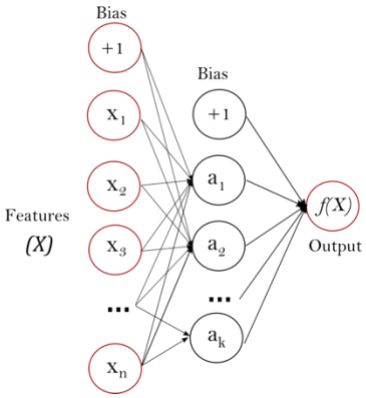

Figure 5 Multilayer Perceptron Architecture

The backpropagation training method allows for the adjustment of weights that connect nodes in a fully connected network exhibited by MLPs. This iterative adjustment is based on the discrepancy in error between the expected result and output during data processing [9]. Table 4 outlines the pros and cons of multilayer perceptrons.

Table 4 Pros and Cons of Multilayer Perceptrons

PROS	CONS
• capability to learn nonlinear models • capability to learn models in real-time	• MLP with hidden layers have a non-convex loss function where there exists more than one local minimum. • A large number of hyperparameters to tune, including the number of hidden layers and neurons, and epochs. • Sensitive to feature scaling

4. Hypothesis Statement

This study aims to compare and contrast the performance of a support vector machine and a multilayer perceptron for emotion recognition in text. The following hypotheses were made:

- The SVM will show greater predictive performance than that of the MLP, albeit moderately. SVMs and other learning-based algorithms have shown better results than their ANN counterparts on most popular benchmark problems [8]. Regarding text classification, Allouch et al. observed that the SVM gave a marginally higher prediction accuracy than that of the MLP model [11].
- SVMs perform effectively in high dimensional spaces, where the number of features is greater than the number of observations, as a result of the model's complexity of , compared to the MLP model with a complexity of . However, a large model size makes SVMs computationally time-consuming and expensive for real-time applications [6].
- Either model may not achieve high predictive performance due to indistinct boundaries between emotion classes and the absence of contextual analysis [1].

5. Evaluation Methodology

The dataset was divided into a training and test set, whereby 80% was towards training and validation of each model. The remaining 20% was used for testing and comparison of both SVM and MLP models. Validation of the models was done by a 3-fold cross-validation. A higher number of folds were considered but determined to be too computationally expensive. Model performance was assessed through the average of the validation accuracies.

Classification performance was evaluated by three commonly used metrics: accuracy, recall, and precision for both SVM and MLP models. One consideration to account for is that accuracy only works well in the context of balanced classification problems. Alone classification accuracy is typically not an informative account of the robustness of a model. As such, confusion matrices were also used to get a detailed understanding of the classification accuracy for each class. Additional measures of model performance such as receiver operating curves and learning curves were employed.

Several optimisation techniques were used to ascertain the best parameters and hyperparameters for each model that would yield the best predictive performance. For both SVM and MLP models, a random grid search followed by grid search was performed over various parameters and hyperparameters in MLP and hyperparameters for SVM. Random grid search was used to narrow down the search before employing a more computationally expensive grid search method. Support vector machine hyperparameters of concern were: C, the misclassification cost; kernel type, RBF, linear, and poly; degree, the degree of the polynomial kernel function, which is ignored by other kernels; and gamma, the kernel coefficient for RBF and polynomial kernels. The multilayer perceptron model's hyperparameters of concern were the number of hidden layers and neurons, optimiser, activation function of the hidden neurons, number of epochs, and training batch size. The number of input and output neurons was predetermined by the number of features and classes, respectively. Since this is a classification problem, the Softmax activation function was used at the output layer, and cross-entropy was used as a loss function.

6. Experimental Results

A support vector machine's good performance relies heavily on two parameters, the regularisation parameter or misclassification cost- which establishes the balance between minimisation of both training error and model complexity, and the kernel function- which implicitly delimits nonlinear mapping from input space to some high-dimensional feature space [12]. Furthermore, depending on the choice of kernel hyperparameters such as gamma in RBF, and degree for polynomial kernel needs to be accounted for. Since there was a sizeable search area considering the aforementioned (hyper)parameters, a random search was used as a base point prior to implementing a more computationally expensive grid search. The best SVM hyperparameters were a misclassification cost of 50, RBF kernel, and a gamma value of 0.01. The final SVM model's performance metrics are shown in Figure 6, where the final SVM model had an accuracy of 88%.

--- SVM CLASSIFICATION REPORT ---

	precision	recall	f1-score	support
sadness	0.90	0.94	0.92	1159
joy	0.88	0.92	0.90	1352
fear	0.85	0.86	0.85	475
anger	0.90	0.85	0.88	542
love	0.84	0.70	0.76	328
surprise	0.81	0.67	0.73	144
accuracy			0.88	4000
macro avg	0.86	0.82	0.84	4000
weighted avg	0.88	0.88	0.88	4000

--- MLP CLASSIFICATION REPORT ---

	precision	recall	f1-score	support
sadness	0.88	0.93	0.90	1159
joy	0.88	0.91	0.90	1352
fear	0.83	0.84	0.84	475
anger	0.88	0.84	0.86	542
love	0.78	0.68	0.73	328
surprise	0.82	0.58	0.68	144
accuracy			0.87	4000
macro avg	0.84	0.80	0.82	4000
weighted avg	0.86	0.87	0.86	4000

Figure 6 Classification report for SVM and MLP

For MLP, performance is largely dependent on the number of neurons in one or more hidden layers. Although a more significant number of neurons allows for mapping more complex relationships between input and output, this comes at the cost of high computation time and the possibility of overfitting. Conversely, too few neurons and hidden layers reduce the model's ability to capture the intricacies of relationships [13]. Parameters of concern for the MLP were the number of hidden layers and neurons, activation function, and optimiser, in addition to hyperparameters of drop out, batch size, epochs, and learning rate. Dropout was used since it has been shown to reduce overfitting significantly and improves performance [14]. Again, a random search was used as a base point prior to implementing a more computationally expensive grid search was used to find all (hyper)parameters except for the number of epochs. The number of epochs during training was set to 500, and early stopping was initiated when the accuracy of the model did not increase over ten sequential epochs. This was done to accelerate the training process and decrease the risk of overfitting [7]. The best performing MLP model contained one hidden layer of 100 neurons, two dropout layers (p=0.5), using Adam as an optimiser, ReLu as the activation function for hidden neurons, and the number of training epochs at 62. The performance metrics of the final MLP model is shown in Figure 6; where the model had an accuracy of 87%

7. Analysis and Critical Evaluation

As seen from the confusion matrices (Figure 7), love and surprise emotions have significantly lower accuracy than other emotions present; this coincides with significantly lower representation in the dataset at 8% and 3.5%, respectively. It can also be seen that love is often misclassified as joy and surprise as fear. This misclassification could be due to significant overlap in relevant words or lack of context. As hypothesised, even though the models reached accuracy in the high eighties, increased context could lead to increased model performance.

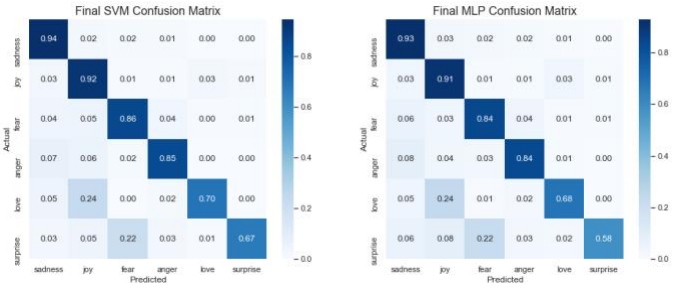

Figure 8 Confusion matrices for MLP and SVM Classifiers

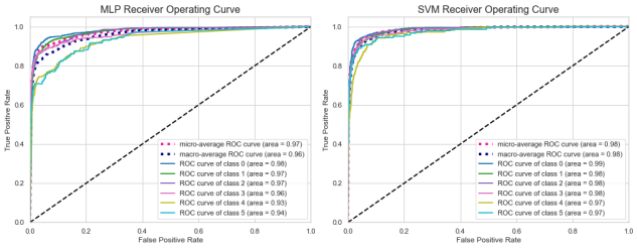

Figure 7 ROC Curves for MLP and SVM Classifiers

This misclassification of love and surprise could be remedied by incorporating further context analysis and theme extraction. In this study, unigrams and bigrams were both used. Unigrams do not offer enough specificity for phrase extraction and context. Instead, they are of most use when determining themes. Bigrams offer this specificity, allowing for some contextual understanding of key phrases and themes. Conversely, bigrams are prone to noise if stop words are present, and the high incidence is not always indicative of relevance to classification.

Receiver operating curves (ROC) and for both SVM and MLP were plotted (Figure 8). Each class is plotted in a one-versus-all methodology. From both models' ROC curves, extreme concavity is exhibited, resulting in a high area under the ROC (AUC). This high area shows a good measure of separability and ability to distinguish between classes [15] for both SVM and MLP models, with micro-averaged AUCs of 0.98 and 0.96, respectively. Similar to lower results in the confusion matrices, AUC for classes 4 (love) and 5 (surprise) are lower than other classes. Occurrences of near 1 AUC values are indicative of a model having "reasonable" quality [16], which is exhibited by both models. Although the average AUC for both models is comparable, marginally higher AUCs and, therefore performance, is displayed by the SVM model.

Model	Train/Predict	Time(s)
SVM	Train	248
	Predict	7.37
MLP	Train	60.6
	Predict	0.133

Table 5 Train and test times for SVM and MLP models

Additionally, from Table 5 it can be seen that the SVM model had a significantly higher training time compared to the MLP model. This aligns with the initial hypothesis that SVM's complexity and large model size would make it computationally expensive compared to the MLP model.

The higher predictive performance of the SVM over MLP seen could be attributed to several factors. Firstly, the SVM's advantageous optimisation methods [6]. Obtainment of support vectors by the SVM is through a convex optimisation problem, which yields a global minimum and unique solution. In the case of MLPs, there are local minima and solutions as a consequence of training via gradient decent- a non-convex method, thus possibly resulting in non-convergence to the optimal/global solution [6]. Another point of concern is each model's ability to handle class imbalance. MLPs, a form of artificial neural network, are less resilient to imbalanced and noisy data compared to SVMs [6]. Moraes et al. [6] performed document-level sentiment classification using an SVM and ANN with balanced and imbalanced versions of the same dataset. They found that while results between the two models were comparable, SVM outperformed ANN in the case of imbalanced data. These results are consistent with the performance seen by models in this study.

8. Conclusion

This study critically compared and contrasted the training methodologies and performance of MLP and SVM for text-based emotion recognition. The predictive accuracy of the SVM and MLP model was 88% and 87%, respectively. It was determined that the difference in the two models' predictive performance was not overly significant, with a difference in accuracy of approximately 1%. This performance is consistent with the initial

hypothesis that the SVM would marginally outperform the MLP. Parameter and hyperparameter optimisation were of crucial importance as it leads to significant performance increase for both models.

Future investigation should be done to derive contextual information and value. As such, feature engineering and text preparation will play a critical role in the continued performance increase of both models. Feature space should contain unigrams, bigrams, POS tags, and counts of sentiment words [17]. Desmet et al. [18] found that the most crucial features were trigram and lemma bag-of-words and subjectivity clues when performing emotion detection. In addition to the BOW and TF-IDF used in this study, latent semantic indexing could be implemented. This would allow for discovering the semantic association between terms and overcoming concerns produced by statistically derived conceptual indices, with added dimensionality reduction [19].

It has also been noted that text pre-processing has a significant effect on model performance. Further investigation into how the use or non-use of specific pre-processing steps such as stop-word removal, lemmatisation, and multiword grouping affects the model. Specifically, using a predefined stop word list could be detrimental, especially in emotion representation. For example, if a simple word such as 'not' is removed, this would result in false sentiment expression as a sentence is changed from "not happy" to "happy".

It has been seen that not only are both syntactic and semantic disambiguation required, but a shift from context-assisted techniques to context-centric is necessary for NLP to be successful. As such, both SVM and MLP are promising models for emotion classification when further contextual communication is considered.

9. References

[1] H. Binali, C. Wu, and V. Potdar, 'Computational Approaches for Emotion Detection in Text', *Th IEEE Int. Conf. Digit. Ecosyst. Technol.*, p. 7, 2010.

[2] F. A. Acheampong, C. Wenyu, and H. Nunoo-Mensah, 'Text-based emotion detection: Advances, challenges, and opportunities', *Eng. Rep.*, vol. 2, no. 7, Jul. 2020, doi: 10.1002/eng2.12189.

[3] A. Y. Liu, 'The Effect of Oversampling and Undersampling on Classifying Imbalanced Text Datasets'.

[4] C. Padurariu and M. E. Breaban, 'Dealing with Data Imbalance in Text Classification', *Procedia Comput. Sci.*, vol. 159, pp. 736–745, 2019, doi: 10.1016/j.procs.2019.09.229.

[5] A. Fernandez, S. Garcia, F. Herrera, and N. V. Chawla, 'SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary', *J. Artif. Intell. Res.*, vol. 61, pp. 863–905, Apr. 2018, doi: 10.1613/jair.1.11192.

[6] R. Moraes, J. F. Valiati, and W. P. Gavião Neto, 'Document-level sentiment classification: An empirical comparison between SVM and ANN', *Expert Syst. Appl.*, vol. 40, no. 2, pp. 621–633, Feb. 2013, doi: 10.1016/j.eswa.2012.07.059.

[7] C. M. Bishop, *Pattern recognition and machine learning*. New York: Springer, 2006.

[8] E. A. Zanaty, 'Support Vector Machines (SVMs) versus Multilayer Perception (MLP) in data classification', *Egypt. Inform. J.*, vol. 13, no. 3, pp. 177–183, Nov. 2012, doi: 10.1016/j.eij.2012.08.002.

[9] C. N. Kamath, S. S. Bukhari, and A. Dengel, 'Comparative Study between Traditional Machine Learning and Deep Learning Approaches for Text Classification', in *Proceedings of the ACM Symposium on Document Engineering 2018*, Halifax NS Canada, Aug. 2018, pp. 1–11, doi: 10.1145/3209280.3209526.

[10] S. Osowski, K. Siwek, and T. Markiewicz, 'MLP and SVM Networks – a Comparative Study', p. 4.

[11] M. Allouch, A. Azaria, R. Azoulay, E. Ben-Izchak, M. Zwilling, and D. A. Zachor, 'Automatic Detection of Insulting Sentences in Conversation', in *2018 IEEE International Conference on the Science of Electrical Engineering in Israel (ICSEE)*, Eilat, Israel, Dec. 2018, pp. 1–4, doi: 10.1109/ICSEE.2018.8646165.

[12] K. Duan, S. S. Keerthi, and A. N. Poo, 'Evaluation of simple performance measures for tuning SVM hyperparameters', *Neurocomputing*, vol. 51, pp. 41–59, Apr. 2003, doi: 10.1016/S0925-2312(02)00601-X.

[13] F. Itano, M. A. de Abreu de Sousa, and E. Del-Moral-Hernandez, 'Extending MLP ANN hyper-parameters Optimisation by using Genetic Algorithm', in *2018 International Joint Conference on Neural Networks (IJCNN)*, Rio de Janeiro, Jul. 2018, pp. 1–8, doi: 10.1109/IJCNN.2018.8489520.

[14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, 'Dropout: A Simple Way to Prevent Neural Networks from Overfitting', p. 30.

[15] Z. H. Hoo, J. Candlish, and D. Teare, 'What is an ROC curve?', *Emerg. Med. J.*, vol. 34, no. 6, pp. 357–359, Jun. 2017, doi: 10.1136/emermed-2017-206735.

[16] C. Marzban, 'The ROC Curve and the Area under It as Performance Measures', *WEATHER Forecast.*, vol. 19, p. 9, 2004.

[17] S. Sun, C. Luo, and J. Chen, 'A review of natural language processing techniques for opinion mining systems', *Inf. Fusion*, vol. 36, pp. 10–25, Jul. 2017, doi: 10.1016/j.inffus.2016.10.004.

[18] B. Desmet, 'Emotion detection in suicide notes', *Expert Syst. Appl.*, p. 8, 2013.

[19] S. K. Srivastava, S. K. Singh, and J. S. Suri, 'Effect of incremental feature enrichment on healthcare text classification system: A machine learning paradigm', *Comput. Methods Programs Biomed.*, vol. 172, pp. 35–51, Apr. 2019, doi: 10.1016/j.cmpb.2019.01.011.

10. Appendix

10.1 Glossary

Activation Function	A function (for example, ReLU or sigmoid) that takes in the weighted sum of all of the inputs from the previous layer and then generates and passes an output value (typically nonlinear) to the next layer.
AUC (Area under the ROC Curve)	An evaluation metric that considers all possible classification thresholds. Under the ROC curve, the area is the probability that a classifier will be more confident that a randomly chosen positive example is positive than a randomly chosen negative example.
Backpropagation	The primary algorithm for performing gradient descent on neural networks. First, the output values of each node are calculated (and cached) in a forward pass. Then, the partial derivative of each parameter's error is calculated in a backward pass through the graph.
Bag of Words	A representation of the words in a phrase or passage, irrespective of order. Each word is mapped to an index in a sparse vector, where the vector has an index for every word in the vocabulary.
Baseline	A model used as a reference point for comparing how well another model (typically, a more complex one) is performing
Batch	The set of examples used in one iteration of model training
Bigram	An N-gram in which N=2
Confusion Matrix	An NxN table that summarises how successful a classification model's predictions were; that is, the correlation between the label and the model's classification. One axis of a confusion matrix is the label that the model predicted, and the other axis is the actual label. N represents the number of classes.
Cross-Entropy	A generalisation of Log Loss to multiclass classification problems. Cross-entropy quantifies the difference between two probability distributions
Cross-Validation	A mechanism for estimating how well a model will generalise to new data by testing the model against one or more non-overlapping data subsets withheld from the training set.
Early stopping	A method for regularisation that involves ending model training before training loss finishes decreasing. In early stopping, you end model training when the loss on a validation dataset starts to increase, that is, when generalisation performance worsens.
Epoch	A full training pass over the entire dataset such that each example has been seen once. Thus, an epoch represents N/batch size training iterations, where N is the total number of examples.
Feature set	The group of features your machine learning model trains on
Feedforward neural network (FFN)	A neural network without cyclic or recursive connections
Fully Connected Layer	A hidden layer in which each node is connected to every node in the subsequent hidden layer.

Hidden layer	A synthetic layer in a neural network between the input layer (that is, the features) and the output layer (the prediction). Hidden layers typically contain an activation function (such as ReLU) for training
Hyperparameter	The "knobs" that you tweak during successive runs of training a model. For example, learning rate is a hyperparameter.
Layer	A set of neurons in a neural network that process a set of input features, or the output of those neurons.
Learning Rate	A scalar used to train a model via gradient descent. During each iteration, the gradient descent algorithm multiplies the learning rate by the gradient. The resulting product is called the gradient step.
Logistic Regression	A classification model that uses a sigmoid function to convert a linear model's raw prediction into a value between 0 and 1
Neuron	A node in a neural network, typically taking in multiple input values and generating one output value. The neuron calculates the output value by applying an activation function (nonlinear transformation) to a weighted sum of input values.
N-gram	An ordered sequence of N words.
One-vs-all	Given a classification problem with N possible solutions, a one-vs.-all solution consists of N separate binary classifiers—one binary classifier for each possible outcome.
Optimiser	A specific implementation of the gradient descent algorithm.
Sentiment analysis	Using statistical or machine learning algorithms to determine a group's overall attitude—positive or negative—toward a service, product, organisation, or topic.
Softmax	A function that provides probabilities for each possible class in a multiclass classification model. The probabilities add up to exactly 1.0.

Definitions are taken from Google Developer's Machine Learning Glossary

10.2 Implementation Details

Multilayer Perceptron and Support Vector Machines models were selected for this study. The dataset contains a collection of documents and multiclass responses in the form of associated emotion. From these properties, It is apparent that this is a classification problem and that regression models do not apply.

For SVM, an initial model was trained. The hyperparameters to be optimised from that model were C, kernel type, degree, and gamma. This optimisation was achieved through grid search implementation. The model's accuracy at varying hyperparameters was determined, and the combination yielding the lowest associated error was chosen. The model was cross-validated via k- fold cross-validation. Finally, the final classifier was trained with the optimised hyperparameters and later implemented.

The exact implementation was followed with the MLP model, although the choice of hyperparameters for investigation and later incorporation into the final classifier were the number of hidden layers and neurons, optimiser, activation function of the hidden neurons, number of epochs, and training batch size. Additionally, the number of layers and drop out layers was chosen without grid search. This was achieved through changes in the model architecture and determining the model's accuracy after training ten epochs.