Comparing Machine Learning Methods in Fake News Detection [*]

Krystal Dowling
914430

Charlotte Williams
834810

Laura Heard
994900

Yifan Luo
926261

Abstract

This report seeks to find a solution to the problem of fake news- an epidemic of articles feeding harmful and misleading disinformation to the public. Fake news detection was attempted with a Support Vector Machine model, a Convolutional Neural Network, and a Long Short-Term Memory Neural Network. The best performing classifier ended up being a LSTM-CNN model that had an accuracy of 90.25%, and broad generalisability to new data. This indicated that the data had both significant position-invariant features and context-dependant semantic relationships. The success of the research gives opportunity to extend on the models for effective fake news detection in the future.

1 Introduction

In a world of COVID-19 and an increasingly polarised and unstable political climate, fake news has become a real and credible threat to citizen liberty and health. It continues to disseminate insidious disinformation within a public sphere that increasingly turns to online social media for news [24][28][35].

The rise of fake news can be accredited to the fact that news can now be created and published online, thus bypassing the existing regulatory steps that bind traditional news media such as newspapers and television. Approximately 68% of Americans get their news from social media [34], where fake news runs rampant masquerading as reputable news. Within politics specifically, fake news is known to re-enforce confirmation bias of hyper-partisan views, encourage increased political polarisation, and undermines modern-day democracy [19][30].

The failure of regulatory oversight stems from the overwhelming volume of published content produced daily by online sources. Conventional regulatory techniques to audit new content, such as human fact checkers, present an expense and time inefficiency that is unfeasible for companies to contend with. A natural solution is through the use of machine learning techniques to automate the decision of truth and disinformation. Machines pose a scalable, efficient solution that may also present additional benefit through the reduction or elimination of human bias from news regulation.

This task of detecting fake news falls under the umbrella of natural language processing (NLP); more specifically, under text classification. NLP is a branch of machine learning which addresses the extraction of semantic and syntactic structure from human language [42]. It represents a complex problem; text is highly diverse and dimensional data [11] and is not easily quantified.

In this report, previous studies and results will be taken into consideration to compare statistical machine learning with deep learning techniques. Methods used were those that have been proven to be successful in previous studies; thus for statistical models, a Naïve Bayes classifier was used as a baseline [7] and Support Vector Machine (SVM) as the best performing statistical machine learning

[*] All code and associated visualisations for this paper can be found at https://github.com/krystaldowling/MAST30034_Final_Project-

Preprint. Under review.

model for fake news detection [7] [37]. Furthermore, two variants of deep learning were evaluated for this task: Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). CNNs are cited as being appropriate for longer text classification [2], and they have been found to be effective in NLP problems as they are able to utilise the presence or absence of features, words in this situation, as a distinguishing factor [2].

Similarly for RNNs, they have been shown to be effective for sequence-based predictions such as NLP tasks in the past [41]. A RNN's success in learning sequential data is a product of a chain-like structure underlying the model. However, the RNNs historically suffer from the vanishing gradient problem resulting in difficulties when learning long-term temporal dependencies in analysed sequences [25]. A long short-term memory (LSTM) seeks to resolve the RNN's 'short-term' memory through the use of gating that regulates flow of information. The intent is to enable the LSTM to hold onto information that spans broader lengths of the input sequences.

In order to accurately judge each model and their relevance within the context of society, accuracy, precision, recall, and the generalisability of each model have been considered as evaluation metrics.

2 Data

To better model the real-life application of fake new detection, multiple datasets from different sources have been included to train the classifiers. This is to ensure that the falseness of the news articles is being captured rather than the syntactical and semantic biases between sources. The aim of using several sources was to enable more generalisable models and consequently, to better reflect how well such models perform in an unsupervised environment.

The first four datasets listed were sourced from different authors on Kaggle. Examples of real and fake news gathered from Buzzfeed and PolitiFact were combined into datasets available from FakeNewsNet [36]. The 'Getting Real About Fake News' dataset contained fake articles gathered from across 244 websites [29]. The final Kaggle dataset 'Fake News' was itself made from a collection of other datasets found on Kaggle by the author, the sources of these were not disclosed [20]. To keep an even distribution of fake and real news, a sample of true news articles obtained from Signal media were also included [8].

3 Method

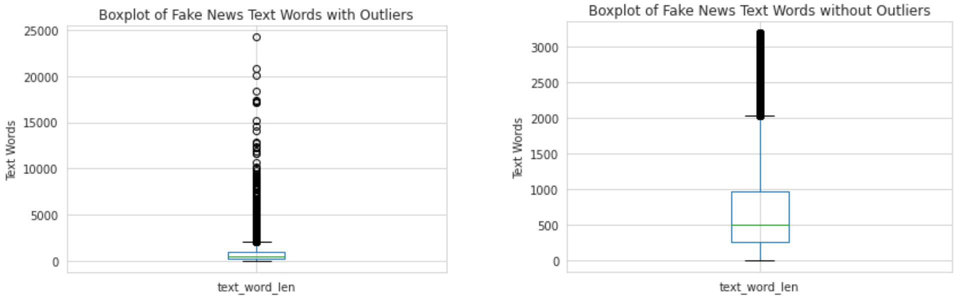
3.1 Pre-Processing

Data cleaning and pre-processing steps were performed in order to prepare the data for model training. Firstly, the attributes text and label were the only necessary features and all additional attributes were removed. Next, to clean the data, missing values and non-English instances were removed. Similarly, the dataset was verified for any duplicated records that may have accidentally been input or resulted from the varied datasets. There were 11,033 duplicated rows, indicating there may have been some significant overlap in the datasets. All duplicated rows excluding the first instance of each were removed.

Next, new attributes surrounding the word and letter length of the text were defined and global outliers based on word and letter length for the text were detected and removed. The effect of this can be seen in Figure 1a and 1b, showing an example of the data before and after outlier detection for the number of words in the text. It can be seen after removing the outliers, the distribution of the data became significantly less skewed.

The subsequent step was visually observing the data for any contextual outliers. All edge cases that were outside expected boundaries for title and text length were removed, such as restricting the number of words in the text of an article to greater than 5. Under the given circumstances of the dataset, any records lying out of that range were considered unreasonable for news articles and thus removed.

Furthermore, steps were taken to remove unnecessary confounding information. These steps included removing any punctuation or other unwanted phrases (such as newline symbols) from text, converting the text to lower case, and removing stop words as they do not provide any semantic meaning for the models to use in analysis.



(a) With Outliers

(b) Without Outliers

Figure 1: Outlier removal for number of words in text

Various pre-processing steps removed 14,683 instances from the data resulting in 29,476 data points for training.

3.2 Feature Engineering

A number of methods were used to generate feature embeddings that facilitate mapping of text to low-dimensional, learned continuous vector representations that capture meaningful relationships within language for classification [17][31].

3.2.1 TF-IDF

Term Frequency Inverse Document Frequency (TF-IDF) vector representations measure the importance of a word based on the number of times it appears in an article against the number of articles the word appears in [13].

$$w_{t,d} = \left(\frac{n_{t,d}}{\sum_k n_{t,d}}\right) log(\frac{N}{df_t})$$

Where t represents a term, d documents, N the total number of documents and df_t the document frequency of word t.

3.2.2 GloVe

GloVe embeddings refer to a pretrained weight matrix trained using an unsupervised learning algorithm. This seeks to use word-word co-occurrence probabilities to infer semantic relationships between words [27]. As with regular embedding approaches, GloVe represent the relationships of two words with linear substructures which capture the relationships across a vector. The result of this is the weight matrices can be used to infer a probabilistic likelihood of the association of two words. For example, the embedding assigns the words ice and solid occurring together a higher probability than ice and gas [27]. This shows that the GloVe embeddings are able to capture associations between words that intuitively reflect the human understanding of how semantic relationships between words are perceived.

For this project, the GloVe embeddings were trained on Wikipedia 2014 and Gigaword 5 [27]. It has 6 billion tokens and contained a vocabulary of 400 thousand words and 100-dimension vectors. This trained embedding was chosen as the vocabulary from this dataset would likely be the most similar to our dataset, in addition to having less tokens to reduce complexity.

3.2.3 Neural Network Embeddings

Unlike one-hot encoded vectors which are often high-dimensional with uninformed mapping, word embeddings used in neural networks are continuous vectors from learned low-dimensional representations of discrete data that are able to capture relationships within language [17] [23] [31]. While these dense vector representations can be pre-trained, such as with GloVe, they can also be learned within the neural network and thus tailored to the training data [4]. The input is required to be integer-encoded for this, which was done using Tokenizer [4]. Padding was also set to the maximum article length within the data to highlight words on the edge of the input matrix and improve the robustness of the model [40]. Once input into the embedding layer the neural network learns optimal weights for the tokenized data to minimise loss [4] [17].

3.3 Models

The baseline model for this research was a Multinomial Naïve Bayes classifier. This model was produced using the TF-IDF embeddings, and the 20,000 most significant word features as determined by the Chi-squared test.

3.3.1 Support Vector Machines

Support Vector Machine (SVM) classification was explored as a strong performing statistical machine learner for fake news detection. The SVM aims to find a hyperplane which maximises the distance between the decision boundary and the support vectors [9].

This model was trained separately with two different embedding techniques; TF-IDF and GloVe. After plotting the number of features against the model accuracy, the highest accuracy was achieved using the best 10,000 Chi-squared features. To generate the features using GloVe, each word was mapped to a GloVe embedding, with the mean of the word vectors used to represent each document. Since the attributes for the GloVe embeddings are the dimensions of a single vector, feature selection was not an option as removing any features would change the representation of the vectors. Grid search was performed to determine the best kernel to fit the data. The regularisation parameter was kept at its default of 1 to limit any misclassifications.

3.3.2 Convolutional Neural Network

CNN's comprise of several convolutional layers (a convolution is a mathematical combination of two relationships to produce a third relationship [6]) applied over the input layer with nonlinear activation functions applied to results, in this case ReLU, the rectified linear activation function [1][23].

The standard model starts with an input layer and an embedding layer [4]. Next, the convolutional layer applies different filters to the input, automatically learning the weights of its filters through back-propagation during training [3][14]. ReLU, used in the convolutional layer, is a linear function that aims to introduce non-linearities to the network and overcome the vanishing gradient problem common in other activation functions [1][4]. ReLU outputs zero for non-negative inputs, and returns the direct input otherwise: $g(z) = max(0, z)$ [1][4].

Further, a max-pooling layer was used to consolidate output while reducing the dimensional complexity and preserving salient information [6] [23] [32], it does this by only forwarding the maximum value from each feature map onto the next layer [14]. A flatten layer then converts the data into a 1-D vector [6][32].

Lastly, the final two dense layers are fully-connected layers that connect the nodes of the previous layer to the next [40]; the first uses the ReLU function, and the second the Sigmoid function, which maps to a value in $[0, 1]$ and is thus particularly useful as a final layer for this model as it is predicting the probability of classes [33].

This standard model was extended in the study to find the best performing CNN model, with iterations existing that utilised random search and a multi-channel CNN model.

3.3.3 Long Short-Term Memory

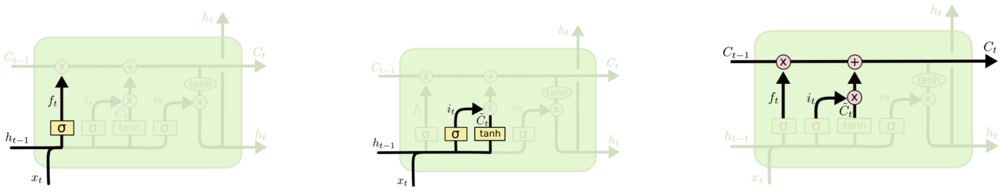
The LSTM model fundamentally consisted of a LSTM layer which receives input followed by a dense layer which integrates the activations of the LSTM layer to produce an output of the model. A common practice is often to include a proceeding embedding layer as was the case for CNNs. This was implemented for the base model in addition to using the GloVe pretrained embedding weights.

LSTM layers are made up of memory cells which each contain 3 gates; a forget gate, an input gate and an output gate. The three gates are relevant to the integration of previous information, the integration of current input and the relevance of the output to future processing respectively. Each makes use of sigmoid activation functions to scale the relevance of the information being processed to between 0 and 1. A number of these memory cells work in unison to extract information from input and to reconstruct it depending on its relevance to the output. This process is shown in Figure 2 [25].

Similar to the use case for CNNs, dropout layers were implemented by means of stochastic regularisation per training example [5].

The base LSTM model that has been assessed in this project consisted of two sequential LSTM layers, each of 20 memory cells, followed by 2 dense layers of 256 and 128 nodes respectively. Interspersed between the dense layers is a dropout layer with a dropout rate of 20%. Loss was assessed using binary cross entropy and optimisation was performed using the ADAM optimiser.

The LSTM-CNN model implemented for this task was a fused two simple CNN units and four stacked LSTM layers. The CNN units consisted of a CNN layer of 32 filters of 5x1 kernel size, a 20% dropout layer and a 4x1 MaxPool layer to downsize the data. The LSTM layers used were of 20 cells. GLoVe embeddings were used for the extended model too. The convolutional layers were implemented in an effort to extract the local patterns of text which then are passed to the LSTM layers to analyse broader dependencies [26].



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

(a) Forget Gate

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

(b) Input Gate

$$o_t = \sigma\left(W_o \ [h_{t-1}, x_t] \ + \ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

(c) Output Gate

Figure 2: LSTM Gates

4 Results

The Multinomial Bayes baseline model performed at an accuracy of 83.52%, with a precision of 89% and recall 70.70%.

4.1 Support Vector Machines

As shown in Table 1, the SVM achieved its highest accuracy with the TF-IDF embedding. The SVM model with GloVe performed worse than the baseline model in all accuracy metrics. The model with TF-IDF also had a significantly higher precision and recall than the other models.

Table 1: SVM Results

Embedding	SVM accuracy	Precision	Recall	F1 Score
TF-IDF	87.43%	88.31%	81.81%	84.93%
GloVe	76.92%	76.62%	68.40%	72.28%

4.2 Convolutional Neural Network

The two main CNN models was a standard CNN set-up, and a multi-channel CNN model with three parallel channels comprised of the standard set-up; both sets of results are shown in Table 2.

Table 2: CNN Results

CNN Model	Accuracy	Precision	Recall	F1 Score
Standard	86.98%	89.44%	80.93%	84.97%
Multi-channel	89.75%	88.92%	87.14%	88.02%

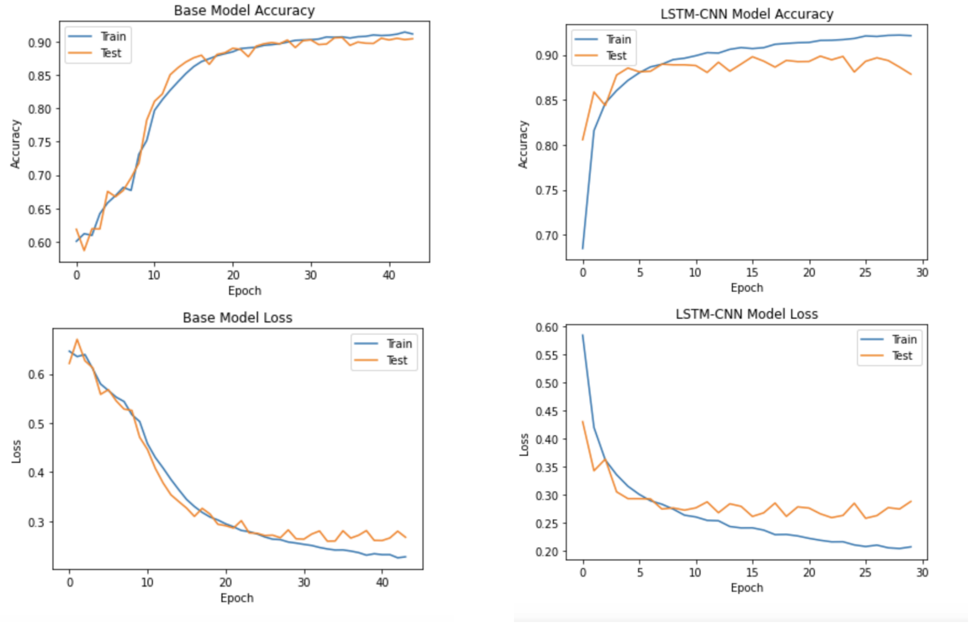
4.3 LSTM Neural Network

Table 3 shows the evaluation metrics of the two LSTM models. The LSTM-CNN model achieved 0.63% higher accuracy on average than the base LSTM model. The LSTM-CNN model also outperformed the base model's precision while they both resulted in the similar recall. The LSTM-CNN model produced a higher F1 score than the base model.

Table 3: LSTM Results

LSTM Model	Accuracy	Precision	Recall	F1 Score
Base	89.62%	89.61%	85.79%	87.66%
LSTM-CNN	90.25%	90.70%	85.79%	88.18%

The training of both models was stopped when there were 10 successive epochs of no improvement to validation loss.



(a) Base Model

(b) LSTM-CNN Model

Figure 3: Train and Test Accuracy and Loss of LSTM Models

Figure 3 shows the training and validation curves of the base LSTM model. It indicates a steady learning rate over the epochs with the average model epochs-to-peak accuracy of 48 epochs.

Figure 3 also shows the training and validation curves of the LSTM-CNN model. It demonstrates a much steeper learning curve, suggesting a faster convergence to a solution. The model's average model epochs-to-peak accuracy was 24 epochs.

5 Discussion

5.1 Support Vector Mechines

The final results demonstrate that the SVM model with TF-IDF embeddings and a linear kernel is the best performing statistical machine learner on the task of fake news detection, indicating the data is linearly separable. This result came as a surprise, given the GloVe embeddings capture the information of each article in a much more sophisticated manner compared to TF-IDF. The poor result of the SVM using GloVe could be attributed to the fact the embeddings were pretrained on an external dataset, rather than the training data[21]; or that by taking the mean of the word vectors losses too much information to adequately represent each article. The sucess of the LSTM models when using GloVe suggest the latter to be more likely. Another explanation for the weaker performance of GloVe embeddings may be due to the change in vector space, one that can not be separated as effectively by the SVM. This change in vector space is demonstrated by the different kernels which were found to have the best performance with these embeddings; a polynomial kernel for GloVe, compared to the linear kernel used for TF-IDF.

5.2 Convolutional Neural Network

Various convolutional neural network models were attempted in fake news detection, starting at a simple standard model and building on this model to the final best performing model; a multi-channel CNN. The first CNN attempted was a standard CNN modified from Janakiev (2020), with 128 filters and a kernel size of 5. This standard CNN had an accuracy of 86.98% at a batch size of 16 and with 6 epochs. The precision was a respectable 89.44%; however, the recall was lower at 80.93%, indicating the model is more inclined to make correct guesses of fake news rather than identify all fake articles.

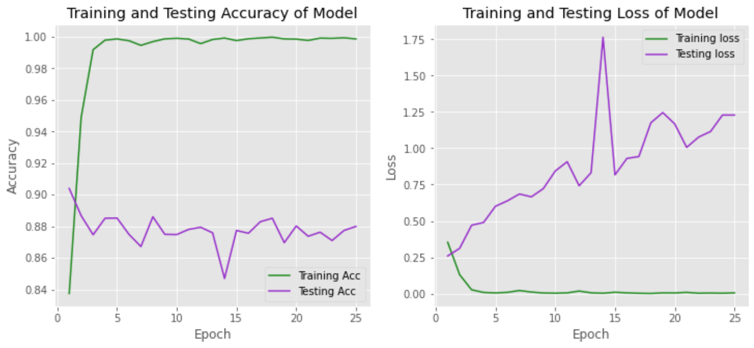
Next, random search with cross validation was utilised to optimise the number of filters (32, 64, 128), and kernel size (3, 5, 7) [14]. There were 4 folds of cross validation fitted for 5 iterations. The model peaked at 87.25% accuracy, with 64 filters and a kernel size of 3 as opposed to the standard model's 5. This is a very slight increase from the standard CNN model, likely indicating that the number of filters and kernel size do not have a huge influence on improving the accuracy of the model. However, the kernel size in the CNN for text classification defines the number of words, or n-grams, the convolution considers as a group [4]- thus the results indicating that variable kernel sizes can result in the best performing model could indicate that a variety of groupings of words are significant in building the model. This may be why the next model, a multi-channel CNN, performs best.

The next model created was a multi-channel CNN based on Yoon Kim's approach in his 2014 paper 'Convolutional Neural Networks for Sentence Classification' [16], which used multiple channels of the standard model with different sized kernels, thus allowing the data to be "processed at different resolutions or different n-grams (groups of words) at a time, whilst the model learns how to best integrate these interpretations", according to Brown (2017). The standard model differs slightly from the original created model, as it uses a dropout layer for regularisation to prevent initial batches of training data from disproportionately influencing the learning of the model [4].

The 3-channel CNN was fit with 32 filters in each channel and kernel sizes of 4, 6, and 8 respectively. The highest accuracy was 89.75% for the multi-channel CNN. Like the standard model, the multi-channel CNN had a higher precision than recall; however, the difference was much smaller than in the standard model, with a precision of 88.92%, recall of 87.14%, and F1-Score of 88.02%. This indicates a fairly effective model as they are all high values and the precision and recall are fairly balanced, as the F1-score indicates. A more precise model is also preferred over one with a higher recall for this task, as there is likely more associated risk with misidentifying a fake news story as true than incorrectly identifying a true news story as fake, as fake news tends to be more insidious and damaging when believed and encourages the spread of disinformation.

Despite this, the model is most accurate at low epoch values, finding its peak accuracy of 89.75% at 2 epochs and a batch size of 32. By analysing the models training and test accuracy and loss for large epoch values, such as at 25 epochs in figure 4, we can see that training accuracy reaches a value close to 100% fairly quickly (99.10% training accuracy for 2 epochs and 99.95% for 25 epochs). Further, the testing accuracy starts at a higher value and subsequently quickly decreases in early epochs before remaining fairly stable, if slightly decreasing over epochs. Similarly, while the training loss decreases to almost zero, the testing loss continues to increase as the model fits more data.

Figure 4: Train and Test Accuracy and Loss of Multichannel-CNN



This seems fairly indicative that the model is quickly over-fitting to the data, despite the added dropout layers to help prevent such [4]. This may be because the model is too complex for the problem and may be using too many filters on the data [40]. This further indicates that the model will likely not perform well when introduced to new data, as it is over-fitting to this dataset and is thus not very generalisable.

Despite this, the model still outperforms the SVM model explored above. While the standard model is fairly comparable to the SVM accuracy of 87.44%, the multi-channel model does give a higher accuracy, precision, recall, and F1-score. This could be due to the CNNs ability to utilise the presence or absence of features as a factor [2], and the SVM's inability to learn structure or order in text [39]. Further, the word embeddings in neural networks allow the CNN to recursively build more complex text representations that benefit the model [39].

The success of the multi-channel model is likely as the ability to perform parallel analysis on multiple n-gram features enhances the model's ability to interpret the different contributions of words to the semantic relationships of the text [22] [18]. However, the fact that these words have a significant semantic relationship relevant to their location in the text strongly indicates that a recurrent neural network may be highly effective on this data.

5.3 LSTM Neural Network

The base LSTM model performed more than 2% better than the SVM model on the validation set. The difference can be accounted for by both the difference in the approaches of SVM models and LSTM models to separating the classes and the nature of the data with which each works. The LSTM model seeks to learn the features of the vector space which define fake news whereas the SVM attempts to find a decision boundary in the vector space separating the two classes. As fake news becomes increasingly sophisticated it becomes progressively indistinguishable from real news. This overt mimicry renders it difficult to define distinct boundaries to be drawn between classes. As such, where the SVM may find it difficult to draw a clear hyperplane, the LSTM is able to focus more on the patterns and features of fake news in the vector space [26].

Further, the models differ in that the SVM has strongly supervised feature selection whilst the LSTM is given the opportunity for unsupervised and internalised feature selection. In the case of the LSTM model, the feature selection is refined during network training through weight updates that determine each feature's relevance, as opposed to the method of feature selection used in SVM. It has been shown that automated feature extraction is preferred to hand crafted features [12] as they allow absorption of a greater degree of context in the selection of the relevant features for a given training example. The LSTM and the LSTM-CNN models show a marginal difference across all evaluation metrics. Figure 3a and 3b show that both models demonstrate great stability upon achieving their maximum accuracies. This suggests both are resistant to overfitting and may be used as a surrogate to suggest appropriate generalisation has occurred.

However, the greatest difference between the networks' performance lay in the time they take to converge to a solution. Figure 3 shows it takes the base model approximately 20 epochs to reach an accuracy of 85% whereas the LSTM-CNN takes 3 epochs. The convolutional layer extracts smaller patterns more effectively while the LSTM layer looks for long term temporal dependencies arising across the processed text. This approach may have yielded a rapid acquisition of recurrent pattern allowing the model to quickly converge to a solution. However, the CNN component of the model may also be an inhibiting factor in the eventual identification of long-term trends as it may too readily dismiss broader dependencies in its analysis of local and smaller scale relationships.

5.4 Analysis between Models

Given that GloVe embeddings were so successful for the LSTM, they were also attempted on the best performing CNN model. The GloVe embeddings resulted in a significant drop of testing accuracy, from 89.75% to 83.25% for the multi-channel CNN. This is likely as the embeddings in the multi-channel CNN without GloVe are better able to be tailored to the channels [4] and capture salient relationships within the text [17][31]. However, on the flip-side, this is likely part of what is contributing to the multi-channel CNN overfitting. This is evidenced in the fact that the GloVe embeddings for the multi-channel CNN do not result in the same degree of overfitting; the training accuracy has a much slower rise to its peak and the test accuracy does not decrease in later epochs. Similarly, the GloVe embeddings working so effectively on the LSTM-CNN model are likely contributing to the model not overfitting to the training data like the CNN and therefore making it more generalisable as a model.

The fact that the CNN performs well indicates that the data does have some salient position-invariant features that largely contribute to whether an article can be classed as 'fake news' or not, such as certain phrases in articles [10]. However, the success of the LSTM models also indicate that context-dependent "long-range semantic dependenc[ies]" are very significant to fake news detection as well [10]. Thus, is it unsurprising that the combined CNN-LSTM model, which is able to both extract spatial features with the CNN layer and utilise contextual information with the LSTM, is the best performing classifier [15].

6 Conclusion

Analysis of the models indicated that the data had both salient spatial features and significant contextual semantic dependencies, as both CNN and LSTM models performed well. This was further evidenced in the best performing model; the LSTM-CNN model, as it had high values in evaluation metrics, specifically accuracy and precision, which is preferable for this task as it places more importance on restricting the spread of disinformation. Moreover, this model proved the most generalisable and is therefore ideal for future use in fake news detection.

To follow up the findings of this paper, future research could be fitting models that improve on the problems of the models implemented within this report. As the CNN model was prone to overfitting, a CNN-based residual network would be a natural extension as it is especially resilient to overfitting [32]. Further, seq2seq models have demonstrated strength in extracting semantic meaning beyond simple lexical and syntactic analysis [38].

References

[1] A. Amidi and S. Amidi. *Convolutional Neural Networks cheatsheet*. 2020.

[2] F.T. Asr and M. Taboada. “The global rise of “fake news” and the threat to democratic elections in the USA”. In: *Public Administration and Policy: An Asia-Pacific Journal* 22.1 (2019), pp. 15–24. DOI: https://doi.org/10.1108/PAP-04-2019-0008.

[3] D. Britz. *Understanding Convolutional Neural Networks for NLP*. 2015. URL: http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/. (accessed: 18.09.2020).

[4] J. Brownlee. *How to Develop a Multichannel CNN Model for Text Classification*. 2018. URL: https://machinelearningmastery.com/develop-n-gram-multichannel-convolutional-neural-network-sentiment-analysis/. (accessed: 13.09.2020).

[5] Gaofeng Cheng et al. “An Exploration of Dropout with LSTMs”. In: Aug. 2017, pp. 1586–1590. DOI: 10.21437/Interspeech.2017-129.

[6] V. Choubey. *Text classification using CNN*. 2020. URL: https://medium.com/voice-tech-podcast/text-classification-using-cnn-9ade8155dfb9. (accessed: 20.10.2020).

[7] N. Conroy, V. Rubin, and Y. Chen. “Automatic Deception Detection: Methods for Finding Fake News”. In: Oct. 2015. DOI: https://doi.org/10.1002/pra2.2015.145052010082.

[8] David Corney et al. “What do a Million News Articles Look like?” In: *Proceedings of the First International Workshop on Recent Trends in News Information Retrieval co-located with 38th European Conference on Information Retrieval (ECIR 2016), Padua, Italy, March 20, 2016.* 2016, pp. 42–47. URL: http://ceur-ws.org/Vol-1568/paper8.pdf.

[9] Rohith Gandhi. *Support Vector Machine — Introduction to Machine Learning Algorithms*. 2018. URL: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47.

[10] S. Ghelani. “Text Classification — RNN’s or CNN’s?” In: *towards data science* (2019).

[11] Suresh Babu Golla. “Challenges Of Implementing Natural Language Processing”. In: (2020). URL: https://analyticsindiamag.com/challenges-of-implementing-natural-language-processing/.

[12] Aditi Gupta et al. *TweetCred: Real-Time Credibility Assessment of Content on Twitter*. 2015. arXiv: 1405.5490 [cs.CR].

[13] Yassine Hamdaoui. *TF(Term Frequency)-IDF(Inverse Document Frequency) from scratch in python*. 2019. URL: https://towardsdatascience.com/tf-term-frequency-idf-inverse-document-frequency-from-scratch-in-python-6c2b61b78558.

[14] N. Janakiev. *Practical Text Classification With Python and Keras*. 2020. URL: https://realpython.com/python-keras-text-classification/#convolutional-neural-networks-cnn. (accessed: 06.10.2020).

[15] Beakcheol Jang et al. “Bi-LSTM Model to Increase Accuracy in Text Classification: Combining Word2vec CNN and Attention Mechanism”. In: *Applied Sciences* 10.17 (2020), p. 5841. ISSN: 2076-3417. DOI: 10.3390/app10175841. URL: http://dx.doi.org/10.3390/app10175841.

[16] Yoon Kim. “Convolutional Neural Networks for Sentence Classification”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1746–1751. DOI: 10.3115/v1/D14-1181. URL: https://www.aclweb.org/anthology/D14-1181.

[17] W. Koehersen. “Neural Network Embeddings Explained”. In: *towards data science* (2018).

[18] P. Kumar. “An Introduction to N-grams: What Are They and Why Do We Need Them?” In: *XRDS* (2017).

[19] T. Lee. “The global rise of “fake news” and the threat to democratic elections in the USA”. In: *Public Administration and Policy: An Asia-Pacific Journal* 22.1 (2019), pp. 15–24. DOI: https://doi.org/10.1108/PAP-04-2019-0008.

[20] William Lifferth. *Fake News*. 2016. URL: https://www.kaggle.com/c/fake-news/data.

[21] Tom Lin. *[NLP] Performance of Different Word Embeddings on Text Classification*. 2019. URL: https://towardsdatascience.com/nlp-performance-of-different-word-embeddings-on-text-classification-de648c6262b.

[22] Z. Liu et al. *Multichannel CNN with Attention for Text Classification*. 2020. arXiv: 2006.16174 [cs.CL].

[23] M.M. Lopez and J. Kalita. *Deep Learning applied to NLP*. 2017. eprint: 1703.03091.

[24] G. Nyilasy. "Fake News in the Age of COVID-19". In: *Inside Business* (2020).

[25] Christopher Olah. "Understanding LSTM Networks". In: (2015). URL: `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`.

[26] Ray Oshikawa, Jing Qian, and William Yang Wang. *A Survey on Natural Language Processing for Fake News Detection*. 2020. arXiv: `1811.00770 [cs.CL]`.

[27] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation". In: *In EMNLP*. 2014.

[28] A. Rajan. "Fake news: Too important to ignore". In: *BBC* (2017).

[29] Meg Risdal. *Getting Real about Fake News: Text & metadata from fake & biased news sources around the web*. 2016. URL: `https://www.kaggle.com/mrisdal/fake-news`.

[30] M. Rosenwald. "Making media literacy great again". In: *Columbia Journalism Review* (2017).

[31] R. Ruizendaal. "Deep Learning 4: Why You Need to Start Using Embedding Layers". In: *towards data science* (2017).

[32] S. Saha. "A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way". In: *towards data science* (2018).

[33] S. Sharma. "Activation Functions in Neural Networks". In: *towards data science* (2017).

[34] Elisa Shearer and Katerina Eva Matsa. "News Use Across Social Media Platforms 2018". In: (2018). URL: `https://www.journalism.org/2018/09/10/news-use-across-social-media-platforms-2018/`.

[35] K. Shu et al. "Fake News Detection on Social Media: A Data Mining Perspective". In: *SIGKDD Explor. Newsl.* 19.1 (2017), pp. 22–36. DOI: `https://doi.org/10.1145/3137597.3137600`.

[36] Kai Shu et al. "FakeNewsNet: A Data Repository with News Content, Social Context and Dynamic Information for Studying Fake News on Social Media". In: *arXiv preprint arXiv:1809.01286* (2018).

[37] N. Smitha and R. Bharath. "Performance Comparison of Machine Learning Classifiers for Fake News Detection". In: *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*. 2020, pp. 696–700. DOI: `10.1109/ICIRCA48905.2020.9183072`.

[38] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. *Sequence to Sequence Learning with Neural Networks*. 2014. arXiv: `1409.3215 [cs.CL]`.

[39] R. Trusov. *Why would any one use Recursive Neural Nets for text classification as against SVM or Naive Bayes or any traditional statistical models?* 2016. URL: `https://www.quora.com/Why-would-any-one-use-Recursive-Neural-Nets-for-text-classification-as-against-SVM-or-Naive-Bayes-or-any-traditional-statistical-models`. (accessed: 22.10.2020).

[40] A.G. Walters. *Convolutional Neural Networks (CNN) to Classify Sentences*. 2019. URL: `https://austingwalters.com/convolutional-neural-networks-cnn-to-classify-sentences/`. (accessed: 25.10.2020).

[41] Wenpeng Yin et al. *Comparative Study of CNN and RNN for Natural Language Processing*. 2017. arXiv: `1702.01923 [cs.CL]`.

[42] Diego Lopez Yse. *Your Guide to Natural Language Processing (NLP)*. 2019. URL: `https://towardsdatascience.com/your-guide-to-natural-language-processing-nlp-48ea2511f6e1`.