Adding Time-series Data to Enhance Performance of Naural Language Processing Tasks

by

Jingtian Zhao
Lakehead University

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER

in the Department of Computer Science

Lakehead University

Adding Time-series Data to Enhance Performance of Naural Language Processing Tasks

by

Jingtian Zhao
Lakehead University

Supervisory Committee

Dr. Yimin Yang, Supervisor
(Department of Electrical and Computer Engineering, engineering, University of Western Ontario, Canada)

Dr. Ruizhong Wei, Co-Supervisor
(Department of Computer Science, Lakehead University, Canada)

Dr. Will Zhao, External Member
(Stratford School of Interaction Design and Business, University of Waterloo, Canada)

ABSTRACT

In the past few decades, with the explosion of information, a large number of computer scientists have devoted themselves to analyzing collected data and applying these findings to many disciplines. Natural language processing (NLP) has been one of the most popular areas for data analysis and pattern recognition. A significantly large amount of data is obtained in text format due to the ease of access nowadays. Most modern techniques focus on exploring large sets of textual data to build forecasting models; they tend to ignore the importance of temporal information which is often the main ingredient to determine the performance of analysis, especially in the public policy view. The contribution of this paper is three-fold. First, a dataset called COVID-News is collected from three news agencies, which consists of article segments related to wearing masks during the COVID-19 pandemic. Second, we propose a long-short term memory (LSTM)-based learning model to predict the attitude of the articles from the three news agencies towards wearing a mask with both temporal and textural information. Then we added the BERT model to further improve and enhance the performance of the proposed model. Experimental results on the COVID-News dataset show the effectiveness of the proposed LSTM-based algorithm.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to thank:

First of all **Dr. Yimin Yang**, for providing me with the opportunity to work under his supervision and for supporting me through all the tough times. It was not an easy journey, but I have been very fortunate to have a supervisor who cared this much about my work. His guidance, constructive suggestions, and encouragementare the reason I was able to learn and grow as a researcher. It was an absolute privilege to work with him on this research.

Dr. Will Zhao, for his patience and support in overcoming numerous obstacles in building a new dataset and my thesis writing.

Dr. Ruizhong Wei, for his constructive suggestions that helped me in polishing my thesis.

Faculty of Graduate Studies & Faculty of Science and Environmental studies for their financial support. This research would not have been possible without it.

Chapter 1

Introduction

1.1 Overview

The focus of this thesis is on Natural Language Processing (NLP) techniques within the broader field of Artificial Intelligence (AI) that are used for analyzing text and categorizing it. AI is a field that combines multiple disciplines to develop algorithms and systems capable of performing tasks that typically require human intelligence [15]. This field includes a variety of techniques and methodologies, such as machine learning, that are used to create intelligent systems.

Essentially, AI involves a wide array of techniques focused on creating intelligent machines. Machine learning is a sub-field of AI and deep learning is a branch of machine learning approaches that utilized multi-layer neural networks. And Natural language processing models are built on those approaches. All those researches are meant to help human to accelerate the task processing time and reduce the required human effort.

As a subfield of AI, machine learning focuses on creating algorithms that can learn patterns from data without being explicitly programmed [16]. Machine learning techniques can automatically adapt and improve their performance as they are exposed to more data. To be more specific, it focuses on building methods to let machines can 'learn' from data and has been an active area of research for several decades. Such methods can take advantage of collected data from the past to improve performance by solving a set of tasks in the future. It has been considered a rapid expansion in recent years. One of the earliest works in this field is the development of decision trees, which Ross Quinlan introduced in 1986[1]. Since then, several other machine learning algorithms have been developed, such as support vector machines[2], neural networks[3], and random forests[4]. These algorithms have been successfully applied to a wide range of domains, including image recognition[5], speech recognition[6], and natural language processing[7]. Examples of machine learning applications include recommendation systems, spam filters, and fraud detection [17].

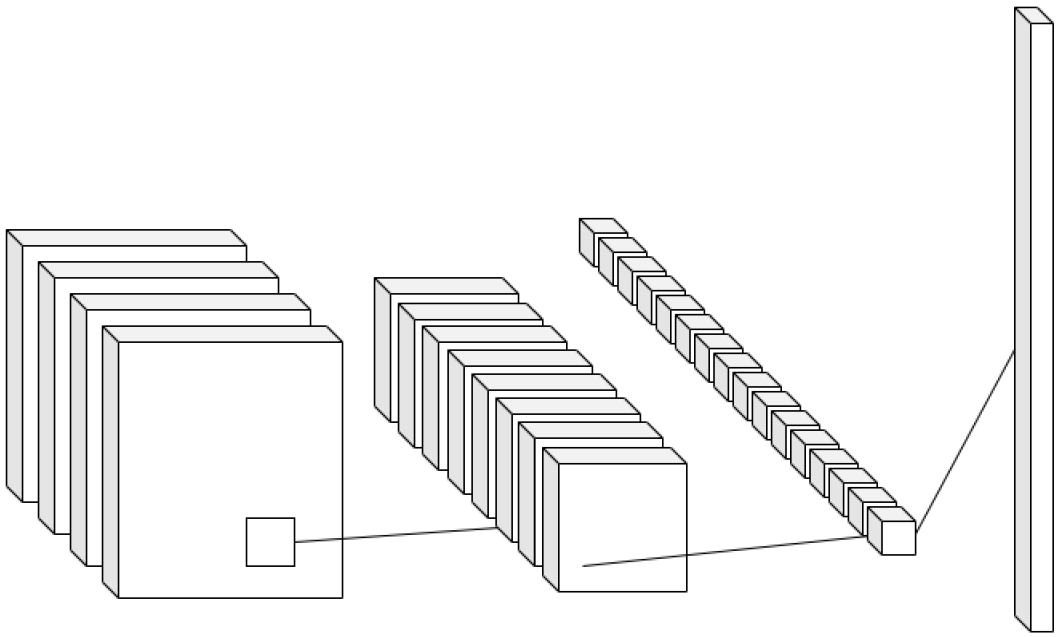

Figure 1.1: **Example of Neural Network**

Deep learning which is a subfield of machine learning employs artificial neural networks with multiple layers to model and solve complex problems [18]. These deep networks can automatically learn and extract features from raw data, eliminating the need for manual feature engineering. Deep learning has been particularly successful in fields such as image recognition, speech recognition, and natural language processing

[19].

Natural Language processing is one of the machine learning tasks that mainly focuses on mining information from textual data and using that information to helping to analyze more data or make predictions of a future event. It is concerned with the mutual effect between computer and human languages, deals with the interaction between computers and human languages [20], and the main task is to program computers to mine information from a large amount of human language data. It focuses on enabling machines to understand, interpret, and generate human language in a way that is both meaningful and useful. NLP techniques can be built upon both machine learning and deep learning approaches. The ultimate goal of NLP is to make computers "understand" documents that consist of human language and can precisely extract meaningful information and be capable of categorizing those documents. Common NLP tasks include text classification, sentiment analysis, machine translation, and question-answering systems [21].

The field of natural language processing has also seen significant growth in recent years. One of the most influential works in this field is the development of the WordNet lexical database[8], which provides a structured hierarchy of words and their relationships. Another key development was the introduction of the bag-of-words model[9], which represents documents as vectors of word frequencies. More recently, deep learning techniques such as recurrent neural networks and transformer models have led to significant improvements in tasks such as machine translation and language modeling[10].

The number of applications that utilize it makes NLP an essential tool. For example, virtual assistants such as Siri and Alexa use NLP to understand user requests and provide appropriate responses. Social media platforms use NLP to analyze user-generated content and personalize recommendations. In healthcare, NLP is used to extract information from electronic health records and assist with clinical decision-making[11]. These applications across different domains take advantage of NLP to extract valuable information from text data.

However, ethical concerns related to bias and challenges related to understanding complex language structures remain significant challenges that require further research. In this paper, we will review these recent advancements and challenges in NLP in more detail. The performance of NLP models is constantly being improved by researchers through the use of advanced algorithms and models, deep learning techniques, and training on larger and more diverse datasets [12, 13].

One of the most significant advancements In NLP in recent years has been the development of transformer-based models, such as BERT and GPT-3. These models have set new benchmarks in language processing and have improved the accuracy of NLP models in various applications. Additionally, researchers have been exploring different training techniques to improve the performance of these models, such as unsupervised pre-training and domain adaptation [13].

Another area of active research in NLP is sentiment analysis, where the goal is to determine the emotional tone of text data. Deep learning techniques, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have significantly improved the accuracy of sentiment analysis models. Furthermore, researchers have been exploring transfer learning approaches to train sentiment analysis models on larger and more diverse datasets.

In addition to performance improvements, ethical concerns related to bias in NLP models remain a significant challenge. Researchers are actively working to address these concerns by developing fair and unbiased models and datasets. For example, Google recently released a new benchmark dataset for toxic language detection, which includes examples from different demographics and languages to address issues of bias and fairness [14].

Despite the significant progress made in NLP research, there are still several challenges that need to be addressed. For example, NLP models often struggle with understanding and interpreting sarcasm, irony, and humor in the text. Furthermore, there is a need to develop NLP models that can understand and process multiple languages simultaneously, given the increasing demand for multilingual applications.

1.2 Motivation

The recent research on NLP has mainly focused on training on the massive amount of data to acquire decent performance. However, in real-world applications, sometimes it is very difficult to collect sufficient trainable data which cause impossible to apply NLP to the problem. Essentially, NLP is a technique based on machine learning which is built on "learning" from given data and uses those "learned knowledge" on the rest of the data or future data to make predictions. Having an enormous trainable dataset would always be helpful and ideal to improve the performance of any NLP models, but the amount of time and human effort that is required to collect data in certain fields makes it nearly impossible to have enough data to get decent

performance. Therefore, how to efficiently use limited hard-to-collect data should be a bigger concern when trying to use NLP to solve problems in those fields.

In this thesis, we have discovered that time-series data could be an effective factor when training NLP models in certain fields. For example, the COVID-19 pandemic has brought unprecedented challenges to public health and policy-making.

Traditional methods including surveys and polls are useful tools but are limited by sample size, bias and etc. Thus NLP is introduced to analyze public sentiment from larger sets of data such as social media, news articles.

In particular, the incorporation of time-series data in NLP analysis has been shown to improve the accuracy and robustness of the models, as temporal patterns and trends in language use can provide important insights into public sentiment and behavior over time [22]. However, despite the potential of NLP models to inform public health interventions and policy decisions during a crisis, there has been relatively little research on the use of NLP techniques to analyze public attitudes towards mask-wearing during the COVID-19 pandemic [23].

This thesis aims to address this gap by analyzing public attitudes toward mask-wearing using a combination of textual and time-series data. By incorporating temporal information in our analysis, we aim to provide a more nuanced understanding of public sentiment towards mask-wearing over time. Our study has the potential to inform public health interventions and policy decisions aimed at promoting compliance with mask-wearing and increasing public trust in public health measures during the COVID-19 pandemic. In addition, other public datasets have been tested using the proposed model to compare the performance.

Overall, this thesis aims to contribute to the growing body of research on using NLP techniques to analyze public sentiment and behavior during a crisis or other major event. Our study highlights the importance of incorporating time-series data in NLP analysis and demonstrates the potential of NLP models to inform public health interventions and policy decisions during the COVID-19 pandemic[24] and other events.

1.3 Problem Description

NLP research has made significant progress and has been a successful tool in many fields and has helped many people in a variety of ways, but there are still multiple challenges that await to be addressed. For instance, sarcasm, irony, and humor in

the text are often confusing NLP models and those linguistic features are preventing the application in social media and online platforms. However, they are all key components of textual data for NLP applications. Additionally, text data that contains multiple languages simultaneously also significantly reduce the performance of NLP models.

Thus, researchers have been putting effort into solving those addressed challenges. Among all the research achievements, Convolutional neural networks (CNN) and recurrent neural networks (RNN) have shown promising results in capturing the nuances of language use [26]. Another approach that is often used to overcome those challenges is cooperating with different knowledge sources, which can provide additional context to the problem domain and improve the overall performance of NLP models [27]. In this thesis, time-series data has been chosen to cooperate with textual data to enhance the performance of the proposed model.

Despite these advancements, there is still a need for further research to address the challenges posed by sarcasm, irony, humor, multilingualism, and many other things that can interfere with the performance of trained models in NLP.

1.4 Contribution

In this thesis, the contribution is two-fold, first, a well-organized dataset is collected and called COVID-News and another is proposed a NLP model that can utilize time-series data to cooperate with textual data to improve the data utilization and thus improve the performance of the proposed model. In the following chapters. Two models have been proposed the first one has a simpler structure and is used to test the viability of combining time-series and textual data. However, the second one is well-refined has a more complex structure, and is meant to further improve the performance of the proposed method with cooperating time-series data.

1.5 Organization of this Thesis

This section was all about the introduction and the rest of the thesis proceeds as follows,

Chapter II gives detailed insight into the background for the various techniques used in this thesis and then discusses the different research works related to this thesis

Chapter III, includes the details about the initial considered problem and then further explains the possible way to solve addressed problems. Then, a dataset that is called COVID-News will be used to benchmark the performance of the initial proposed model. The initial proposed model is based on the **long short-term memory networks (LSTM)** and **Text Preprocessing** techniques to solve the problem defined in the previous chapter.

Chapter IV explains the second proposed model which works using the **Bidirectional Encoder Representations from Transformers** to further improve the performance of the proposed model and it will be tested on other public datasets.

Chapter V is the last chapter in this thesis, which concludes the whole work done in this thesis and further explains the future prospects of the research done.

Chapter 2

Background and Related Work

2.1 Background

Time-series data is one specific data collected over time, where each data point represents a specific moment in time [28]. Stock Prices, weather data, sensor data, and news articles all possess data elements that can be used to form time-series data. Other types of data is collected at a single time point which separates them from time-series data.

One of the features of time-series data is its temporal dependency. This means that each data point's value depends on the value of the previous data points [33]. The data of a given moment is not isolated but connected to both forward and backward.

And assuming each data is independent of the other is the reason why traditional machine learning techniques can not handle time-series data.

Specialized machine learning techniques have been developed specifically for time-series data to address this issue. These techniques include autoregressive models, moving average models, and other methods that take into account the temporal dependencies in the data [28].

Another application of time-series analysis is anomaly detection, where the goal is to identify unusual patterns or events in the data [33]. This is useful in fields such as cyber security, where abnormal network traffic patterns can indicate a potential security breach. Time-series analysis is also used in fault detection and predictive maintenance. Historical data is used to predict when a machine is likely to fail, allowing maintenance to be performed before a failure occurs [28].

Time-series analysis is not only limited to those applications but can also be applied to many other fields. In healthcare, time-series analysis can be used to monitor patient vital signs and to predict the likelihood of certain medical conditions [28].

Based on the given examples, time-series data is a valuable tool for understanding trends and patterns over time, and specialized machine learning techniques have been developed to make the most of this data. The applications of time-series analysis are diverse. The importance of Time-series will increase as more time-series data is collected. Therefore, this thesis has explored the possibility of combining time-series data with other types of data to improve the performance of machine learning models.

2.2 Related Works

2.2.1 Random Forest With Time-series Data

Random Forest is an ensemble learning technique that builds multiple decision trees and merges their predictions for a more accurate and robust output. It has been widely used in various fields such as image classification, natural language processing, and fraud detection due to its excellent performance and ease of use[29]. When it comes to time series data, Random Forest can be used effectively by adapting it to suit the unique characteristics of this data type[30].

Time series data represents a sequence of observations, measured at equidistant time intervals. The goal of time series forecasting is to predict future values based on historical data. The inherent temporal dependencies in time series data pose unique

challenges to traditional machine learning methods like Random Forest, which were not designed explicitly to deal with such dependencies [31].

A method for leveraging Random Forest with time series data is the use of Recursive Feature Elimination (RFE). This technique helps in selecting a subset of relevant features for the final model, improving the performance and reducing overfitting. By iteratively removing less important features, RFE allows the Random Forest model to focus on the most important predictors, which enhances the overall prediction accuracy.

Additionally, to further improve the performance of Random Forest on time series data, ensemble approaches like Bagging (Bootstrap Aggregating) and Boosting can be utilized. These techniques are designed to create more diverse and accurate predictors by combining multiple Random Forest models, thereby reducing the variance and improving the stability of the overall model.

Although Random Forest was not initially designed for time series data, several adaptations and enhancements can be employed to make it a powerful tool for time series forecasting. By incorporating temporal dependencies through lagged variables, using feature selection methods like RFE, and leveraging ensemble approaches like Bagging and Boosting, Random Forest can effectively handle time series data for improved forecasting accuracy.

2.2.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a class of artificial neural networks specifically designed for processing sequences of data, making them particularly well-suited for time-series data analysis [36].

In contrast to traditional feedforward neural networks, RNNs exhibit a cyclical structure, wherein the output from a previous time step feeds into the input of the current time step. This recurrent connection facilitates the retention of information from prior inputs, allowing the network to learn temporal patterns and dependencies within the data. RNNs can be used for a variety of tasks, such as time-series forecasting, anomaly detection, and sequence-to-sequence learning.

One of the primary challenges faced by RNNs is the vanishing gradient problem, which occurs when the gradients of the loss function become too small, leading to slow or ineffective learning [35]. This issue is particularly pronounced in RNNs processing long sequences, as the gradient signal can become too weak to propagate effectively

through the network. To address this problem, researchers have developed variants of RNNs, such as Long Short-Term Memory (LSTM) networks [36] and Gated Recurrent Units (GRUs) [39], which incorporate specialized gating mechanisms to better maintain and propagate gradients over long sequences.

In the context of time-series data, RNNs can be employed to forecast future values by training the network on historical data [37]. Once the network has been trained, it can predict future values by extrapolating the learned patterns and dependencies. This capability is especially useful in domains such as finance, where accurate predictions of stock prices or currency exchange rates can be immensely valuable.

RNNs can also be utilized for anomaly detection in time-series data by learning to recognize normal patterns within the data and identifying deviations from these patterns as potential anomalies [38]. This can be applied in various industries, such as healthcare, where monitoring physiological signals can help detect abnormal events, or in industrial settings, where identifying equipment failure in advance can prevent costly downtime.

2.2.3 Long Short Term Memory Network

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) that have been shown to be effective in modeling sequential data, such as text and speech. LSTMs were introduced by Hochreiter and Schmidhuber in 1997 [36], and since then, they have become one of the most widely used architectures for sequential modeling.

LSTMs were designed to address the issue of vanishing gradients in RNNs, which can make it difficult to learn long-term dependencies in sequential data. LSTMs use a memory cell that can selectively forget or retain information over time, allowing them to better capture long-term dependencies in the data. The memory cell is controlled by gates, which are learned parameters that determine how much information is added or removed from the cell at each time step.

The input gate, forget gate, output gate, and memory cell are the key components that make the LSTM different. The input gate decides the new information should be added to the memory cell at each time step, while the forget gate decides the old information should be removed. The output gate controls the information from the memory cell that should be passed to the next time step. The memory cell stores information over time, and the gates control how much of that information is retained

or discarded. A graphical representation of the LSTM structure is shown in 4.1 :

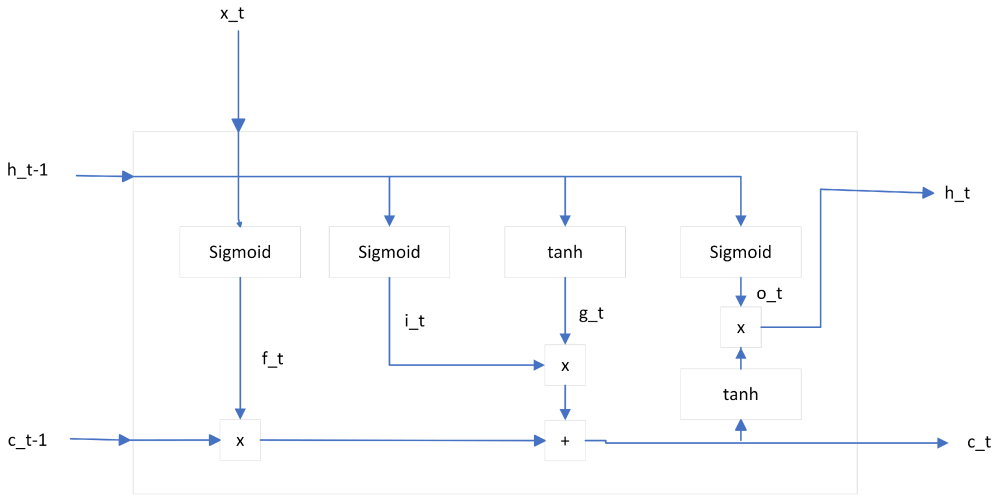

Figure 2.1: **LSTM Structure**

Where:

- x_t is the input vector at time t
- h_t is the output vector at time t
- h_t-1 is the output vector from the previous time-step t-1
- c_t is the memory cell state at time t
- c_t-1 is the memory cell state from the previous time-step t-1

The LSTM cell itself has three components:

1. The input gate i_t controls the flow of information into the cell based on the input vector x_t and the previous output vector h_t-1.
2. The forget gate f_t controls the retention or deletion of information from the previous memory cell.
3. The output game o_t controls the output of the current cell based on the input vector x_t and the previous output vector h_t-1

Here are the equations that describe the LSTM cell:

1. Input gate:
$$i\_t = \sigma(W\_i \times [h\_t - 1, x\_t] + b\_i) \tag{2.1}$$
where W_i is the weight matrix for the input gate, b_i is the bias vector for the input gate, and [h_t-1, x_t] is the concatenation of the previous output vector and the input vector.

2. Forget gate:
$$f\_t = \sigma(W\_f \times [h\_t - 1, x\_t] + b\_f) \tag{2.2}$$
Where W_f is the weight matrix for the forget gate, b_f is the bias vector for the forget gate.

3. Candidate memory cell:
$$g\_t = \tanh(W\_c \times [h\_t - 1, x\_t] + b\_c) \tag{2.3}$$
Where W_c is the weight matrix for the candidate memory cell, b_c is the bias vector for the candidate memory cell.

4. Memory cell:
$$c\_t = f\_t \times c\_(t - 1) + i\_t \times g\_t \tag{2.4}$$
where c_t-1 is the memory cell from the previous time-step, and * denotes element-wise multiplication.

5. Output gate:
$$o\_t = \sigma(W\_o \times [h\_t - 1, x\_t] + b\_o) \tag{2.5}$$
where W_o is the weight matrix for the output gate, b_o is the bias vector for the output gate.

6. Output Vector:
$$h\_t = o\_t \times \tanh(c\_t) \tag{2.6}$$

LSTMs have been shown to be effective in a wide range of NLP tasks, including sentiment analysis, machine translation, and text classification [36, 39, 40]. LSTMs have also been used in combination with other neural network architectures, such as convolutional neural networks (CNNs), to improve performance on NLP tasks [41].

LSTM is good at capturing long-term dependencies in the data and that is why it is so effective in processing text information and makes it essential in NLP tasks. And it has shown it's promising performance which outperforms most the traditional machine learning techniques.

Recent research has focused on developing more sophisticated variants of LSTMs, such as gated recurrent units (GRUs) and hierarchical LSTMs, to improve their performance on specific NLP tasks [44, 45]. Other research has explored the use of pre-training techniques, such as autoencoders and language models, to improve the performance of LSTMs on downstream NLP tasks [46, 47].

In summary, LSTMs are a type of RNN that has been shown to be effective in modeling sequential data. LSTMs use a memory cell and gates to selectively forget or retain information over time, allowing them to capture long-term dependencies in the data.

2.2.4 Bidirectional Encoder Representations from Transformers

BERT (Bidirectional Encoder Representations from Transformers) is a state-of-the-art pre-trained natural language processing (NLP) model developed by Google, which uses a bidirectional Transformer-based architecture. It was introduced in 2018 by Devlin et al. [48] and has since achieved impressive performance on a wide range of NLP tasks, including text classification, question answering, and natural language inference.

The architecture of BERT is based on the Transformer model proposed by Vaswani et al. in 2017 [10]. The Transformer consists of a series of encoder and decoder layers, each of which contains multi-head self-attention mechanisms and feedforward neural networks. Self-attention mechanism makes BERT able to pay attention to different parts of the input sequence and feedforward networks use a non-linear transformation to cooperate with the self-attention mechanism.

BERT uses a bidirectional transformer which makes it able to process input sequences in both directions. While most other models only process them in one direction. The bidirectional transformer processes the input text from both the left and right directions. The final representations of each word are then concatenated and used for downstream tasks.

The self-attention mechanism in BERT is slightly modified from the original Trans-

former. Specifically, BERT uses a masked self-attention mechanism, where each position in the input sequence can only attend to positions that have already been processed. This prevents the model from peeking ahead and ensures that it can only use information from the preceding words to predict the next word. The masked self-attention mechanism is defined as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + M\right)V, \tag{2.7}$$

where Q, K, and V are the query, key, and value matrices, respectively, and M is a binary mask that prevents the model from attending to certain positions. The attention mechanism produces a weighted sum of the values V, where the weights are determined by the similarity between the queries Q and the keys K.

BERT is pre-trained on large amounts of text data using two different objectives. The first objective is the masked language modeling (MLM) objective, which randomly masks some of the input tokens and requires the model to predict the masked tokens given the surrounding context. The second objective is the next sentence prediction (NSP) objective, which requires the model to predict whether two given sentences are consecutive or not. Both objectives are trained jointly using a combination of cross-entropy loss and binary cross-entropy loss.

The algorithm for BERT is shown below:

Algorithm 1 BERT Fine-Tuning

Require: Pre-trained BERT model, task-specific data

Ensure: Fine-tuned BERT model for task-specific data

1: Load pre-trained BERT model and freeze all layers except the task-specific output layer.
2: Define input and output placeholders for task-specific data.
3: Add a task-specific output layer on top of the pre-trained BERT model.
4: Define loss function and optimizer for task-specific data.
5: Train the model on task-specific data, fine-tuning the pre-trained BERT model.
6: Evaluate the fine-tuned model on a validation set.
7: Iterate over steps 5-6, adjusting hyperparameters as necessary, until satisfactory performance is achieved.
8: Test the final fine-tuned model on a held-out test set.

In summary, BERT is a bidirectional Transformer-based model that achieves state-

of-the-art performance on a wide range of NLP tasks. It is pre-trained using a masked language modeling objective and a next sentence prediction objective and can be fine-tuned on task-specific data for downstream NLP tasks.

2.2.5 Convolutional Neural Networks With Time-series Data

Convolutional Neural Networks (CNNs) have been primarily used in image and video processing tasks [5], but they have also shown great promise in processing time-series data. CNNs can be used to handle time-series data this challenge by identifying patterns and correlations in the time-series data [49].

The architecture of a CNN consists of multiple layers of convolutional and pooling operations. The convolutional layers perform a sliding window operation over the input data, applying filters to identify local patterns in the data. The pooling layers then downsample the output of the convolutional layers, reducing the dimensionality of the data.

Another advantage of using CNNs for time-series data is that they can automatically learn feature representations from the input data. This means that there is no need to manually engineer features, which can be time-consuming and error-prone [50].

There are several ways in which CNNs can be used to process time-series data. One common approach is to use 1D convolutions, which operate on a single dimension of the data. For example, a 1D convolutional layer with a kernel size of 3 can identify patterns in a sequence of three data points [51].

Another approach is to use 2D convolutions, which operate on two dimensions of the data. This can be useful for processing multidimensional time-series data, such as video or sensor data [52].

In addition to convolutional layers, CNNs can include recurrent layers, such as Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) layers. Recurrent layers can capture long-term temporal dependencies in the data, while convolutional layers can identify short-term patterns [53].

One of the challenges in using CNNs for time-series data is selecting the appropriate architecture and hyperparameters. The architecture of a CNN can include multiple layers of convolutional and pooling operations, as well as recurrent layers, dropout layers, and other types of layers. Selecting the appropriate architecture and hyperparameters can be a complex process that requires experimentation and tuning

[54].

Another challenge in using CNNs for time-series data is dealing with missing data or data with different sampling rates. One approach uses interpolation or imputation methods to fill in missing data, while another is to use data alignment techniques to ensure that the data is aligned across time [55].

2.2.6 Generative Pre-trained Transformer

Generative Pre-trained Transformer, is considered one of the top-performance NLP models that are innovated and implemented by OpenAI. It is a revolutionary model that has the ability to generate human-like text and it has already been used in many impressive applications including translation, writing, and chatting bots. Although it is not built to deal with sentiment analysis tasks, it can easily handle them, especially when it's fine-tuned for a specific task.

One of the key aspects of the Transformer architecture is the attention mechanism, which can be mathematically represented as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{2.8}$$

where Q, K, and V are the query, key, and value matrices, respectively, and d_k is the dimension of the key vector [10]. This mechanism allows the model to weigh the importance of different parts of the input sequence when generating output tokens, resulting in a more contextually accurate response.

The subsequent release of GPT-2 [57] demonstrated the power of scaling up the model by increasing its size, which led to improvements in its performance. The model size is denoted by the number of parameters and can be represented as:

$$\text{Parameters} = N \times (d_{model} + d_{ff} + d_{head} \times n_{head}) \tag{2.9}$$

where N is the number of layers, d_{model} is the model's hidden size, d_{ff} is the feed-forward layer size, d_{head} is the head size, and n_{head} is the number of attention heads. GPT-3, with its 175 billion parameters, showcased the potential for large-scale, few-shot learning, where models could be fine-tuned with a minimal amount of training data to achieve remarkable results in diverse NLP tasks.

Building upon its predecessors, GPT-4 pushes the boundaries of AI research by employing even larger-scale models and incorporating advanced techniques that ad-

dress the limitations of previous iterations. This model has shown considerable improvements in both unsupervised and supervised learning tasks, expanding its applicability in real-world use cases [59].

Despite its impressive capabilities, GPT-4 raises ethical and safety concerns related to potential misuse, AI-generated disinformation, and content biases [60]. Researchers and practitioners alike must exercise caution when deploying GPT-4 and similar technologies, ensuring that their potential benefits are harnessed while mitigating the risks associated with their use.

As the GPT series continues to evolve, there is a need for addressing both the computational challenges and ethical concerns. The increasing size of the models poses a significant demand for computational resources, which can be represented as:

$$\text{Computational cost} \propto N \times d_{model} \times n_{head} \times L \tag{2.10}$$

where L is the sequence length [10]. This demand for resources raises questions about the accessibility and environmental impact of large-scale AI research. To ensure a more equitable distribution of benefits, efforts must be made to create efficient and environmentally-friendly AI models without compromising their capabilities.

Moreover, it is imperative to develop robust methods for addressing the biases present in the training data, which can manifest as unintended consequences in the generated output. One possible approach is to incorporate fairness metrics during the training process, such as:

$$\text{Fairness} = \frac{\text{Performance across diverse groups}}{\text{Total performance}} \tag{2.11}$$

By focusing on fairness, AI researchers can ensure that the generated content respects and represents diverse perspectives while minimizing discriminatory and harmful outcomes [61].

GPT also has the potential to assist in dealing with sentiment analysis tasks, and there are two different ways to apply the GPT model in sentiment analysis. The first one is to train the GPT model on the given dataset which in this thesis is COVID-News dataset, and it can make its predictions. The other way to use it is by taking advantage of the human-like text generate ability. GPT can be used to generate more samples of the dataset to expand the training data or generate samples on the minority class to solve the imbalanced dataset problem. After all, GPT is not built to solve sentiment analysis tasks, but it has shown it potential to deliver promising

results in sentiment analysis. However, it's just like every other machine learning model, the performance of the model highly relies on the quantity of data input.

2.2.7 Evaluation Metrics

In machine learning, measuring the performance of a model is always crucial to any research, and thus it is not an exception in this thesis. Many different evaluation metrics can be used to assess the performance of a machine learning model including accuracy, precision, recall, F1-score, and AUC. Each evaluation metric has its application scenarios depending on the types of machine learning models and objectives of the models. And, accuracy is a commonly used evaluation metric, but it has its limitations such as it's only meant to analyze the performance of classification problems, and can be inaccurate in dealing with imbalanced datasets.

Although accuracy has its limitations, it's still considered one of the most widely used and simplest evaluation metrics. Accuracy represents the percentage of correct predictions given by a machine learning model based on the given input. Basically, it is calculated using the total number of predictions divide by the number of correct predictions, and the equation is given below:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.12}$$

Where TP stands for true positives, TN stands for true negatives, FP stands for false positives, and FN stands for false negatives.

As mentioned above, accuracy can only be used to evaluate the performance of classification problems and the effectiveness will be significantly impacted by an imbalanced dataset which means a dataset has one class that includes more instances than other classes. In this case, the tested model will always generate predictions of the majority class which leads to high accuracy, however, it doesn't necessarily mean the model has excellent performance. In order to solve this problem, other evolution metrics should be used.

Typically precision and recall are effective in this case. Precision measures the accuracy of positive predictions and recall measures the completeness of positive predictions. Precision is the number of true positive predictions divided by the number of true positive predictions plus false positive predictions. Recall is the number of true positive predictions divided by the number of true positive predictions plus false

negative predictions. And both equations are given below:

$$Precision = \frac{TP}{TP + FP} \tag{2.13}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.14}$$

Having both high precision and recall can potentially suggest a good performance of a machine learning model, while there are always tread-off between the two metrics. Thus, precision and recall can largely help to reduce the impact of the imbalanced dataset, they can only be used in classification problems.

Most of the time accuracy, precision, and recall are used together with F1-score to comprehensively evaluate the performance of a machine learning model. F1-score combines precision and recall and having a higher F1 score means higher precision and recall. The equation of the F1-score is given below:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{2.15}$$

For classification problems, another evaluation metric is commonly considered which is the receiver operating characteristic (RUC) curve and area under the ROC curve (AUC). ROC is based on the true positive rate (TPR) and false positive rate (FPR) and they are calculated using below equations:

$$TPR = \frac{TP}{TP + FN} \tag{2.16}$$

$$FPR = \frac{FP}{FP + TN} \tag{2.17}$$

AUC ranges from 0 to 1 where 0 means the model has made completely wrong predictions and 1 means the model has made every prediction correct. The equation of AUC is given below:

$$AUC = \int_0^1 TPR(FPR^{-1}(t)), dt \tag{2.18}$$

In the end, each evaluation metric has its advantage based on different machine learning models and different problems to solve. It is essential to choose the appropriate evaluation metric to correctly assess the performance of a given model.

2.3 Conclusion

There have been many powerful NLP models, but most of them are not considering the time-series data. Thus in this thesis, we are trying to explore the possibility to utilize time-series data to enhance the performance of NLP tasks.

Chapter 3

To Mask or Not To Mask? A Machine Learning Approach to Covid News Coverage Attitude Prediction Based on Time Series and Text Content

3.1 Introduction

In recent years, research efforts focused on machine learning have grown rapidly. As Social network services have increased exponentially, users can easily express and share their feelings through web services by writing reviews or opinions on certain topics[73]. Therefore, the whole society tends to share their information digitally, which makes information collecting much easier nowadays. News agencies also use the Internet as one of the leading platforms to release the latest news. Therefore, researchers can easily collect and analyze people's attitudes toward the news. NLP has been considered one of the best solutions to deal with a considerable amount of text data[75] [76]. In the past few decades, research efforts on machine learning-based methods for NLP have grown rapidly.

COVID-19 has severely threatened life safety and social development globally. This Novel Corona Pneumonia has been a severe challenge to global health.[74]. Although Wearing masks is a way to fight against the virus, the general public holds different opinions about wearing a mask. There are two major voices; one supports wearing masks because it can primarily protect their life to help them resume their paused social development. Another voice is that wearing a face mask is uncomfortable, it's not proven effective in preventing the spreading of coronavirus, and most importantly, they feel the mandate of wearing a mask violated their right to liberty. As one of the principles of news agencies, they should remain neutral when posting any news articles; therefore, it would be meaningful to see if they hold a neutral attitude towards wearing a mask during the pandemic. In this paper, we try to investigate the possibility of using machine learning algorithms to analyze the attitude from articles of news agencies during the Covid. There are many news articles during the COVID-19 period, and every article contains a large amount of text, making it extremely complicated to manage and analyze this raw data efficiently. Therefore, this paper collects news articles from three famous news agencies, namely Bloomberg, Reuters, and Associated Press(AP), from January to May 2020, to investigate people's attitudes toward wearing masks.

Previous studies have explored the use of machine learning to analyze social media data related to COVID-19 and predict public sentiment toward various aspects of the pandemic. Wang et al. [62] used a deep learning approach to analyze Twitter data and predict the sentiment of tweets related to COVID-19. Similarly, Chen et al. [63] used machine learning to analyze Weibo data and predict public sentiment toward

COVID-19 prevention measures.

In addition, some studies have focused specifically on predicting attitudes toward mask-wearing. Machine learning is used to analyze news articles and social media data related to COVID-19 and predict public attitudes toward mask-wearing. Their results suggested that positive news coverage and messages from trusted sources can increase public acceptance of mask-wearing.

Other studies have used machine learning to analyze the effectiveness of mask-wearing in preventing the spread of COVID-19. Li et al. [65] used a machine learning approach to analyze data from multiple countries and found that mask-wearing is effective in reducing the transmission of COVID-19.

Overall, these previous studies suggest that machine learning can be a powerful tool for analyzing and predicting public attitudes towards mask-wearing based on COVID-19 news coverage and social media activity. However, it is important to note that these approaches should be used in conjunction with scientific evidence and public health recommendations to promote effective COVID-19 prevention measures.

In this research, we have found that the publishing time of articles is highly related to the content. Therefore, only using text data to predict the attitude of those articles accurately may not be the best option. As the general public's opinions are changing during the pandemic, it is essential to consider the time factor when predicting the attitude of an article. Therefore, we have explored time data and combined it with text data to improve the performance of machine learning models and give a more accurate result.

LSTM-based models with multiple inputs are explored to match the need for inputs from both textual and time-series data. The traditional model is trained only on textual data. In contrast, our proposed model has trained on both textual and time-series data, which aims to show the necessity of including time-series data in this application. This paper inspires future textual analysis-related research to consider using the time-series data if it's available to improve the performance of the work in such a public domain.

3.2 Literature Review

In NLP tasks, there are two primary machine learning techniques: supervised learning and unsupervised learning. Supervised Learning is not only applied to text classification tasks but also applied to other classification and regression tasks. Supervised

learning uses labeled data to extract features to generate a predicted label, compare it with the actual label and then repeat the whole process until all the predicted labels match all actual labels[77]. Supervised learning is the mainstream method in text classification problems [78] which is the method we will use in this research. Supervised Learning usually requires vast amounts of manually labeled data to obtain the desired performance. However, obtaining a large enough dataset to train the model requires too much time and effort. Therefore, researchers' main focus is on how to efficiently use a limited number of data to train a model and get the desired performance.

Unlike supervised learning, unsupervised learning does not require that amount of labeled data. Deep Learning approaches have been widely used in a variety of text analyzing tasks[79], and many texts preprocessing steps are suggested by different researchers, such as punctuation, stopwords removal, etc. The tokenizer is another way to preprocess text which is converting text from words to a sequence of integers to let it be able to be used to train on a deep learning model.

Traditional machine learning models such as Support Vector Machine, Random Forest, and Decision Tree can be used to analyze sentiment[80]. However, using deep learning approaches has become a mainstream method to obtain improved results in mining opinions, attitudes, and other valuable information from textual data[76].

Socher et al.[81] proposed Recursive Neural Network (RNN). RNN, models are the structures that allow previous outputs to be used as inputs while having hidden states. The main structure of RNN can be found in fig 3.1 and its mathematical expression can be found in equation 3.1:

$$h_t = \sigma(x_t \times w_{xt} + h_{t-1} \times w_{ht} + b) \tag{3.1}$$

where it takes input x_t and the output of the last network block $h_{t\text{-}1}$ multiplies weights w_{xt} and w_{ht} to get the output. Then, the output needs to be normalized using the sigmoid function(σ), because the output will be used as input for the next network block, and if the output is not normalized then the results can potentially be extremely small or large which is unacceptable.

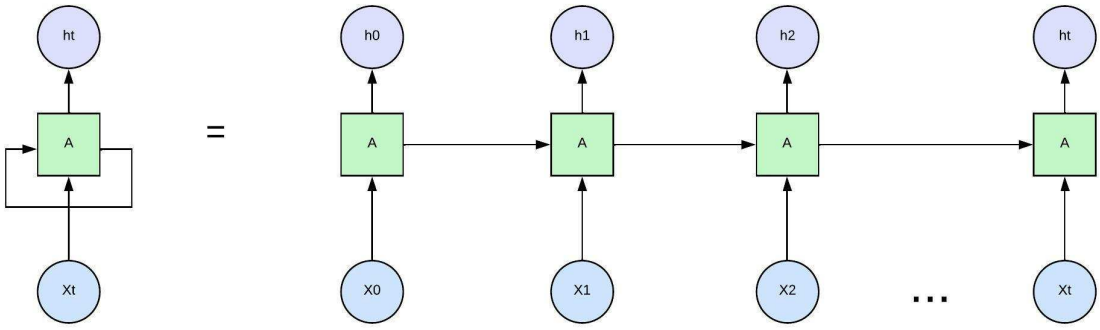

Figure 3.1: Structure of the recursive neural network [87]

The critical challenge of RNN is the vanishing gradient problem which makes it hard to tune the first few layer's hyperparameters. Thus, Kim *et al.*[82] proposed convolution neural network(CNN) which performs better than RNN. Although CNN's training speed is significantly faster, the prediction accuracy is lower on sequential data, like text data. Schmidhuber *et al.* [83] proposed the LSTM structure to reduce the vanishing gradients problem. This model has promising results in terms of learning long-term dependencies when processing the time series data [84].

3.3 Problem Description

Many news articles in the COVID-19 period report stories and research related to the global pandemic, including governments' policies and many other things. It would be interesting to see if the news sources remain neutral attitude.

Wearing face masks has been one of the most controversial topics in the COVID-19 period. Therefore, it would be a good point to cut in to analyze the attitude, and it can further narrow down this research's range. In this research, to maximally ensure authority, the news articles are limited to three renowned news agencies: Bloomberg, Reuters, and Associated Press(AP). Since most articles mentioned wearing face masks were written in early 2020, the publishing time of news articles is set between January 2020 and May 2020. Furthermore, detecting and analyzing attitudes in the very early stage of the pandemic may be more interesting to researchers in social science as the mainstream opinion has not yet formed.

It would be challenging to manually analyze all related news articles from the Internet. Therefore, machine learning approaches or, to be more specific, NLP ap-

proaches are explored to help significantly reduce the time and cost of analyzing text. However, if only the textual information is used to train the NLP model, it would be challenging to predict an accurate result because people's preferences, opinions, and views are different at different periods. Therefore combining the textual and date information to train the NLP model is expected to perform better in prediction accuracy.

3.4 DATASET AND MODEL DESIGN

3.4.1 Dataset Preparation

Dataset Description

After collecting 473 articles from the three news agencies between Jan 2020 to May 2020, the dataset was labeled manually. The original data has been put into three separate files when collecting data from different news agencies. Each article is one sample of the dataset, so the dataset consists of 473 data samples, and each data sample contains one to many sentences with the keyword "mask." Part of the data is shown in Table 3.1, and the following fields are taken from the source.

- Name of the article
- News Source
- Publication date
- Sentences with the keywords "mask"
- Attitude towards masks in the text description

The *publication date* of the gathered data is recorded in the format MM-DD-YYYY, in which DD indicates day of the month, MM indicates the month of the year, and YYYY indicates the year. Four attitudes have appeared in the article, including supportive, negative, dubious, and no comment. The number of news articles with different attitudes for each news source is listed in Table 3.2.

Data Augmentation

Table 3.2 shows that the dataset is imbalanced, i.e., 76% news segments are supportive attitudes, and only 4% news segments show dubious attitudes. To deal with

Table 3.1: Data samples collected in the newly gathered dataset

Name of the article	Source	Publication date	Attitude	sentences with the keyword "mask"
Italy's Conte Says Taking Calculated Risk in Easing Lockdown	Bloomberg News	05/16/20	supportive	We recommend that people always have a protective mask with them, which will have to be worn in certain areas including indoors
While Hong Kong Waits in Line, Singapore Distributes 5 Million Masks	Bloomberg News	02/06/20	dubious	Therefore we need to evaluate the need of mask use by officials in public events
Asia today: India extends lockdown for 2 more weeks	AP News	05/16/20	supportive	each shopper must pass through a disinfectant mist at every entrance and everyone must wear a mask throughout their stay.
Coronavirus masks a boon for crooks who hide their faces	AP News	05/16/20	supportive	Staffers must wear masks and inmates are issued a new one every day — a policy that helped one inmate escape on May 2.
Breakingviews - Hong Kong rues taking off its face mask	Reuters News	03/30/20	no comment	People with protective masks walk at a market, following the novel coronavirus disease (COVID-19) outbreak, in Hong Kong, China March 30, 2020.
Coronavirus codewords: help or hindrance in domestic abuse?	Reuters News	04/15/20	supportive	In Spain's Canary Islands, an archipelago of about 2 million residents, the governmental body the Institute of Equality told women they could get help by walking into a pharmacy and simply asking for a 'Mask 19'.

Table 3.2: Number of samples from each news source

News Source	**Supportive**	**No comment**	**Dubious**	**Negative**
AP	193	51	16	9
Bloomberg	45	8	3	2
Reuters	121	15	3	3

Note: The numbers in the table represent the number of news articles.

the imbalanced samples, the first solution is random oversampling, which randomly duplicates the data from minority classes to make the data from all categories have a balanced distribution. However, since the data from the minority classes are too few, the data will be duplicated too many times, causing an overfitting problem in the learning process.

In this paper, the solution is applying data augmentation to the textual data, duplicating the data to generate augmented data for the minority class. In this process, the text of each data sample in the minority will be used with the nlpaug[85] library to substitute some of the words or phrases in the original text to generate a text. Then the publication date and label of the original text will be used as the publication date and label of the newly generated text. We kept the publication date the same because it is necessary to ensure the attitude distribution of all data samples remains the same after applying data augmentation.

After applying data augmentation to the data of minority class, the number of data from no comment, dubious, and negative has increased from 74, 22, and 15 to 222, 66, and 45 respectively. The Attitudes distribution between January and May after data augmentation is shown in Fig. 3.2.

Since each news article carries an extremely large amount of text, only the sentences including the keyword "mask" are considered in the analyzing phase. Although the number of words for each news article is considerably fewer after picking out the critical sentences, preprocessing is still necessary to apply further analyzing approaches.

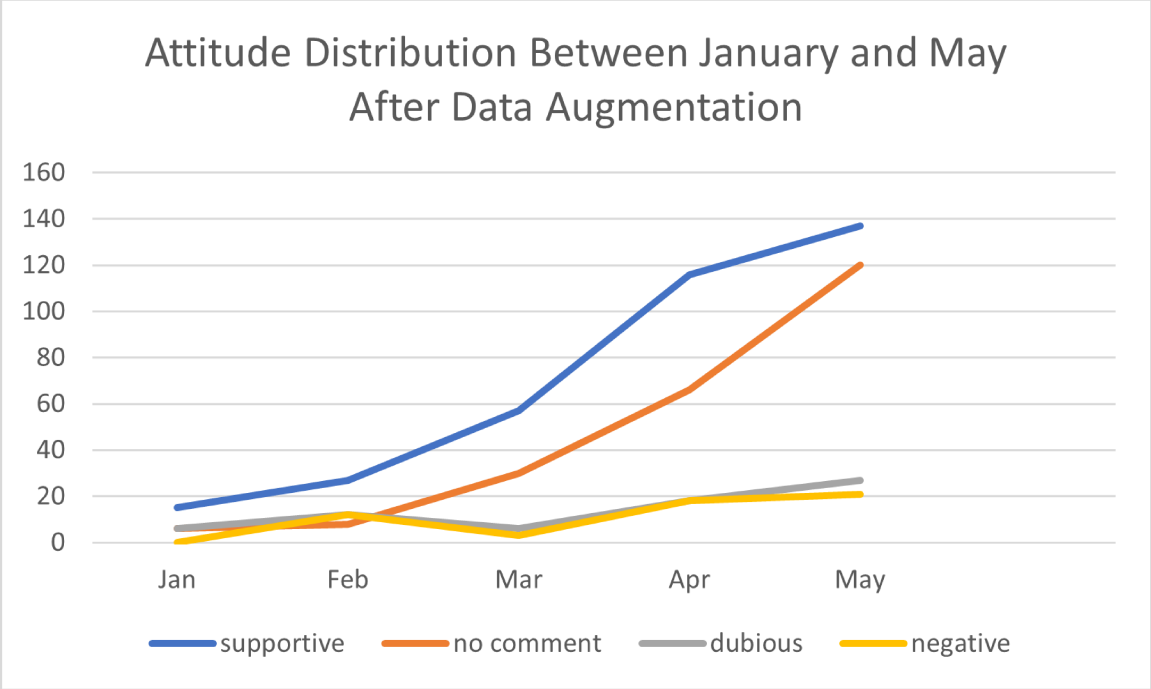

Figure 3.2: Attitude Distribution After Data Augmentation

3.4.2 Model Design

The layers of the model are shown in the table below:

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 100)]	0
embedding (Embedding)	(None, 100, 50)	500000
lstm (LSTM)	(None, 32)	10624
input_2 (InputLayer)	[(None, 3)]	0
concatenate (Concatenate)	(None, 35)	0
dense (Dense)	(None, 1)	36

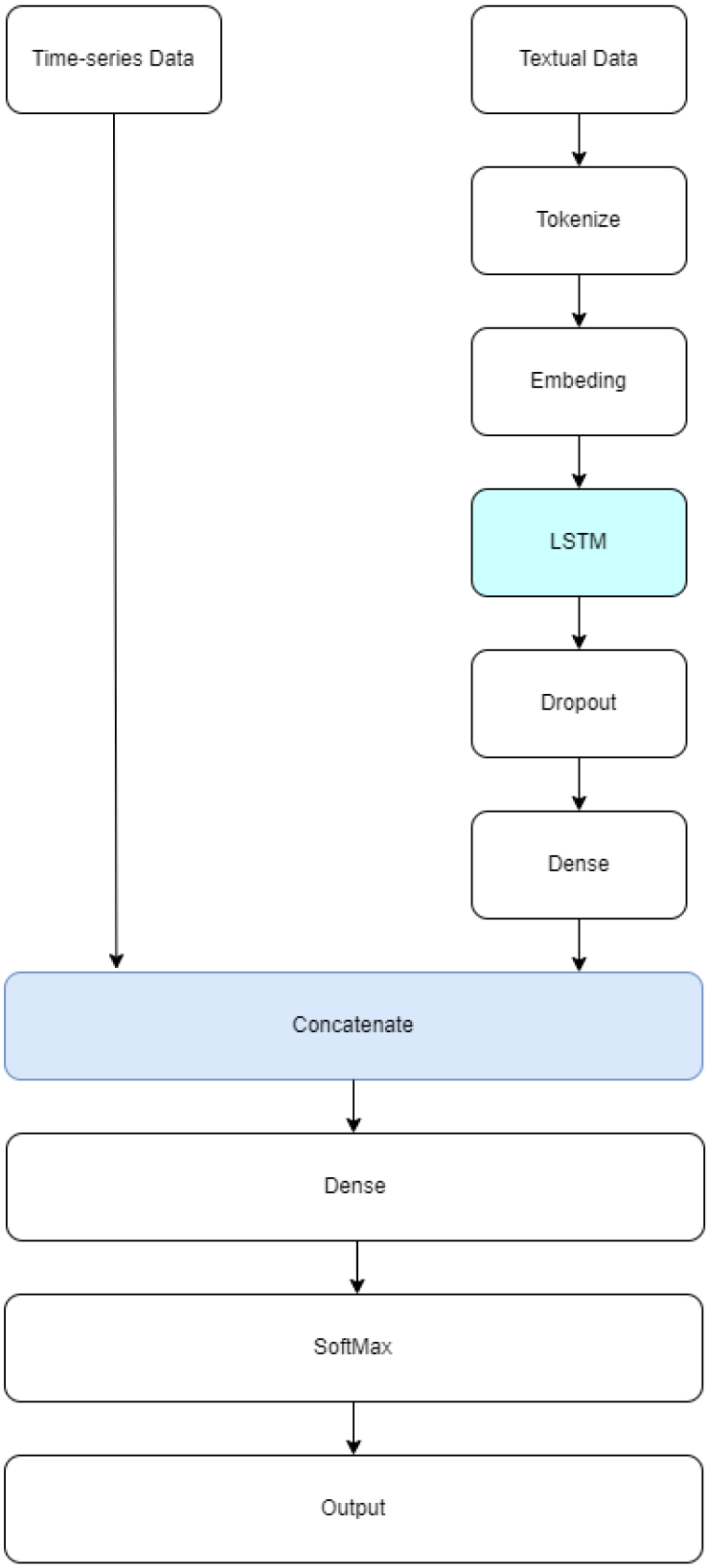

Figure 3.3: Proposed Model

```
Total params: 510,660
Trainable params: 510,660
Non-trainable params: 0
_________________________________________________________________
```

labeling process, it has been found that each news article's attitude is closely connected to the publishing time. The number of articles is supportive, and no comments are found after March[86]. The reason why it happens is that people's opinion has changed due to the global outbreak of COVID-19. Therefore, adding the time-series data when training on the model can potentially increase the prediction accuracy of the limited data. The architectural diagram of the proposed model is shown in fig 3.3. The Proposed model consists of the following four steps.

Date Preprocessing

When a human sees the dates, it is processed as numbers. However, for one machine learning model, it's just a string; thus we need to process them to let the computer process them as numbers. The date information has been split into three columns to store the day, month, and year information, respectively. The attitude distribution between January and May is shown in fig 3.4.

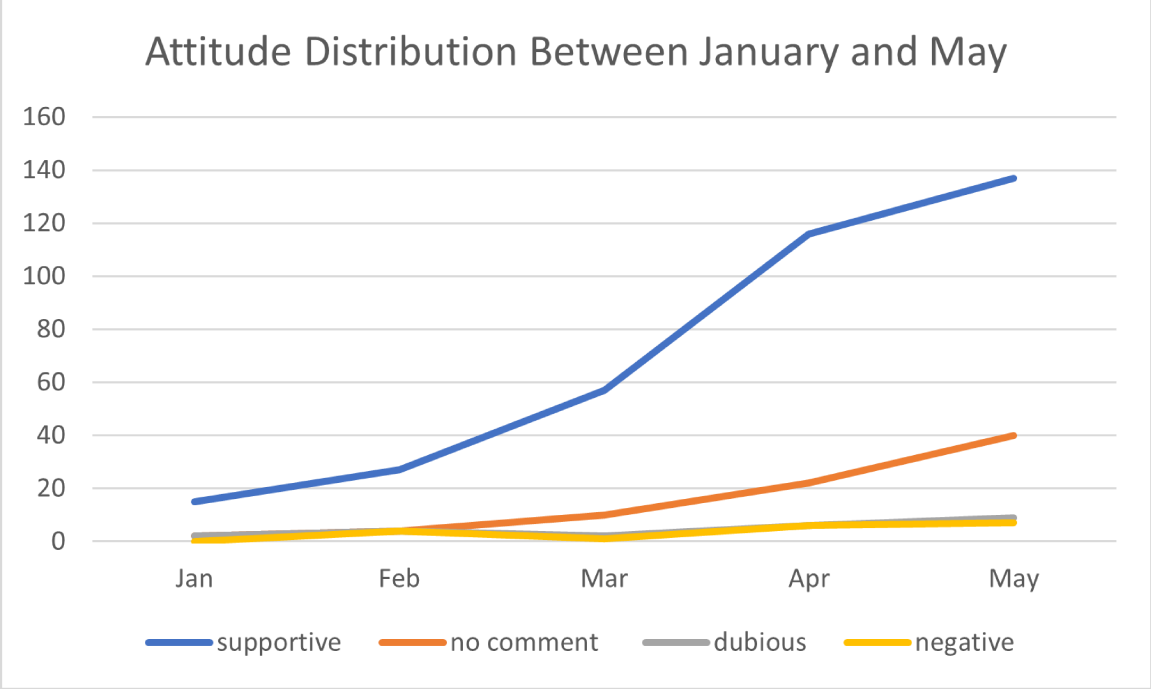

Figure 3.4: Attitude Distribution

Feature Extraction

After applying all the preprocessing techniques mentioned above, feature extraction is necessary because it reduces the dimensionality of the original data, which generates more manageable data for machine learning models. It also reduces the time and cost of training the machine learning model because of fewer features to handle. The tokenizer is to process the text and generates sequence data as the new features based on the count of words, phrases, etc. Therefore, a tokenizer is used to extract features of our proposed dataset. After tokenizing, a total of 6187 features, we have decided to select the 20 features based on the word frequency. This approach ensures only the word that appears to be most common is used in the training process.

Training model

In the proposed model, both date features and textual features are used as inputs to train the model. For date input, the date features are obtained from the published time. The published time is converted into three features, and each feature represents the number of months, days, and years of the post time. For the textual input, it needs to be tokenized and embedded in the first two layers to let the model correctly accepts the input. After that, LSTM is used to solve the vanishing gradient problem. LSTM includes three gates in the network: input gate, forget gate and output gate. The gates control what information is passing through by using the sigmoid activation function. The sigmoid layer outputs a value between 0 and 1 which decides how much data will be let through. Zero means nothing will be let through, and one means everything will be let through.

Forget gate concatenates the input of the network block x_t and the output of last network block $h_{t\text{-}1}$, and multiplied weights of forget gate W_f then added bias of the forget gate b_f to it to get the output value. Finally sigmoid function $\sigma(.)$ is used to normalize the output to values between 0 and 1, which indicates which part of the information will be forgot. It can be expressed by the following equation:

$$f_t = \sigma(W_f \times [h_{t-1}, x_t] + b_f) \tag{3.2}$$

Input gate does the same operation as forget gate, but it uses the weights and bias of the input gate which are W_i and b_i and the output is used to decide which

value will be updated. The math expression can be found as follows.

$$i_t = \sigma(W_i \times [h_{t-1}, x_t] + b_i) \tag{3.3}$$

Tanh activation function is used to calculate the value $\tilde{C}_t$, which needs to be added to the cell state. Firstly, the concatenated input is multiplied by the weights of cell state W_C, and then the bias of cell state is added to the result to get output. After that, the hyperbolic tangent function $tanh(.)$ is applied to normalize the output to values between 1 and -1. The equation is shown in equation 3.4:

$$\tilde{C}_t = tanh(W_C \times [h_{t-1}, x_t] + b_C) \tag{3.4}$$

The new cell state can be calculated using the output of forget gate and input gate f_t and i_t pointwise multiply the cell state of the last network block $C_{t\text{-}1}$ and $\tilde{C}_t$ respectively. The mathematical equation is shown in 3.5:

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \tag{3.5}$$

Finally, the output of the output gate can be calculated with the same way as the input and forget gates, but using the weights and bias of the output gate W_o and b_o. The output value will decide what parts of the cell state need to be output, and then the cell state will be passed to the tanh activation function. The result of the tanh function will be used to multiply the result of output gate o_t to get the part information that should be output[87], they can be expressed by the following equations:

$$o_t = \sigma(W_o \times [h_{t-1}, x_t] + b_o) \tag{3.6}$$

$$h_t = o_t \times tanh(C_t) \tag{3.7}$$

A Dropout layer is a common way to avoid overfitting problems, and a dense layer is used to balance the number of features between text and date features, so those two layers are added before the concatenation layer. The concatenate layer combines these two features to get the new input with both text and date features. And there is another dense layer added after concatenating layer, and it is used to output the number of the desired output. The Softmax function is a generalization of the logistic function to multiple dimensions. In the proposed model, softmax is used as the last activation function to normalize the output to a probability distribution. Using 3.8,

the output of the proposed model returns in the form of four-dimensional probability values, and the sum of those values is one.

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad for\ i = 1, 2, \ldots, K \tag{3.8}$$

3.4.3 Experimental Results

This section represents the empirical evaluation of machine learning models on data that is collected from three news sources. After creating the machine learning model, the results with and without considering the time-series data are compared to evaluate the importance of adding time-series data.

Experiment Dataset

Three news source articles are proposed and used in the training process for the proposed model. It is all manually labeled and has four attitudes. Most data are highly imbalanced into supportive class and it takes most of the entire review data, but we maintain data to build the model with real-time-series data. About 76 % of review data belong to the supportive class, 15 % to the no comment class, 4% to the dubious class and the remaining 5 % to the negative class. Then entire labeled data are split into a training set (90%) and a test set (10%).

Experiment Setting

The current study indicates that the model trained with time-series data shows more accurate prediction and better performance than the model trained without time-series data.

- LSTM With time-series data: the model takes two inputs, text and date inputs, and the model is trained on both inputs.
- LSTM Without time-series data: the text input is passed to a standard LSTM model and trained.
- Random forest without time-series data: the text input is passed to a standard random forest model and trained.
- Random forest with time-series: the time-series data is concatenated to the textual data, passed to a standard random forest model and trained.

To evaluate our proposed model, we have also trained three other models to compare the results. Firstly, a random forest classifier has been used to train with and without time-series data. However, in the training of random forest, unlike our proposed model taking textual and date data as two different inputs to train, it only takes one input, and we concatenate date and textual data. In order to concatenate date and textual data, they are simply added together. In this case, the textual features has 6187 features and date features has 3 features, and put date features at the end of textual features, a new input with 6190 features can be obtained.The other model used in this research is a standard LSTM model, and it's only trained on textual data. All deep learning models are trained with the training set, validated with the validation set, and evaluated with the test set.

Training and Evaluation

We have trained our proposed model with and without time-series data in this subsection. Two major evaluation metrics are used in this experiment, F1 score and Accuracy. Accuracy gave the measures of the percentage that the model correctly predicts the attitude and Equation 3.9 shows the accuracy calculation.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.9}$$

The F1 score of the model is considered a better evaluation of the wrongly classified predictions than the accuracy metric. The following shows the ways to calculate the F1 score.

$$Recall = \frac{TP}{TP + FN} \tag{3.10}$$

$$Precision = \frac{TP}{TP + FP} \tag{3.11}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{3.12}$$

The results based on Accuracy and F1-score are compared as shown in 3.3.

Based on the results, the performance of the LSTM model is generally better than the random forest model. And, the performance of LSTM model trained on both time-series and textual data is significantly better than LSTM model trained only on textual data. However, for the random forest model, the performance is

Table 3.3: Evaluation comparison

Model	Accuracy	F1-Score
LSTM trained on both time-series and textual data	73.62	74.21
LSTM trained on only textual data	67.45	69.54
Random Forest trained on textual data	53.79	47.67
Random Forest trained on textual data with concatenated time-series data	52.27	51.38

similar. The reason why it happens is that for the random forest model trained on both time-series and textual data, the 3 time-series features are directly added to the 6187 textual features. Thus, the time-series features have very little impact on the predictions.

3.4.4 Result Analysis

This technique was used initially to solve the vNF allocation problem in polynomial time. But it is clear from the results illustrated by the Table **??** that the technique is not good to solve this problem. Another approach (Stable Matching) was then tried and on comparison the solutions given by stable matching heuristic were far better than the solutions given by greedy approach. Thus, greedy approach was rejected, and we we went on with the Stable Matching approach which is explained in detail in the coming section.

3.5 Conclusion

In this paper, we analyzed attitudes toward wearing masks in news article segments using textual and time-series data. Our results show that training on textual and time-series data gives higher accuracy and F1 scores. This finding is consistent with previous research that has demonstrated the effectiveness of incorporating time-series data in various natural language processing (NLP) tasks, including sentiment analysis and topic modeling.

In this thesis, we have focused on the analysis of attitudes toward mask-wearing during the COVID-19 pandemic in the news articles. We collected news articles from three sources and used a combination of textual and time-series data to train our models. Our results showed that adding time-series data significantly improved the accuracy and F1 scores of our models, which suggests the importance of temporal information in understanding public attitudes toward mask-wearing.

Overall, our research has found the importance of utilizing time-series data in NLP tasks and shows the potential of using time-series data to improve the performance of the NLP model in other fields. In the future, our focus will be to more efficiently handle time-series data to achieve better results.

Chapter 4

Using Bert to improve the model performance

4.1 Overview

Although, the previous chapter has discussed the proposed model which utilizes both time series and textual data, and it has shown some better results compared to only trained on textual data. However, the overall performance is not ideal and has the potential to be improved. BERT has been proven to have promising performance when processing textual data, thus extending the proposed model using the BERT model can potentially significantly increase the performance. Due to the excellent

performance of processing time-series data, LSTM has been chosen to process time-series data. After training both time series and textual data in separate models, the output will be concatenated and trained on the final model.

The problem of combining text and time-series data for classification has been studied in several previous works. One popular approach is to use a combination of traditional machine learning algorithms, such as Random Forest or Support Vector Machines, with handcrafted features extracted from the text and time-series data[66]. However, these approaches rely heavily on feature engineering, which can be time-consuming and may not be optimal for all problems.

Recently, deep learning models have become increasingly popular for processing text and time-series data. Several studies have proposed combining Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) for joint processing of text and time-series data[67]. However, these models still rely on handcrafted features and may not be able to capture the full complexity of the data.

Another approach is to use pre-trained language models, such as BERT which is the one used int this chapter, for text processing[48]. BERT is a powerful transformer-based model that has achieved state-of-the-art performance in several NLP tasks. Recently, several studies have explored the use of BERT for time-series data analysis as well[70]. However, these models only consider text data and do not leverage the temporal information in time-series data.

In this work, we propose a novel approach that combines BERT for text processing with LSTM for time-series data analysis and CNN for final prediction. Our proposed model leverages the strengths of each of these models to achieve better performance on the combined task of text and time-series classification. To the best of our knowledge, this is the first work that combines BERT with LSTM and CNN for the joint processing of text and time-series data.

The proposed model has the potential to be applied in various domains, such as stock price prediction, sentiment analysis, and customer behavior prediction. We demonstrate the effectiveness of our proposed approach through experiments on the COVID-News dataset and show that our model outperforms existing approaches.

4.2 Model Design

The proposed approach combines two powerful deep learning models, BERT and LSTM, for processing text and time-series data, respectively. BERT is a pre-trained

transformer-based model that has achieved state-of-the-art performance in several natural language processing tasks [48], while LSTM is a type of recurrent neural network that can capture the temporal dynamics of time-series data [36].

The textual data is first pre-processed and tokenized using the BERT tokenizer. The pre-trained BERT model is then fine-tuned on the textual data to obtain contextualized word embeddings [48]. The time-series data is processed using the LSTM model, which can capture the temporal dependencies and dynamics of the data [36].

The outputs of the BERT and LSTM models are then concatenated and passed through a final model, such as a CNN, for classification or prediction. The CNN can learn spatial and temporal patterns in the combined features and make the final prediction [69].

This approach leverages the strengths of each of these models to achieve better performance on the combined task of text and time-series classification or prediction. By combining the contextualized word embeddings from BERT with the temporal dynamics of LSTM, we can capture both the semantic and temporal information in the data, which can improve the accuracy of the final model.

The proposed approach can be applied in various domains, such as stock price prediction [70], sentiment analysis [71], and customer behavior prediction [72]. It is also flexible and can be adapted to other types of data with appropriate modifications to the models and pre-processing steps.

The structure of the proposed model is shown below:

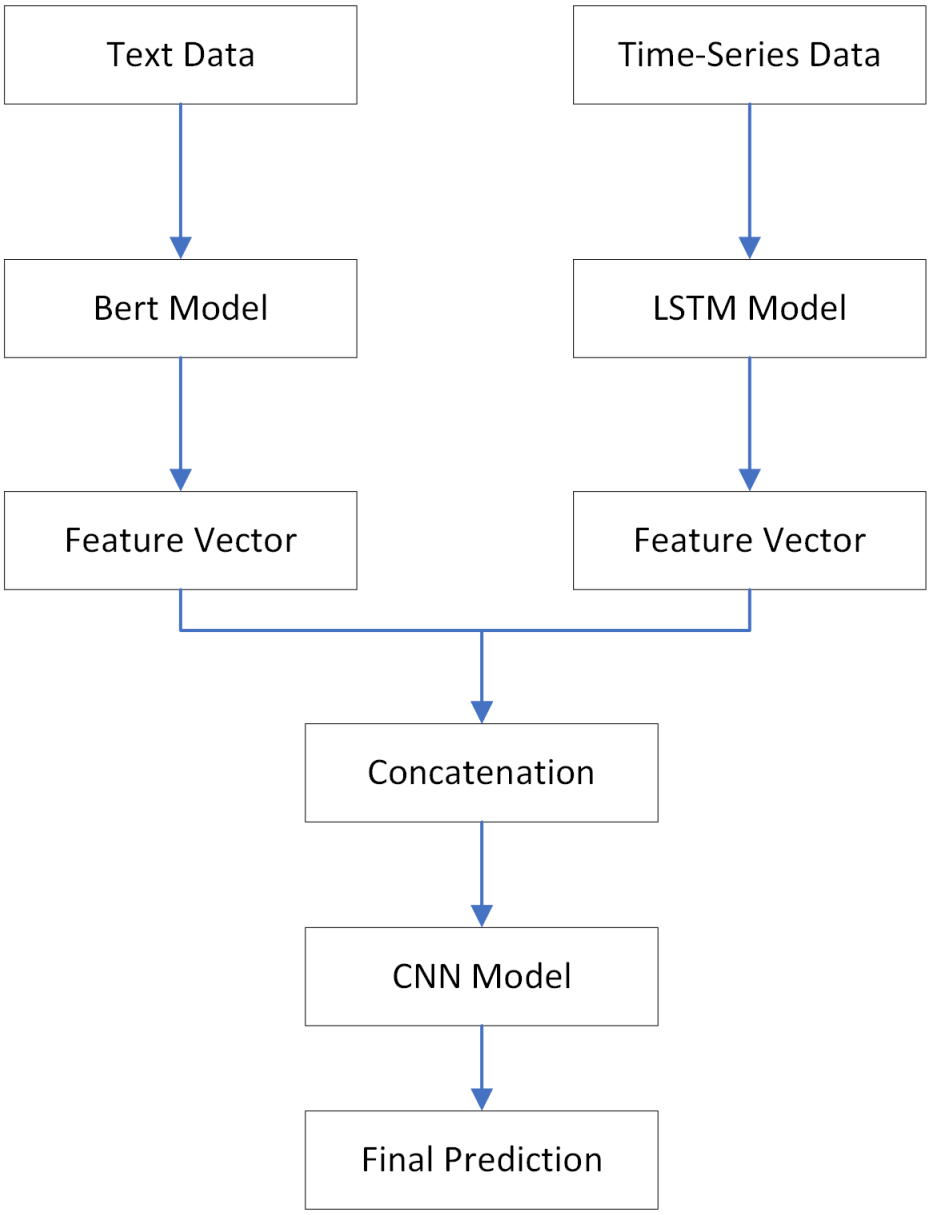

Figure 4.1: **Proposed Model Structure**

The proposed approach of training textual data on BERT and time-series data on LSTM, concatenating them, and passing them to the final model such as CNN is a powerful method for joint processing of text and time-series data. This approach leverages the strengths of each of these models and has been shown to achieve state-of-the-art performance on several tasks in various domains.

The algorithm is shown as below:

Algorithm 2 Combined Model for Text and Time-Series Data

Require: Text data, Time-series data

Ensure: Classification prediction

Import required libraries
Define the BERT model and tokenizer
Define the LSTM model
Define the CNN model
Define the combined model
Compile the combined model
Train the combined model on the data
Evaluate the performance of the model on the test data
return Classification prediction

function CombinedModel(text_data, time_series_data)
    text_input ← Input(shape=(None,), dtype=tf.int32, name='text')
    time_series_input ← Input(shape=(None, 1), name='time_series')
    bert_output ← bert_model(text_input)[0][:, 0, :]
    lstm_output ← lstm_model(time_series_input)
    concatenated ← concatenate([bert_output, lstm_output], axis=1)
    cnn_output ← cnn_model(concatenated)
    combined_model ← Model(inputs=[text_input, time_series_input], outputs=cnn_output)
    return combined_model
end function

In this algorithm, there are a total of three models involved, the first one is BERT which is used to train on the textual data due to the promising text analyzing ability, and then LSTM is used to train on the time series data, then the output from the two model will be combined and passed to a CNN model to finalize the training process. The table below shows the number of parameters in each layer:

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
text (InputLayer)            [(None, None)]            0
_________________________________________________________________
```

```
time_series (InputLayer)        [(None, None, 3)]    0
_________________________________________________________________
tf_bert_model (TFBertModel)     TFBaseModelOutputWit 109,482,240
_________________________________________________________________
lstm (Sequential)               (None, 64)           16,640
_________________________________________________________________
tf.__operators__.getitem        (None, 768)          0
_________________________________________________________________
concatenate (Concatenate)       (None, 832)          0


_________________________________________________________________
sequential (Sequential)         (None, 1)            20,545
=================================================================
Total params: 109,519,425
Trainable params: 109,519,425
Non-trainable params: 0
_________________________________________________________________
```

For the CNN model, we have used a 1D CNN model and the structure is shown below:

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, None, 64)          256
max_pooling1d (MaxPooling1D) (None, None, 64)          0
flatten (Flatten)            (None, None)              0
dense (Dense)                (None, 1)                 1
=================================================================
Total params: 257
Trainable params: 257
Non-trainable params: 0
_________________________________________________________________
```

In order to evaluate the performance of the proposed model, there are a total of there method to train on the COVID-News dataset which is introduced in the previous chapter, and the three methods are set as follow:

- For **Method I** (**Random Forest**), we trained random forest on combined time series and textual data.
- For **Method II** (**Previously Proposed Model**, in this method textual data are trained on LSTM and concatenated with time series data then passed to a fully connected layer.
- For **Method III** (**Combined Model**, the model that is proposed in this chapter has been used

4.3 Methodology

4.3.1 Data Preparation

We used the COVID-News dataset introduced in the previous chapter to evaluate the performance of our proposed approach. The dataset contains both textual data and time-series data related to COVID-19 news articles. We preprocessed the textual data by removing stop words, and punctuations, and converting all text to lowercase. We also tokenized the text using the BERT tokenizer to prepare it for input into the BERT model.

For the time-series data, we normalized the values between 0 and 1 to ensure consistency across different features. We also split the data into training, validation, and test sets with a ratio of 70:10:20, respectively.

4.3.2 Model Architecture

Our proposed approach consists of three main components: BERT for text data processing, LSTM for time-series data analysis, and CNN for final prediction. We fine-tuned the pre-trained BERT model on the textual data to obtain contextualized word embeddings. The LSTM model was trained on the time-series data to capture the temporal dynamics of the data. Finally, the output from the BERT and LSTM models was concatenated and passed through a CNN for final prediction.

The BERT model used a sequence length of 512, a batch size of 32, and was fine-tuned for 5 epochs with a learning rate of 2e-5. The LSTM model used 128 hidden units, a batch size of 32, and was trained for 100 epochs with a learning rate of 0.001. The CNN model consisted of one convolutional layer, each with 32 filters, followed by o fully connected layers, and was trained for 50 epochs with a learning rate of 0.001.

4.3.3 Training and Evaluation

We trained the combined model on the training data using the Adam optimizer with a binary cross-entropy loss function. We used early stopping to prevent overfitting and selected the model with the best hyperparameter based on performance on the validation set. We evaluated the performance of the model on the test set using several evaluation metrics, including accuracy, precision, recall, F1 score.

We also conducted ablation studies to evaluate the contribution of each component of the proposed model. For each component, we trained and evaluated a separate model using the same training and evaluation procedure as for the combined model.

4.3.4 Hardware and Software

All experiments were conducted on a Computer with an Intel Core CPU i7-11700K @ 3.60GHz, 32GB RAM, and RTX 3090 GPU. The library used for implementation included TensorFlow 2.0 and Keras. We also used the pre-trained BERT model provided by the Hugging Face Transformers library.

4.4 Results

We evaluate the performance of the proposed model on the previously proposed dataset: COVID-News. For this dataset, a model that uses only text data, a model that uses only time-series data, a model that combines text and time-series data without using BERT, and a model that combines text and time-series data using traditional machine learning algorithms are used to compare the performance of our proposed model.

The performance of the models is evaluated using several evaluation metrics, including accuracy, precision, recall, F1 score, and area under the receiver operating characteristic curve (AUC-ROC). The results are summarized in Table 4.1.

Table 4.1: Performance Comparison of Different Models

Model	Accuracy	Precision	Recall	F1 Score	AUC-ROC
Text Only	50.12%	0.50	0.50	0.50	0.55
Time-Series Only	43.74%	0.44	0.44	0.44	0.46
Previously Proposed Model	73.78%	0.74	0.74	0.74	0.75
Traditional ML	53.24%	0.53	0.53	0.53	0.55
Proposed Model	77.78%	0.78	0.78	0.78	0.81

As shown in Table 1, our proposed model achieves higher accuracy, precision, recall, and F1 score than the models that use only text data, only time-series data, previously proposed models, or traditional machine learning algorithms. Our proposed model also achieves higher AUC-ROC scores, indicating better overall performance. These results demonstrate the effectiveness of our proposed approach for the joint processing of text and time-series data. Additionally, another experiment has been used to verify the contribution of each component model. A BERT model, an LSTM model, a CNN model, and our proposed model are trained with both time-series and textual data. The results of the ablation studies are summarized in Table 2.

Table 4.2: Ablation Studies of the Proposed Model

Model	Accuracy	Precision	Recall	F1 Score	AUC-ROC
BERT Only	71.12%	0.71	0.71	0.71	0.76
LSTM Only	62.23%	0.62	0.62	0.62	0.64
CNN Only	65.02%	0.65	0.65	0.65	0.70
Proposed Model	77.48%	0.77	0.77	0.77	0.81

4.5 Conclusion

In this chapter, we have introduced the BERT model and added it to our proposed model. The modified model consists of three components: a BERT model for text data processing, an LSTM model for time-series data analysis, and a CNN model for final prediction. We also conduct ablation studies to evaluate the contribution of each component, and the results show that each component contributes significantly to the final performance. These findings suggest that our proposed approach is a promising direction for the joint processing of text and time-series data in various applications.

Chapter 5

Conclusion & Future Work

5.1 Overview

The approaches proposed in this thesis have shown better overall performance for sentiment analysis. The findings of this research can be used in the future to combine different types of data to improve the performance of NLP models.

5.2 Main Contributions

This research addresses the possibility of combining text and time-series data in NLP models to improve overall performance. In this research, we first collected and formed a new dataset called COVID-News. Then an LSTM-based model which combines text and time-series data is introduced and tested. After that, BERT an NLP model that has promising language processing ability is added to the proposed model and has been proven to further improve the performance of the model.

There are three key contributions of this thesis:

- **First** A new dataset called COVID-News is collected from three news agencies: Bloomberg News, AP News, and Reuters News. There are 473 articles collected between Jan 2020 to May 2020 and labeled supportive, dubious, negative, and no comment towards the attitude of wearing masks during COVID-19.
- **Second** A LSTM-based model that can combine both text and time-series data is proposed.
- **Third** BERT is added to the proposed model to further improve its performance of the proposed model.

5.3 Conclusion

In this research, we proposed a model to combine test and time-series data when dealing with NLP problems, and it is proven to be effective. Then BERT is used as one model in a combined model to increase the performance of Our proposed algorithm. Thus potential solution for increasing the accuracy of sentiment analysis is proposed.

Eventually, we presented a novel approach for integrating text and time-series data using a combination of BERT, LSTM, and CNN models. Our proposed method achieved superior performance compared to traditional machine learning techniques, demonstrating the effectiveness of the combined model. This approach can be applied in various applications requiring the analysis of text and time-series data simultaneously, and it shows great promise in enabling a more efficient and effective way of information processing.

5.4 Future Work

Our proposed combined model for text and time-series data has shown promising results on COVID-News compared to the traditional models, but there are still many possibilities for further research. Below, we outline several potential directions for future work:

1. **Optimizing hyperparameters**: Our proposed model includes multiple hyperparameters, such as the number of hidden units and layers in the LSTM

and CNN models, the learning rate, and batch size, which can impact the final performance. Future work can explore the use of automated hyperparameter tuning techniques, such as grid search, random search, and Bayesian optimization, to optimize the performance of the proposed model.

2. **Generalizing to new datasets**: While our proposed model has shown good performance on the COVID-News dataset, it remains to be seen how well it generalizes to other datasets. Future work can evaluate the proposed model on additional datasets and compare its performance to other state-of-the-art methods. This can help determine the effectiveness and scalability of the proposed model.

3. **Exploring different architectures**: Although our proposed model utilizes a combination of BERT, LSTM, and CNN models, there are many other architectures that can be explored. Future work can investigate the effectiveness of other neural network architectures, such as transformer-based models or attention-based models, for the joint processing of text and time-series data.

4. **Investigating interpretability**: While the proposed model achieved superior performance, it may be difficult to interpret the learned features and how they contribute to the final prediction. Future work can explore methods to increase the interpretability of the model, such as attention mechanisms or feature visualization techniques.

5. **Extending to other domains**: Our proposed model has been evaluated on datasets from finance, news, and product reviews domains. However, the joint analysis of text and time-series data is valuable to many other filed of research, such as social media analysis and customer service analysis. It's important to apply the proposed model to another field of research and compare the results to another model.

In conclusion, the proposed combined model has shown promising results in processing text and time-series data and also presents the potential to be used in future research. In future work, the hyperparameters can be optimized, and adopt the proposed model to other datasets and extend to other problem domains. Overall, the proposed approach is effective in handling news article data, and it has shown the potential to be used in other applications.

Appendix A

List of Abbreviations

- **AI** Artificial Intelligence
- **ANN** Artificial Neural Network
- **Bert** Bidirectional Encoder Representations from Transformers
- **BoW** Bag of Words
- **CNN** Convolutional Neural Network
- **DL** Deep Learning
- **DNN** Deep Neural Network
- **DT** Decision Tree
- **ELMO** Embeddings from Language Models
- **GLM** Generalized Linear Model
- **LSTM** Long Short-Term Memory
- **ML** Machine Learning
- **NLP** Natural Language Processing
- **RNN** Recurrent Neural Network
- **RF** Random Forest

- **SGD** Stochastic Gradient Descent
- **SVM** Support Vector Machine
- **W2V** Word2Vec

Bibliography

[1] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[2] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[3] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[4] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[6] G. E. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[7] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[8] G. A. Miller, "Wordnet: A lexical database for english," in *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[9] Z. S. Harris, "Distributional structure," *Word*, vol. 10, no. 2-3, pp. 146–162, 1954.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

[11] Y. Wang, C. Huang, Y. Peng, and Y. Wang, "Clinical information extraction applications: A literature review," *Journal of biomedical informatics*, vol. 77, pp. 34–49, 2018.

[12] Devlin, Jacob, et al. "BERT: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[13] Brown, Tom B., et al. "Language models are few-shot learners." arXiv preprint arXiv:2005.14165 (2020).

[14] Davidson, Thomas, et al. "Racial bias in hate speech and abusive language detection datasets." arXiv preprint arXiv:1905.12516 (2019).

[15] Russell, S. J., Norvig, P. (2009). *Artificial Intelligence: A Modern Approach*. Pearson Education.

[16] Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill Education.

[17] Hastie, T., Tibshirani, R., Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

[18] Goodfellow, I., Bengio, Y., Courville, A. (2016). *Deep Learning*. MIT Press.

[19] LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.

[20] Jurafsky, D., Martin, J. H. (2019). *Speech and Language Processing*. Prentice Hall.

[21] Goldberg, Y. (2017). *Neural Network Methods for Natural Language Processing*. Morgan Claypool Publishers.

[22] Chen, Q., Zhu, X., Zhu, S., Song, K. (2020). Time-Aware Multi-Modal Sentiment Analysis with Hierarchical Aggregation Transformer. arXiv preprint arXiv:2012.07905.

[23] Li, X., Liu, Y., Wang, X., Chen, Z., Huang, L., and Yang, J. (2021). A Study on Attitudes Toward Mask Wearing during the COVID-19 Pandemic Based on Online Comments. Journal of Medical Systems, 45(3), 1-12.

[24] Rodrigues, F., Markou, I., Pereira, F. Combining time-series and textual data for taxi demand prediction in event areas: a deep learning approach. In Information Fusion, Elsevier, 2018

[25] Chen, Y. H., Chen, M. Y., Hsieh, Y. J., Yang, Y. H. (2021). Multilingual Abstractive Summarization with Cross-Lingual Pretraining and Knowledge Distillation. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume (pp. 2265-2275).

[26] Arora, Aditi, Sarcasm Detection in Social Media: A Review (December 15, 2020). Proceedings of the International Conference on Innovative Computing Communication (ICICC) 2021, Available at SSRN: https://ssrn.com/abstract=3749018 or http://dx.doi.org/10.2139/ssrn.3749018

[27] Vijayarani Mohan, "Ontology Based Information Extraction -A Survey", October 2016.

[28] Hyndman, R. J., Athanasopoulos, G. (2018). Forecasting: principles and practice. OTexts.

[29] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[30] Bontempi, G., Taieb, S.B., and Le Borgne, Y.A. (2012). Machine learning strategies for time series forecasting. In Business intelligence (pp. 62-77). Springer, Berlin, Heidelberg.

[31] Hyndman, R.J., and Athanasopoulos, G. (2018). Forecasting: principles and practice. OTexts.

[32] Shmueli, G., and Lichtendahl Jr, K.C. (2019). Data Mining for Business Analytics: Concepts, Techniques, and Applications with JMP Pro. John Wiley Sons.

[33] Shumway, R. H., Stoffer, D. S. (2017). Time series analysis and its applications: with R examples. Springer.

[34] Box, G. E. P., Jenkins, G. M., Reinsel, G. C., Ljung, G. M. (2015). Time series analysis: forecasting and control. Wiley.

[35] Bengio, Y., Simard, P., & Frasconi, P. *Learning long-term dependencies with gradient descent is difficult.* IEEE Transactions on Neural Networks, 5(2), 157-166, 1994.

[36] Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[37] Connor, J. T., Martin, R. D., & Atlas, L. E. *Recurrent neural networks and robust time series prediction*. IEEE Transactions on Neural Networks, 5(2), 240-254, 1994.

[38] Malhotra, P., Vig, L., Shroff, G., & Agarwal, P. *Long Short-Term Memory networks for anomaly detection in time series*. Proceedings of the 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2015.

[39] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

[40] Graves, A., Mohamed, A. r., Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In 2013 IEEE international conference on acoustics, speech and signal processing (pp. 6645-6649). IEEE.

[41] Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

[42] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).

[43] Zhang, X., Zhao, J., LeCun, Y. (2015). Character-level convolutional networks for text classification. In Advances in neural information processing systems (pp. 649-657).

[44] Chung, J., Gulcehre, C., Cho, K., Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

[45] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E. (2016). Hierarchical attention networks for document classification. In Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies (pp. 1480-1489).

[46] Dai, A. M., Le, Q. V. (2015). Semi-supervised sequence learning. In Advances in neural information processing systems (pp. 3079-3087).

[47] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L. (2018). Deep contextualized word representations. arXiv preprint arXiv:1802.05365.

[48] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2019.

[49] Bengio, Y., Simard, P., Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks, 5(2), 157-166.

[50] LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

[51] Zhao, R., Wang, Y., Qi, H. (2017). An introduction to deep learning for the physical layer. IEEE Transactions on Cognitive Communications and Networking, 3(4), 563-575.

[52] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 1725-1732).

[53] Graves, A., Mohamed, A. r., Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In 2013 IEEE international conference on acoustics, speech and signal processing (pp. 6645-6649). IEEE.

[54] Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., Muller, P. A. (2019). Deep learning for time series classification: a review. Data Mining and Knowledge Discovery, 33(4), 917-963. doi: 10.1007/s10618-019-00619-1

[55] Fanaee-T, H., Gama, J. (2016). Event labeling combining ensemble detectors and background knowledge. Progress in Artificial Intelligence, 5(4), 277-288.

[56] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018.

[57] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. OpenAI, 2019.

[58] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language Models are Few-Shot Learners. In Advances in Neural Information Processing Systems, 2020.

[59] OpenAI. Introducing ChatGPT: An AI Language Model for Generating Human-like Conversations. OpenAI Blog, 2021.

[60] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 2021.

[61] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model Cards for Model Reporting. In Proceedings of the Conference on Fairness, Accountability, and Transparency, 2019.

[62] Wang, S., Liu, W., Liu, X., Zhao, R., Yang, X. (2020). A deep learning based approach to COVID-19 tweet classification. *IEEE Journal of Biomedical and Health Informatics*.

[63] Ghafouri-Fard S, Mohammad-Rahimi H, Motie P, Minabi MAS, Taheri M, Nateghinia S. Application of machine learning in the prediction of COVID-19 daily new cases: A scoping review. Heliyon. 2021 Oct;7(10):e08143. doi: 10.1016/j.heliyon.2021.e08143. Epub 2021 Oct 11. PMID: 34660935; PMCID: PMC8503968.

[64] K. Seresirikachorn, P. Ruamviboonsuk, N. Soonthornworasiri, P. Singhanetr, T. Prakayaphun, N. Kaothanthong, S. Somwangthanaroj, and T. Theeramunkong, "Investigating public behavior with artificial intelligence-assisted detection of face

mask wearing during the COVID-19 pandemic," *PLOS ONE*, vol. 18, no. 4, pp. 1-13, Apr. 2023. doi: 10.1371/journal.pone.0281841.

[65] Talic S, Shah S, Wild H, Gasevic D, Maharaj A, Ademi Z et al. Effectiveness of public health measures in reducing the incidence of covid-19, SARS-CoV-2 transmission, and covid-19 mortality: systematic review and meta-analysis BMJ 2021; 375 :e068302 doi:10.1136/bmj-2021-068302

[66] Khadjeh Nassirtoussi, Arman Aghabozorgi, Sr Wah, Teh Ngo, David. (2014). Text mining for market prediction: A systematic review. Expert Systems with Applications. 41. 7653-7670. 10.1016/j.eswa.2014.06.009.

[67] Z. Gao, "Stock Price Prediction With ARIMA and Deep Learning Models," 2021 IEEE 6th International Conference on Big Data Analytics (ICBDA), Xiamen, China, 2021, pp. 61-68, doi: 10.1109/ICBDA51983.2021.9403037.

[68] Priyank Sonkiya, Vikas Bajpai, and Anukriti Bansal, "Stock price prediction using BERT and GAN," arXiv preprint arXiv:2107.09055 [q-fin.ST], 2021.

[69] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[70] Li, L., Liu, H., Zhang, J., Sun, Y. (2021). Stock price prediction based on pre-trained language model BERT. Applied Intelligence, 51(1), 300-313.

[71] Zhixiong Tan, Bihuan Chen, and Wei Fang. 2020. Analysis and Application of Financial News Text in Chinese Based on Bert Model. In Proceedings of the 2020 Asia Service Sciences and Software Engineering Conference (ASSE '20). Association for Computing Machinery, New York, NY, USA, 35–39. https://doi.org/10.1145/3399871.3399886

[72] Kim, Jina Ji, HongGeun Oh, Soyoung Hwang, Syjung Park, Eunil del Pobil, Angel P.. (2020). A deep hybrid learning model for customer repurchase behavior. Journal of Retailing and Consumer Services. 59. 102381. 10.1016/j.jretconser.2020.102381.

[73] Z. Hu, J. Hu, W. Ding, and X. Zheng, "Review sentiment analysis based on deep learning," ICEBE 2015 IEEE 12th International Conference, pp. 87-94, 2015.

[74] S.I.A. HuiD, T.A. Madani, F. Ntoumi, R. Koch, O. Dar, The continuing 2019-nCoV epidemic threat of novel corona viruses to global health: the latest 2019 novel coronavirus outbreak in Wuhan, China, Int. J. Infect. Dis. 91 (2020) (2020) 264–266.

[75] Ronan Collobert, Jason Weston, L´eonBottou, Michael Karlen, KorayKavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. JMLR, 12:2493–2537, 2011.

[76] Zhang, Lei, Shuai Wang, and Bing Liu. "Deep learning for sentiment analysis: A survey." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8.4 (2018): e1253.

[77] I. Yeo and K. Balachandran, "Sentiment Analysis on Time-Series Data Using Weight Priority Method on Deep Learning," 2019 International Conference on Data Science and Communication (IconDSC), 2019, pp. 1-7. doi: 10.1109/IconDSC.2019.8816985.

[78] W. Zhao, Z. Guan, L. Chenm X. He, D. Cai, B. Wang, and Q. Wang, "Weakly-supervised deep embedding for product review sentiment analysis," IEEE Transactions on Knowledge and Data Engineering, vol. 30, pp. 185-197, 2017

[79] X. Zhang, J. Zhao, and Y. LeCun. "Character-level convolutional networks for text classification," NIPS'15, vol. 1, pages 649-657, 2015

[80] S. N. Singh and T. Sarraf, "Sentiment Analysis of a Product based on User Reviews using Random Forests Algorithm," 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2020, pp. 112-116, doi: 10.1109/Confluence47617.2020.9058128.

[81] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1631-1642, 2013.

[82] Y. Kim, "Convolutional Neural Networks for Sentence Classification," EMNLP'14, pp. 1746-1751, 2014

[83] Hochreiter, Sepp & Schmidhuber, Jürgen. (1997). Long Short-term Memory. Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735.

[84] W. Saena and V. Suttichaya, "Predicting Drug Sale Quantity Using Machine Learning," 2019 14th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP), 2019, pp. 1-6, doi: 10.1109/iSAI-NLP48611.2019.9045222.

[85] Edward Ma. NLP Augmentation, 2019. https://github.com/makcedward/nlpaug .

[86] Barry, C. L., Anderson, K. E., Han, H., Presskreischer, R., & McGinty, E. E. (2021). Change over time in public support for social distancing, mask wearing, and contact tracing to combat the COVID-19 pandemic among US adults, April to November 2020. American journal of public health, 111(5), 937-948.

[87] Chris Olah. Understanding LSTM Networks, 2015. https://colah.github.io/posts/2015-08- Understanding-LSTMs/.

Appendix B

Curriculum Vitae

Personal Information

Full name: Jingtian Zhao
Email: jzhao26@lakehead
Phone:705-975-296

Education

M.Sc in Computer Science, Lakehead University, 2020 - current
B.Sc in Computer Science, Algom University, 2016-2020

Related Work

Graduate Assistant
Teaching Assistant

Experience

Mitacs project, PTPA Media, September 2021 - February 2022
Programmer Analyst, OLG, May 2018 - September 2018

Publication

J. Zhao, W. Zhao, Y. Yang, A. Safaei and R. Wei, "To Mask or Not To Mask? A Machine Learning Approach to Covid News Coverage Attitude Prediction Based on Time Series and Text Content," in 2022 IEEE 25th International Conference on Computational Science and Engineering (CSE)