International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 /doi:10.14445/22315381/IJETT-V69I5P226
Volume 69 Issue 5, 190-195, May 2021

Collecting Targeted Information About Covid-19 From Research Papers By Asking Questions Based On Natural Language Processing

Abdirahman Osman Hashi[#*1], Octavio Ernesto Romo Rodriguez[*2], Abdullahi Ahmed Abdirahman[#3], Mohamed M. Mohamed[#4]

[#]*Faculty member, SIMAD University, Department of Computing Mogadishu, Somalia*

[*]*Department of Computer Science, Faculty of Informatics, İstanbul Teknik Üniversitesi, İstanbul, Turkey*

[1]wadani12727@gmail.com, [2] oct_romo@hotmail.com, [3]aaayare@simad.edu.so

Abstract – *In the general framework of knowledge discovery, different techniques were used for information extraction from multi-label documents. As the world is currently facing COVID-19, it has made it more important than ever to have such knowledge extraction from previous documents. Therefore, Natural Language Processing (NLP) can be an essential model for tackling such an issue. By taking into consideration that having such a model plays an essential role to generate new insights in support of the ongoing fight against this infectious disease. This work introduces a sophisticated model that is able to read data from various articles about COVID-19, and finally give the most appropriate answer to the questions asked in order to gain insight information automatically. The model is applied to COVID-19 open research dataset challenge (CORD-19) that's has caught the attention of many researchers and it contains over 400,000 scholarly articles. The result of the proposed model has shown a good achievement, as it is explained in the result section. It was found that NLP is a good choice for tackling this global pandemic for information extraction and it contribute a new insight in support of the ongoing fight against this infectious disease.*

Keywords *— Coronavirus; Natural Language Processing; Deep Learning, Artificial Intelligence.*

I. INTRODUCTION

The infectious disease, COVID-19, has become a global priority for research on the subject to obtain more information that can make easier to fight against it[1]. Researchers in a variety of fields, from infectious diseases to clinics, should all have access to the most up-to-date information from the latest publications in their field [2]. To speed up research effort and process, it is important to have a knowledge extraction that can be retrieved from the previous articles by classifying them based on the searched questions such as (by field, topic or specific answer that researchers looking from the article). To present how knowledge discovery can help retrieving information automatically from COVID-19 research articles, the dataset chosen was the COVID-19 open research dataset challenge (CORD-19) that has over 400,000 scholarly articles about COVID-19, SARS, CoV-2 and related coronavirus. It is a free dataset for researchers to apply to the field Natural Language Processing and Artificial Intelligence techniques to find out new information that will make easier to take part in the efforts against coronavirus. It is crucial to have a system that can enable an easier extraction of the information needed from multiple articles using Natural Language Processing [3]. It will play an important role in facilitating the search process related to knowledge discovery as well as search engines. BERT, which is one of the models used in the retrieve automatic answer from documents, will be the one selected for this work. Finally, the required answers will be retrieved from a wide range of documents. This, as mentioned earlier, is important in contributing to the fight against COVID-19, as it will make it easier for researchers to find the information needed, and not waste time reading each article.

II. RELATED WORK

The most significant advantage of utilizing Web Ontology Language (OWL) is more significant semantic efficiency that allows knowledge to be more relevant. This knowledge base would need to use presuppositions and inference components to interpret semantics and realize the Semantic Web domain's enriched content. As published COVID-19 Open Research Dataset (CORD-19) issued by the Allen Institute for AI, a system has been developed to help researchers and the public to access valuable information about COVID 19 by using Search CORD-19 tool which is a search engine that uses CORD-19 data generated by Amazon Comprehend Medical [4][5]. Google has released COVID19 Research Explorer, a top-notch search tool for CORD-19 data. Meanwhile, Covidex Implements multi-step search framework that can extract various data functions [6]. NLP health issue WellAI COVID-19 Research Tool that can create a list of health concepts with a series of options related to COVID-19, tmCOVID7 is a bioconcept. It is a tool for extraction and gaining a valuable information from COVID-19 literature[7][10].

In addition to accessing information, this proposed system will provide a relevant information that is the answer of what user request or asked the system by retrieving from the body of different articles. This paper is organized to be five sections, the next section will explain deeply the proposed methodology. The fourth section will explain the result of the proposed methodology. Finally, the fifth section describes the conclusions based on the results and outlines future work.

III. METHODOLOGY

As mentioned before, the main target of this approach is to gain insight information about COVID-19 by a list of questions and finding their answers using the text corpus that was formed with a number of papers' abstracts and bodies. Since the data that will be used contains non-labeled data, it was decided to address this problem by using models and algorithms designed for this kind of data, such as BERT embeddings.

The process of the current approach is fairly straightforward, it can be divided in the following steps:

- **Loading Data.** This consists of reading the metadata in csv format and the papers' data in json format. From the metadata (csv) the abstract of the papers were read, since they contain the essence of the information of the papers. As well as the papers' bodies from the papers' data, which is the biggest piece of data.
- **Pre-processing Data.** During this step it was necessary to clean the data gathered in the previous step, using regular expressions to remove citations, removing custom stop words, punctuation marks and such. After cleaning the data, the body of the paper was tokenized and stored in a Data Frame.
- **Implementing BERT Question-Answering.** A Question Answering system using the BERT model was implemented. The chosen frameworks were FARM (https://github.com/deepset-ai/FARM) and Haystack https://github.com/deepset- ai/haystack to speed up the implementation, since they count with predefined functionalities to set up a question answering and/or neural search system.
- **Result information.** After pre-processing the data and defining the BERT model, it is possible to get targeted information by making a list of questions, looping through it, and getting an answer for each question. The answers are searched in the text corpus constituted by the tokenized version of all the paper bodies and abstracts. Due to processing power limitations some questions might not have an answer, since it was possible to work only with 300 papers at the same time.

A. Loading Data

As mentioned previously, this step is quite straightforward. After analyzing the information in the dataset, it was found the most important sources of data for this approach, the abstract and body of the papers. The abstract is located in the metadata of the papers and the body of the papers in a JSON formatted file containing their raw information. Both files are using a different format, so two approaches where used to gather the data from them.

Dataset

Since the most recent dataset available was bigger than 7GB, it was decided to use a previous dataset due to processing limitations. The chosen dataset was the one uploaded on June 1, 2020. Besides of that, this dataset contains the paper bodies in a format that facilitates the task of data gathering for this specific attribute.

Metadata

The metadata was read by simply using the function "pd.read_csv" from Pandas. The following figure shows all the attributes contained in metadata:

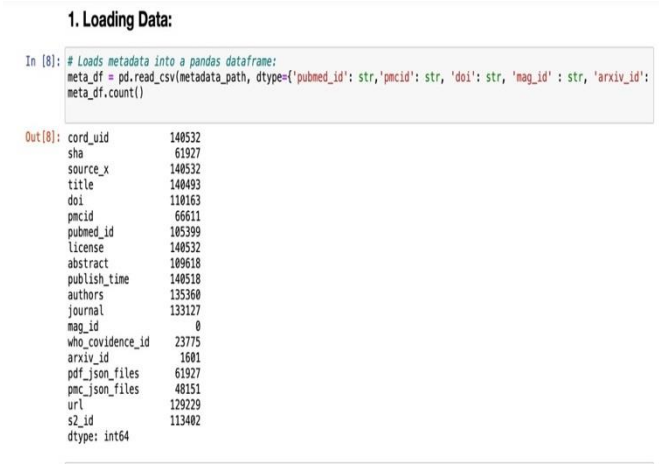

Fig. 1. Loading data

As can be seen, the loaded data has a considerable number of attributes such as cord_uid, title, doi, abstract, published time, authors and published journal. Although all the papers may not have all the detailed information, most of the paper have them and this makes them valuable information to be retrieved from each paper's content, making the process of search engines more efficient as well. Analyzing the metadata allows a much more detailed information from each paper's content. The mentioned information is shown in the following figure:

```
In [9]: # Checks data format:
        meta_df.head()
```

Out[9]:

	cord_uid	sha	source_x	title	doi	pmcid	pubmed_id	license	abstract	publish_time
0	ug7v899j	d1aafb70c066a2068b02786f8929fd9c900897fb	PMC	Clinical features of culture-proven Mycoplasma...	10.1186/1471-2334-1-6	PMC35282	11472636	no-cc	OBJECTIVE: This retrospective chart review des...	2001-07-04
1	02tnwd4m	6b0567729c2143a66d737eb0a2f63f2dce2e5a7d	PMC	Nitric oxide: a pro-inflammatory mediator in l...	10.1186/rr14	PMC59543	11667967	no-cc	Inflammatory diseases of the respiratory tract...	2000-08-15
2	ejx2xln0	06ced00a5fc0421594ba72528f2eeaae1d58927	PMC	Surfactant protein-D and pulmonary host defense	10.1186/rr19	PMC59549	11667972	no-cc	Surfactant protein-D (SP-D) participates in th...	2000-08-25
3	2b73a28n	348055649b6b8cf2b9a376498df9bf41f7123605	PMC	Role of endothelin-1 in lung disease	10.1186/rr44	PMC59574	11686671	no-cc	Endothelin-1 (ET-1) is a 21 amino acid peptide...	2001-02-22
4	9785vg6d	5f48792a5fa08bed9f560164981ae2ca6031b32	PMC	Gene expression in epithelial cells in respons...	10.1186/rr61	PMC59580	11686888	no-cc	Respiratory syncytial virus (RSV) and pneumoni...	2001-05-11

Fig. 2. Metadata of the articles

Papers' Bodies

Gathering metadata is easy, on the other hand, in order to obtain information from the JSON files it was required to define a new function that could ignore and combine the targeted information:

```
In [10]: # Loads data from json:
         glob_json = glob.glob(f'{pdfs_path}/**/*.json', recursive=True)
         print(f'found articles: {len(glob_json)}')

found articles: 65782
```

Fig. 3. Json found articles

Here it is possible to see that there are 65,782 JSON files.

Function to read the files and a showcase of what it retrieves:

```
# Reads json files, combines abstracts and bodies:
class ArticleReader:
    def __init__(self, file_path):
        with open(file_path) as file:
            content = json.load(file)
            self.paper_id = content['paper_id']
            self.abstract = []
            self.body_text = []
            for entry in content['abstract']:
                self.abstract.append(entry['text'])
            for entry in content['body_text']:
                self.body_text.append(entry['text'])
            self.abstract = '\n'.join(self.abstract)
            self.body_text = '\n'.join(self.body_text)
    def __repr__(self):
        return f'{self.paper_id}: {self.abstract[:300]}... {self.body_text[:300]}...'
first_row = ArticleReader(glob_json[0])
print(first_row)
```

```
4fcb95cc0c4ea6d1fa4137a4a087715ed6b68cea: Abnormal levels of end-tidal carbon dioxide (EtCO 2 ) during resuscitati
on in the delivery suite are associated with intraventricular haemorrhage (IVH) development. Our aim was to determ
ine whether carbon dioxide (CO 2 ) levels in the first 3 days after birth reflected abnormal EtCO 2 levels in the
... Improvements in neonatal intensive care have resulted in decreased mortality rates of preterm infants. The dev
elopment of intraventricular haemorrhage (IVH), however, can result in long-term adverse outcomes. Several studies
of preterm infants have demonstrated that abnormal levels of carbon dioxide...
```

Fig. 4. Combined abstract and bodies

The function loops through every document in "pdfs_path," getting the body of each paper, linking it to its abstract and making a DataFrame with it. We can see that the number of items here is 58,396. The discrepancy of abstracts in metadata and bodies in the JSON data is because not all the papers have a body attribute. Hence, the data was leveled by adding only the papers that have a body to the final DataFrame.

```
In [15]: # Processes 300 articles to create a DataFrame:
         # Maps the original attributes of metadata:
         dict_ = {'paper_id': [],'abstract': [], 'body_text': []}
         for i, entry in enumerate(glob_json):
             content = ''
             meta_data = ''
             try:
                 content = PdfReader(entry)
                 # Gets metadata of paper:
                 meta_data = meta_df.loc[meta_df['sha'] == content.paper_id]
             except Exception as e:
                 print(f'skipping article {i}...')
                 continue
             # Metadata not found, skips paper:
             if len(meta_data) == 0:
                 continue
             # Adds info to dictionary:
             dict_['abstract'].append(content.abstract)
             dict_['paper_id'].append(content.paper_id)
             dict_['body_text'].append(content.body_text)

         df_covid = pd.DataFrame(dict_, columns=['paper_id', 'abstract', 'body_text' ])
         print(df_covid.count())

paper_id     58396
abstract     58396
body_text    58396
dtype: int64
```

Fig. 5. Matching articles into the data frame.

The result of this step is a DataFrame that contains 'paper_id', 'abstract' and 'body_text'.

B. Pre-processing Data

Fig. 6. Sentence to Token

During this step, the data that was gathered in the previous step was transformed and processed to get it ready for the

```
def split_sentences(article):
    import nltk.data
    import re
    tokenizer = nltk.data.load('tokenizers/punkt/english.pickle')
    list_df = []
    m_sentences = '***'.join(tokenizer.tokenize(article))
    # Uses regex to remove citations:
    m_sentences = re.sub(r"(\([0-9]*\))",'',m_sentences)
    m_sentences = re.sub(r"(\([0-9]*\))",'',m_sentences)
    list_df  = m_sentences.split('***')
    return list_df
```

```
def sentence_to_tokens(doc):
    tokens = []
    sentences = [nltk.word_tokenize(sents) for sents in doc]
    for sentence in sentences:
        filtered_tokens = []
        filtered_tokens =[words for words in sentence if words not in stop_words]
        # Removes custom stop words:
        custom_removed_tokens = [words for words in filtered_tokens if words not in ['.','shown','fig.','figure']]
        custom_removed_tokens = [words for words in custom_removed_tokens if len(words) > 2]
        tokens.append(custom_removed_tokens)
    return tokens
```

BERT model. The first part of this step is, essentially, a deeper cleaning of the data. Since the data will be transformed into tokens, first of all it is necessary to remove the possible instances of citations, custom stop words and punctuation marks, a fairly standard process.

Once the data is clean enough, it can be tokenized it and stored as a new DataFrame, keeping the previous and adding new columns for the new data.

```
def preprocess_sentences(df):
    # Checks body_text of paper:
    df_sent = df['body_text'].apply(lambda x: split_sentences(x))
    df_lower = df_sent.apply(lambda doc: [sent.lower() if type(sent) == str else sent for sent in doc])
    # Removes punctuation marks:
    df_clean = df_lower.apply(lambda doc: [re.sub(r"[^.a-z]",' ',sent) for sent in doc])
    # Tokenizes sentences:
    df_token = df_clean.apply(lambda doc:sentence_to_tokens(doc))
    preprocess_df=pd.concat([df,df_sent],axis = 1)
    preprocess_df=pd.concat([preprocess_df, df_token], axis=1)
    # Adds columns with new information, keeps previous information:
    preprocess_df.columns=['paper_id','original_body', 'abstract','sentences','token_sentences']
    return preprocess_df
```

Fig. 7. Pre-processing Sentences

The result of this step is a new DataFrame, containing the new clean data:

```
preprocess_df = preprocess_sentences(df_covid)
```

C. Implementing BERT Question-Answering

For this step, pre-defined standards in the FARM and Haystack frameworks were used, as well as the tokenized bodies and abstracts of the publications to generate a question-answering system that can take as input the pre-processed data as a text pool and a question in natural language. It tries to match a possible answer to the question and return it as plain text. The logic of the system is defined in the following cells.

The function "get_finder" returns a Haystack finder, this is basically an adaptable pre-trained model for BERT, it takes the DataFrame and a model to define its strategy:

```
# Returns a haystack finder based on transformers:
def get_finder():
    retriever = TfidfRetriever(document_store=df_covid)
    reader = FARMReader(model_name_or_path="gdario/biobert_bioasq", use_gpu=False)
    finder = Finder(reader, retriever)
    return finder
```

Fig. 8. Haystack finder based on transformer

The function "get_answer" uses the previously received Haystack finder and it uses it besides the already defined the text corpus to look for an answer for the question in the text corpus. In order to do that, the model needs to embed the text corpus and the questions using the BERT model, like so, the tokenized words have an enriched meaning, since they are context dependent.

```
# Returns the first result found by the finder:
def get_answer(finder, top_k_retriever, top_k_reader, question):
    paragraphs = text_corpus
    meta_data = finder.retriever.retrieve(question, top_k=top_k_retriever)
    result = ''
    if len(paragraphs) > 0:
        # Applies reader to get granular answer
        len_chars = sum([len (p) for p in paragraphs])
        predictions = finder.reader.predict(question=question,
            paragraphs=paragraphs,
            meta_data_paragraphs=meta_data,
            top_k=top_k_reader)
        # Gets the first answer only:
        if not predictions["answers"]:
            result = 'answer not found...'
        else:
            result = predictions.answers[0]
    return result
```

Fig. 9. Result from finder implementation

The result of this step is the definition of the Question-Answering model. This model will be used in the next step.

IV. RESULT & DISCUSSION

For this research the results are not numerical, since we worked with non-labelled data. However, some numerical values can be deduced to depict better the accuracy of the results. The result for this research is in the form of specific information retrieved from the research papers about COVID-19. This was achieved by defining a list of questions related to COVID-19, looping through the questions after training the model and getting an answer for each question. The answers are searched in the text corpus constituted by the tokenized version of all the papers' bodies and abstracts. In some cases, there might be more than one answer for a given question, but to keep things simple, and not to get redundant information, the system only takes the first answer that it finds. On the other hand, some questions might not have an answer, since the model was trained with only 300 papers at a time due to limited processing capabilities. The obtained result has shown that the model achieved a good performance which is related to the questions that were asked. This can take part to speed up and simplify the efforts against this infamous virus, for which many researchers are trying to come up with a novel idea that can gather the appropriate information from selected scientific papers. The upcoming figure demonstrates the result of this model, which is correlated with the asked question. Since a clear and direct question helps the model to match the context of the question with the text corpus, achieving better results.

```python
# Defines list of questions:
questions = [
    "What is the incubation period of coronavirus",
    "Is the transmission of coronavirus seasonal?",
    "Is the transmission of coronavirus influenced by humidity?",
    "What factors influence survival of coronavirus in the environment?",
    "Which materials does the virus persist on?",
    "How do people get infected with SARS-CoV-2?",
    "How do infected people transmit SARS-CoV-2?",
    "How is the immune response to SARS-CoV-2?",
    "How is immunity to SARS-CoV-2 developed?",
    "Is personal protective equipment effective at reducing risk of transmission of SARS-CoV-2?",
    "Is transmission of SARS-CoV-2 dependent on environmental factors?",
]
```

Fig. 10. List of asked questions

```python
for question in questions:
    answer = ''
    try:
        answer = get_answer(finder, top_k_retriever, top_k_reader, question)
        answers.append(answer)
    except Exception as e:
        answers.append("answer not found...")
    if not answer:
        answers.append("answer not found...")

for i, question in enumerate(questions):
    print(f'question: 'questions[i])
    print(f'answer: 'answers[i])
```

Fig. 11. Loop to answer questions

```
question: What is the incubation period of coronavirus
answer: as follows: diagnosis, mode of transmission, long incubation period (3 to 14 days), predicting the number
of infected cases in the community, and insu
question: Is the transmission of coronavirus seasonal?
answer: This study provides preliminary evidence that there may be seasonal variability in transmission of SARS-Co
V-2, but this analysis does not imply that
question: Is the transmission of coronavirus influenced by humidity?
answer: higher in cold and dry weather, but unlike influenza, it is also affected by absolute humidity [58] . Cont
act rates were also modified during weekend
question: What factors influence survival of coronavirus in the environment?
answer: answer not found...
question: Which materials does the virus persist on?
answer: mostly unchanged all throughout the infection in bronchial HAE but it is highly upregulated in nasal HAE a
t 48 hpi and onwards ( Fig. 3C and Data S1).
question: How do people get infected with SARS-CoV-2?
answer: answer not found...
question: How do infected people transmit SARS-CoV-2?
answer: ociated with the SARS-CoV-2 and a retrospective study 44 indicated person-to-person transmission [5] . Mor
eover, people seem to be 45 generally suscep
question: How is the immune response to SARS-CoV-2?
answer: hat hospitals deal with on an annual basis, and transmission seems to occur mainly through droplets. Even
a high estimate of the R 0 rate is many ti
question: How is immunity to SARS-CoV-2 developed?
answer: to be driven by acute respiratory distress syndrome (ARDS) and a dysregulated immune response to SARS-CoV-
2 2-4
question: Is personal protective equipment effective at reducing risk of transmission of SARS-CoV-2?
answer: answer not found...
question: Is transmission of SARS-CoV-2 dependent on environmental factors?
answer: the following winter. However, whether SARS-CoV-2 transmission will be affected by seasonal variations rem
ains unclear. Although many infectious dise
```

Fig. 12. Result of proposed framework

As shown above, the answers are not formatted but rather taken directly as they were in their paper of origin, that also adds some context to the answer given. It is also noticeable that some questions do not have an answer, hence the sentence "answer not found…" is retrieved.

Table 1. Results showcase

Totals Questions		
Found Answer	**Not Found Answer**	**Total**
8	3	11

As it was mentioned before, due to processing limitations it was possible to use only 300 documents per execution. Nevertheless, with this limited text corpus it was possible to achieve 72% of discovery ratio for the asked questions, therefore it is logical to assume that a bigger text corpus can achieve a better performance.

V. CONCLUSION

This research was a more real-life approach to implementing a system that uses deep learning. It was developed with a real-life dataset, target and scenario. One of the most important things that was discovered was that in the current situation we live in, having too much information becomes a problem because there are no tools to use it properly. We have a lot of knowledge that gets wasted due to lack of resources to use it. Specially in the area of NLP, we are used to analyze labeled data, training models using a target class and getting an overview of the information that we have. Nonetheless, in a real- life scenario we do not have the luxury of labeled data all the time. In fact, most of the data that can be obtained is not labeled nor organized. We will deal with non-labeled data eventually, so learning how to deal with it is a must.

Moreover, the approach that was taken in this research shows the importance and utility of modern methodologies and models to gather information from non-labeled datasets. It is truly exciting to get a useful result and to be able to find hidden information that can be used for a good cause such as scientific researches.

ACKNOWLEDGMENT

We, the authors of this paper, would like to extend our gratitude to Istanbul Technical University for making this research possible by offering their support, as well as SIMAD University for funding it. We would also like to give special thanks to Dr Lida Ghahremanlou for her work in the area of NLP that inspired this work.

References

[1] Awasthi, R., Pal, R., Singh, P., Nagori, A., Reddy, S., Gulati, A., ... & Sethi, T. (2020). CovidNLP: a web application for distilling systemic implications of COVID-19 pandemic with Natural Language processing. MedRxiv.

[2] Chen, Q., Leaman, R., Allot, A., Luo, L., Wei, C. H., Yan, S., & Lu, Z. (2020). Artificial Intelligence (AI) in Action: Addressing the COVID-19 Pandemic with Natural Language Processing (NLP). arXiv preprint arXiv:2010.16413.

[3] Li, M. D., Lang, M., Deng, F., Chang, K., Buch, K., Rincon, S., ... & Kalpathy-Cramer, J. (2021). Analysis of stroke detection during the COVID-19 pandemic using natural language processing of radiology reports. American Journal of Neuroradiology, 42(3), 429-434.

[4] Wang, L. L., Lo, K., Chandrasekhar, Y., Reas, R., Yang, J., Eide, D., ... & Kohlmeier, S. (2020). Cord-19: The covid-19 open research dataset. ArXiv.

[5] Wolf, M. S., Serper, M., Opsasnick, L., O'Conor, R. M., Curtis, L., Benavente, J. Y., ... & Bailey, S. C. (2020). Awareness, attitudes, and actions related to COVID-19 among adults with chronic conditions at the onset of the US outbreak: a cross-sectional survey. Annals of internal medicine, 173(2), 100-109.

[6] Voorhees, E., Alam, T., Bedrick, S., Demner-Fushman, D., Hersh, W. R., Lo, K., ... & Wang, L. L. (2021, February). TREC-COVID: constructing a pandemic information retrieval test collection. In ACM SIGIR Forum 54(1) 1-12). New York, NY, USA: ACM.

[7] Esteva, A., Kale, A., Paulus, R., Hashimoto, K., Yin, W., Radev, D., & Socher, R. (2021). COVID-19 information retrieval with deep-learning based semantic search, question answering, and abstractive summarization. npj Digital Medicine, 4(1), 1-9.

[8] Brunese, L., Mercaldo, F., Reginelli, A., & Santone, A. (2020). Explainable deep learning for pulmonary disease and coronavirus COVID-19 detection from X-rays. Computer Methods and Programs in Biomedicine, 196, 105608.

[9] Hashi, Abdirahman Osman, et al. A Robust Hybrid Model Based on Kalman-SVM for Bus Arrival Time Prediction. International Conference of Reliable Information and Communication Technology. Springer, Cham, 2019.

[10] Hameed, Shlilan S., et al. Filter-wrapper combination and embedded feature selection for gene expression data. Int. J. Advance Soft Compu. Appl 10.1 (2018)