University of East London

Pioneering Futures Since 1898

Classification and evaluation of spam emails using Naïve Bayes classifier and employing Pyspark

Research Project I (50XXXX-3)

An MSc Research Project submitted as partial fulfillment to

Graduate Studies Committee

Under the Guidance of

Prof. XYZ

By

Mr. Jeevan Reddy

May 2022

Table of Content

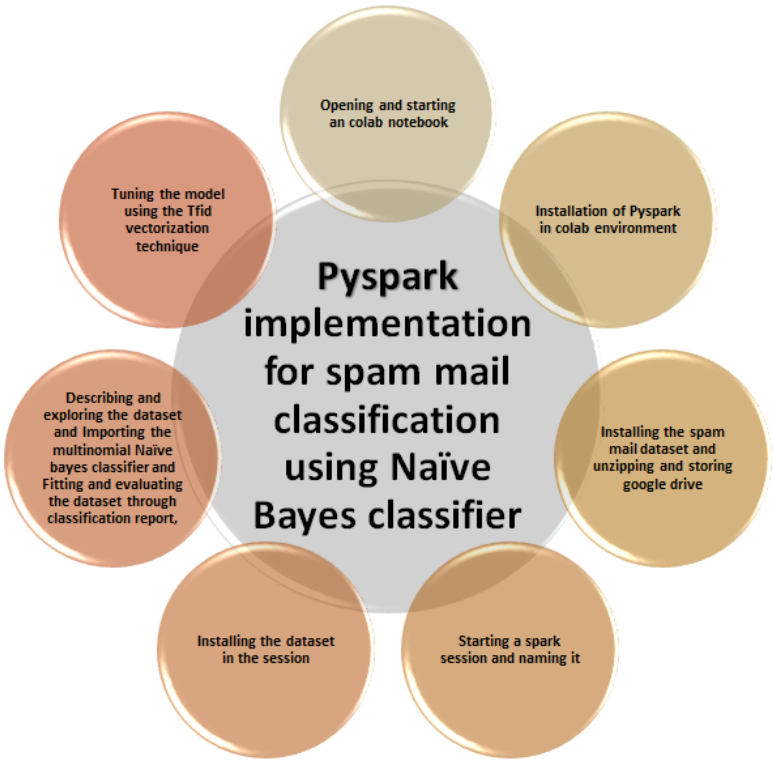
List of Figures

1 Introduction

1.1 Overview

Big data is widely used to compare the performance and operations of the company through personalized solution (Javaid *et al.* 2021). The dimensions of data produced in the real world has increased and require maintainance, cleaning and processing to be employed extensively. There are also technologies to gain information about big data and utilize them for classification tasks. One such technology identified here is Pyspark. This is considered an interface for the famous Apache spark for such big data.

Here in this study the efforts are made to evaluate the working and explore the various features of Pyspark and its core for maintain the big data. Pyspark also helps interface the pandas API for the data maintained along with machine learning library for creating the working models. Not only this but the work is planned to make use of Machine learning algorithms for the classification task identified here. Hence, considering this an spam mail classification model using Naïve bayes classifier is implemented through Pyspark.


Pyspark implementation for spam mail classification using Naïve Bayes classifier
Opening and starting an colab notebook
Installation of Pyspark in colab environment
Installing the spam mail dataset and unzipping and storing google drive
Starting a spark session and naming it
Installing the dataset in the session
Describing and exploring the dataset and Importing the multinomial Naïve bayes classifier and Fitting and evaluating the dataset through classification report,
Tuning the model using the Tfid vectorization technique

nt

use of the Pyspark.

- Describing and exploring the dataset

Printing in the form of schema and methods from the pandas.

- Importing the multinomial Naïve bayes classifier

Importing from the sklearn package

- Fitting and evaluating the dataset through classification report

Metrics module is used from the sklearn package.

- Tuning the model using the Tfid vectorization technique

Feature extraction module is utilized here from the sklearn package.

1.6 Organization of the Thesis

The thesis is organised by considering the chapters and sub-chapters, the chapters are introduction, literature review, methodology, analysis and findings and lastly the conclusions drawn from the experiments.

The introduction is the chapter -1, the overview, aim and objectives are study are discussed. The purpose of the study as well as the methodology of the work carried is discussed here. The chapter-2 which is the literature review consists of the exploration of various works and making review points and carrying out the related work. The chapter -3 explains briefly about the methods used to get the results and explain about how the used methods are appropriate. The chapter-4 shows the results obtained from the analysis and discussion were made on how the results obtained are better satisfies the aim and objectives. Finally, in the chapter-5, the discussions and final conclusions were drawn from the results.

2 Literature Review

2.1 Overview

~~Cancer relapse is the relapse of the disease after the symptoms are suppressed and before the recovery phase takes over. Whereas cancer recurrence is the recurrence of cancer after the recovery phase has taken over mostly. The stage at which the treatment is given to the cancer patient and the success of such treatment is entirely unique to each patient. However, it can be said that the success of the treatment is dependent upon the original type of the tumour and the treatment used to treat it and the time since the end of the treatment.~~

2.2 Introduction to Pyspark and its background

PySpark is the collection of apache spark and python, which is an open-source distributed computing framework it is the set of libraries for the real-time, and data is processed on a large scale (Zecevic *et al.*2019). Where python is a general-purpose and high-level programming language. PySpark is a great language in which it performs the data analysis at scale by building the machine learning pipelines. According to Lokaadinugroho *et al.* (2021), It creates the ETLs (extract, transform, load). ETL is defined as a data integration process that combines data from multiple data sources into a single, consistent data that stores the data warehouse or other target system. Guo *et al.* (2018) suggest that PySpark provides scalable analyses and pipelines. There are different options provided in the PySpark. They are self-hosted, cloud providers, and vendor solutions. Packard *et al.* (2021) say that self-hosted is a cluster that uses bare metal machines or virtual machines. Cloud providers are the spark clusters and it is a third-party company that offers a cloud-based platform, infrastructure, application, or storage services. Vendor solutions is a software solution as well as vendor products which is able to operate and integrate with the software (Alsubaei *et al.*2019). The data frame used in the PySpark is the Spark data frame. It is the table distributed across the clusters which have the functionality of data frames like R and Pandas.

Initially, Spark is started by Matei Zaharia at UC (University of California) Berkeley's AMP Lab in 2009. Alamoudi *et al.* (2020) say that it is open-source and the BSD (Berkeley Source Distribution) license is given in 2010. In 2013 the project was donated to the Apache Software Foundation and it switched its license to Apache 2.0. in 2014 during February spark became the Top-level Apache project. Kapila *et al.* (2020) say that Apache Spark is the unified analytics

engine that is used for big data and machine learning. Ramsingh (2022) says that generally, PySpark is not a programming language but it is an API (Application programming interface) of python which is developed by Apache spark. Hou *et al.* (2018) Spark is used in the integration and works with RDD (Resilient Distributed Dataset) in the python programming language. Dai *et al.* (2019) suggest that this is generally used to perform the computations and tasks on the larger datasets and it is also used to analyze the data. Spark and RDDs (Resilient distributed datasets) were developed in 2012. In the distributed programs a particular linear dataflow structure is created in response to the limitations of the MapReduce cluster computing paradigm (Gong, 2021). MapReduce programs read input data from disk and map a function across the data which reduces the results of the map and stores the reduction.

There are different features consisted in PySpark such as Spark SQL, data frame, streaming, MLlib (machine learning library), and Spark Core. According to Aziz *et al.* (2019), there are some other features in PySpark like in-memory computation, the distribution process using parallelize and many cluster managers are used in this PySpark. While using the data frames

the clusters with the hierarchy achieved either by the significant clusters or by splitting a more massive cluster into smaller clusters. While the partitional cluster divides the single set of different objects into the nonoverlapping subsets.

2.4 Phishing attacks through spam mails

Phishing attacks are the practice of sending fraudulent communications that comes from reputable source. Salahdine and Kaabouch (2019) suggest their goal is to steal sensitive data like credit cards and login information. A phishing attack is a social engineering attack that has a great range of targets depending on the attacker. According to Shin *et al.* (2022) Phishing targets an attack on a specific individual. There are different strategies for spam emails. They are spam, phishing, spear phishing, spoofing, and pharming (Vayansky and Kumar, 2018). Spear phishing is one of the fraudulent methods that occurs when the obtained information is about the person's personal information from the websites or social networking sites, and customize into a phishing scheme. According to Otalbi and Alsuwat (2020), spoofing describes a criminal who gathers personal information from the organization or the business information. Pharming is a malicious website and resembles a legitimate website that is used to gather the usernames and the passwords. The author opines that these phishing strategies cause a huge loss to the people from the attackers.

Phishing is a way that cybercriminals steal confidential information such as online banking logins, credit cards, business login credentials, or passwords by sending fraudulent messages (Deora and Chudasama,2021). Phishing is a fraudulent email or other communication which is designed to lure a victim. This message looks like it comes from a trusted sender. The person

gets fooled and sends the confidential information which results in a fraud attack. According to Basit *et al.* (2021), there are more dangers that happen by phishing attacks. Sometimes attackers attack credit card information or other personal data for the purpose of financial gain. Alkhalil *et al.* (2021) say that phishing's main aim is to obtain the employee login information and the other details for the use of advanced attacks against a particular company. There are some attacks such as advanced persistent threats (APTs) and ransomware. However, the author opines that these attacks result in a huge loss for banking, credit cards, and business login credentials.

There are different ways to protect the information of a particular organization from phishing user education. According to Torani *et al.* (2019) education is most important for all people, high-level executives are the target of phishing. People must have the awareness to recognize the phishing email and what to do when they receive one. Pulin (2018) suggests that security technology is the single cybersecurity technology that can prevent phishing attacks. The organizations must take the layered approach to reduce the number of attacks that lessen the impact when they occurred. Suresh *et al.* (2022) suggests that some of the network security technologies should be implemented which include email and web security, malware protection, user behaviour monitoring, and access control.

- • Hence it was also found that the naive Bayes classifier was implemented using Pyspark with an accuracy of 98% however when the theory vectorisation technique was applied to the data set and when the naive Bayes classifier was implemented the accuracy recorded was 97%. But however, the false positive rate was decreased and the recall rates of identifying a spam classifier was also improved along with the precision values.

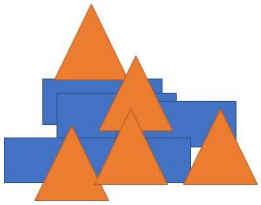
3 Methodology

3.1 Naïve Bayes Classifier

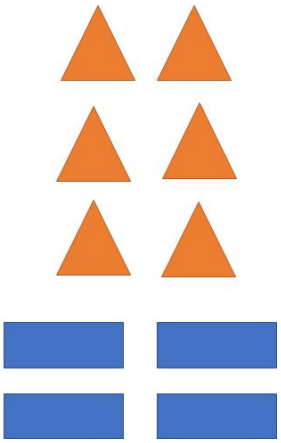
Naive Bayes classifier is one of the traditional and simplest form of classifier in the machine learning. This classifier finds its base on the simple conditional probability concept and Bayes theorem (Berrar, 2018). The Bayes theorem is also based on conditional probability. According to the theorem it states that the occurrences of an event is based on the likelihood of another event. The formula below shows the exact mathematical representation of the Bayes theorem (1). Where A and B are two events.

$$P\left(\frac{A}{B}\right) = \frac{P\left(\frac{B}{A}\right) * P(A)}{P(B)} \tag{3.1}$$

Hence, the classifier is based on the above formula incorporated with the machine language. However, as the study is based on classification of text type. It was found that multinomial naïve bayes classifier can be better used in the context (Kaur *et al*. 2022). The other reason why it can be used is that, the output labels is discrete here and, in such cases, Multinomial Naïve bayes is better to use as seen in study by (Gladence *et al.* 2015). In the same context, the features



$$P\left(\frac{A}{B}\right) = \frac{P\left(\frac{B}{A}\right) * P(A)}{P(B)}$$



rd

the words that have no specific meaning.

The raw data of the text usually contains the unimportant or unnecessary data of the text that contains the results which may not yield appropriate accuracy also makes it difficult to analyse and understand so that the data pre-processing usually done using the raw data. In python, NLTK is one of the open-source library that gives has got modules like classification, stemming, tokenization, tagging and so on. The Gensim is the open-source library is based on statistical semantics (Savytska *et al*. 2021). The statistical semantics provides an estimate for the words meanings that utilizes the statistical methods, just by seeing patterns of words in large texts collection. There is a module known as genism.parsing.preprocessing that consists of various methods for parsing and also pre-processing of strings. Some of the modules are present in pre-processing tools (text) is provided by sci-kit learn. The CountVectorizer() consists of pre-processing of text, tokenizing and also the filteration of stopwords. And the

various attributes in module CountVectorizer() are preprocessor, stop_words and tokenizer. It converts text documents to token counts matrix (Gupta *et al*. 2021).

Each and every text data requires various pre-processing steps. The initial step during pre-processing of data involves encoding in an appropriate format and then convert every upper case letters into a lower case. The computer processes both upper case and lower case as two different words. It is necessary to eliminate the punctuations and tags in the text data (Zelasko *et al*. 2021) as they do not contain any specific meaning. It is also necessary to eliminate all the numbers if present during preparation of data for basic sentiment analysis where generally the numbers are of no signifance. Also it is necessary to get rid of numerous whitespaces in between words during pre-processing (Naseem *et al.* 2021). It is essential to eradicate stopwords as stopwords generally are the most frequently used words. It can be used to enhance the performance a lot. The outcome strings could be utilized for future pre-processing in order to convert into numeric vectors utilizing the techniques of count vectorization.

4 Results and Discussions

4.1 Overview

For the purpose of the project, the data was obtained from the publicly available resource UCI Machine Learning repository. The data is downloaded using "!wget" from the resource, the Wget is a free command-line tool that lets the users to retrieve files from the internet sources, since the data is in a zipped format, therefore, the contents were unzipped using the "!unzip" method and later stored on to the cloud. For the project, the PySpark was used, because of its ability to handle larger DataFrames. PySpark is an open source, distributed computing framework with different set of libraries for real-time, large-scale. The standard libraries for any data science project such as the Pandas, NumPy, matplotlib and seaborn were also imported. The advantage of using PySpark DataFrame over Pandas DataFrame among others are discussed in this chapter.

For the NLP part of the project, the NLTK tool was imported and used, it is a known fact that NLTK is a standard python library that contains a wide variety of algorithms which can be used for the purpose of Natural Language Processing. It is essential for tasks such as classification, stemming, and tagging, semantic reasoning, parsing and tokenization.

It is a fact that every Natural Language Processing, and the tasks involved such as tokenization, text cleaning and other steps are unique to the data. And therefore, it is essential to gather sufficient data using the standard statistical procedures such that the biases in the data does not translate into algorithmic biases. Therefore, it is essential to maintain the statistical standards and ensure that the data does not contain any biases in any form, such that the bias in the data will not translate into the algorithmic bias. For the NLP project, the author utilised the following freely available resources.

```
[ ] import re
    import nltk
    nltk.download('stopwords')
    nltk.download('wordnet')
    from nltk.corpus import stopwords
    from nltk.stem import WordNetLemmatizer
    from nltk.stem.porter import PorterStemmer
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Unzipping corpora/wordnet.zip.
```

```
[ ]  # Importing pyspark libraries
     import pyspark
     from pyspark.sql import SparkSession
     from pyspark.conf import SparkConf
     from pyspark import SparkContext

     # Configuration of Spark Session
     spark = SparkSession.builder.master("local").appName("spam_classifier").getOrCreate()
     sc = spark.sparkContext
     sc
```

SparkContext

Spark UI

Version
v3.2.1
Master
local
AppName
spam_classifier

```
[ ]  df_spark.describe().show()
```

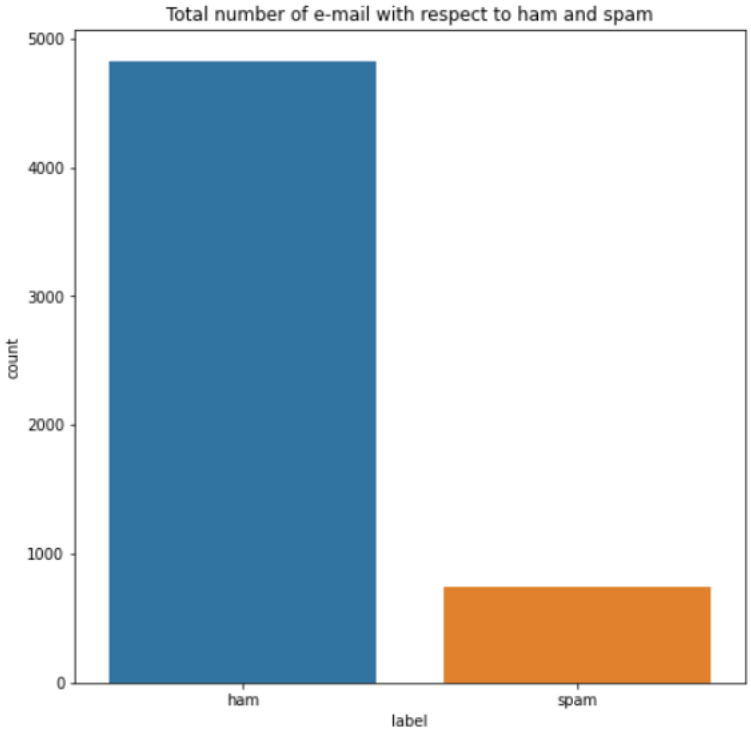
```
+-------+-----+--------------------+
|summary|label|             message|
+-------+-----+--------------------+
|  count| 5574|                5574|
|   mean| null|               645.0|
| stddev| null|                null|
|    min|  ham| &lt;#&gt;  in mc...|
|    max| spam|… we r stayin her...|
+-------+-----+--------------------+
```

```python
print(df_spark.show(5))
print('-+'*50+'\n')
print('What are the variable data types?')
print(df_spark.printSchema())
print('-+'*50)
print('How many observations do we have?')
print(df_spark.count())
```

```
+-----+--------------------+
|label|             message|
+-----+--------------------+
|  ham|Go until jurong p...|
|  ham|Ok lar... Joking ...|
| spam|Free entry in 2 a...|
|  ham|U dun say so earl...|
|  ham|Nah I don't think...|
+-----+--------------------+
only showing top 5 rows

None
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

What are the variable data types?
root
 |-- label: string (nullable = true)
 |-- message: string (nullable = true)

None
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
How many observations do we have?
5574
```

```
[ ]  # Counting Number of spam e-mails and ham e-mails

     plt.figure(figsize=(8,8))
     ax = sns.countplot(x=(df_spark.select('label').toPandas().label))
     plt.title('Total number of e-mail with respect to ham and spam')
```

```
Text(0.5, 1.0, 'Total number of e-mail with respect to ham and spam')
```



```
from sklearn.metrics import classification_report,confusion_matrix
print(classification_report(y_test_cv,y_pred_cv))
print(confusion_matrix(y_test_cv, y_pred_cv))
```

```
              precision    recall  f1-score   support

           0       0.99      0.98      0.99       950
           1       0.90      0.95      0.92       165

    accuracy                           0.98      1115
   macro avg       0.95      0.96      0.95      1115
weighted avg       0.98      0.98      0.98      1115

[[933  17]
 [  9 156]]
```

The data is again split using the train test split method. And the naive bayes classifier is fitted on to the training data “X_train_tfidf” and “y_train_tfidf”. The results obtained by the model on the testing data after performing “TF-IDF” vectorization is shown in the Fig. 4.7. From the results it has been observed that Naïve Bayes classifier model is able to attain an accuracy of 97%, with recall score of label “1” being 1.00 which indicates that the model is performing well when the data is vectorized using the TF-IDF vectorizer.

```
from sklearn.metrics import classification_report,confusion_matrix
print(classification_report(y_test_tfidf,y_pred_tfidf))
print(confusion_matrix(y_test_tfidf, y_pred_tfidf))
```

```
              precision    recall  f1-score   support

           0       0.96      1.00      0.98       950
           1       1.00      0.76      0.87       165

    accuracy                           0.97      1115
   macro avg       0.98      0.88      0.92      1115
weighted avg       0.97      0.97      0.96      1115

[[950   0]
 [ 39 126]]
```

Figure 4.7 Accuracy of the model on testing data (TF-IDFVectorizer).

5 Conclusions

5.1 Conclusions

PySpark is one of the best methods to handle large amounts of data since the Spark DataFrame is faster for a large amount of data, and also because of the ability of the PySpark to support parallelization. The author for the purpose of the project installed PySpark, and NLTK (Natural Language Toolkit). The NLTK is a Python package that can be used to analyze the unstructured yet human readable text data. The author for the purpose of the project, followed the standard procedure of a Natural language processing algorithm. The author initially downloaded the data from a publicly available resource and performed the tokenization and text cleaning such as stopwords, and punctuations removal, and text vectorization and finally implemented the machine learning algorithm (Naïve Bayes classifier, in this project) to identify whether the SMS is spam or ham. The author, as shown initially created the Spark dataframe and observed the of data (shown in Fig. 4.3). During the exploration of the dataframe, the author also observed that the number of observations for the "Spam" label are lower than the "Ham" label. During the project, the author performed the following steps and arrived at the conclusion as follows.

- The author, in the project, chose to perform lemmatization of the text data over stemming because of the flexibility that lemmatization offers. The author opines that lemmatization is comparatively slower than the stemming, yet lemmatization is capable of providing better results since it works on the part-of-speech to output the dictionary form of the word which is more helpful to understand the context of the text data.
- The author for the vectorization step initially used the "CountVectorizer" of the sklearn library. The CountVectorizer transforms the given text data into vectors based on count of each word that occurs in the document. The author also performed the TF-IDF vectorization of the data, the TF-IDF vectorizer on the other hand focuses not only on the frequency of the words but also finds the importance of the words in the corpus.

- The author Naïve Bayes classifier on the data and found that the model accuracy was 98% when the text data was vectorized using the CountVectorizer. And the precision score of label “1” was 90% which has to be even more. However, when the author applied the TF-IDF vectorizer on the text data, the author was able to attain and accuracy of 97%, and the precision score of 100% for the label “1”. Which indicates that the model is able attain higher precision score when the data is vectorized using the TF-IDF vectorizer, as shown in the Fig 4.6 and 4.7 respectively.
- The author opines that the reasons for the model not performing well the CountVectorizer is that the CountVectorizer takes into account the count of the number of times the word appears in the corpus and tend ignore or given less importance to less frequently occurring words, and ultimately, resulting in algorithmic bias.
- The TF-IDF Vectorizer on the other hand, considered the overall weightage of the words by penalizing the most common words in the corpus and considering the importance of the words, which in turn is helpful in removing the less important words and make an efficient model which requires less computational time.
- Thus, the author would like to conclude that the Naïve Bayes classifier model is performing well with accuracy of 97% and the precision score of 100% for the identification of the intended target label “1”, when the text data is vectorized using the “Term Frequency-Inverse Document Frequency” (TF-IDF).

5.2 Recommendations

However, the author would also like to add that the test size is set at 0.2, which represents 20% of the total data. It has been observed by the author that if the number of samples in the data

Javaid, M., Haleem, A., Singh, R.P. and Suman, R., 2021. Significant applications of big data in Industry 4.0. Journal of Industrial Integration and Management, 6(04), pp.429-447.

Jeantet, I., Miklós, Z. and Gross-Amblard, D., 2020, April. Overlapping hierarchical clustering (OHC). In International Symposium on Intelligent Data Analysis (pp. 261-273). Springer, Cham.

Kapila, G., Nguyen, H.H. and Wang, H., 2020. Apache Spark Ecosystem for Big Data Analytics. MSDS SMU.

Kathrine, G.J.W., Praise, P.M., Rose, A.A. and Kalaivani, E.C., 2019, April. Variants of phishing attacks and their detection techniques. In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 255-259). IEEE.

Kaur, W., Balakrishnan, V. and Wong, K.S., 2022. IMPROVING MULTI-LABEL TEXT CLASSIFICATION USING WEIGHTED INFORMATION GAIN AND CO-TRAINED MULTINOMIAL NAÏVE BAYES CLASSIFIER. Malaysian Journal of Computer Science, 35(1), pp.21-36.

Khan, M.A., Karim, M. and Kim, Y., 2018. A two-stage big data analytics framework with real world applications using spark machine learning and long short-term memory network. Symmetry, 10(10), p.485.

Khan, N.A., Brohi, S.N. and Zaman, N., 2020. Ten deadly cyber security threats amid COVID-19 pandemic.

Kontsewaya, Y., Antonov, E. and Artamonov, A., 2021. Evaluating the effectiveness of machine learning methods for spam detection. Procedia Computer Science, 190, pp.479-486.

Kontsewaya, Y., Antonov, E. and Artamonov, A., 2021. Evaluating the effectiveness of machine learning methods for spam detection. Procedia Computer Science, 190, pp.479-486.

Kour, V.P. and Arora, S., 2019. Particle swarm optimization based support vector machine (P-SVM) for the segmentation and classification of plants. IEEE Access, 7, pp.29374-29385.

Lokaadinugroho, I., Girsang, A.S. and Burhanudin, B., 2021. Tableau Business Intelligence Using the 9 Steps of Kimball's Data Warehouse & Extract Transform Loading of the Pentaho Data Integration Process Approach in Higher Education. Engineering, MAthematics and Computer Science (EMACS) Journal, 3(1), pp.1-11.

Luu, H., 2018. Beginning Apache Spark 2: with resilient distributed datasets, Spark SQL, structured streaming and Spark machine learning library. Apress.

Madhavan, M.V., Pande, S., Umekar, P., Mahore, T. and Kalyankar, D., 2021. Comparative analysis of detection of email spam with the aid of machine learning approaches. In IOP conference series: materials science and engineering (Vol. 1022, No. 1, p. 012113). IOP Publishing.

Madhavan, M.V., Pande, S., Umekar, P., Mahore, T. and Kalyankar, D., 2021. Comparative analysis of detection of email spam with the aid of machine learning approaches. In IOP conference series: materials science and engineering (Vol. 1022, No. 1, p. 012113). IOP Publishing.

Makkar, A. and Kumar, N., 2020. An efficient deep learning-based scheme for web spam detection in IoT environment. Future Generation Computer Systems, 108, pp.467-487.

Mohammad, R.M.A., 2020. A lifelong spam emails classification model. Applied Computing and Informatics.

Naseem, U., Razzak, I. and Eklund, P.W., 2021. A survey of pre-processing techniques to improve short-text quality: a case study on hate speech detection on twitter. Multimedia Tools and Applications, 80(28), pp.35239-35266.

Naseem, U., Razzak, I. and Eklund, P.W., 2021. A survey of pre-processing techniques to improve short-text quality: a case study on hate speech detection on twitter. *Multimedia Tools and Applications*, *80*(28), pp.35239-35266.

Żelasko, P., Pappagari, R. and Dehak, N., 2021. What helps transformers recognize conversational structure? Importance of context, punctuation, and labels in dialog act recognition. *Transactions of the Association for Computational Linguistics*, *9*, pp.1179-1195.

Nayak, R., Jiwani, S.A. and Rajitha, B., 2021. Spam email detection using machine learning algorithm. Materials Today: Proceedings.

Nayak, R., Jiwani, S.A. and Rajitha, B., 2021. Spam email detection using machine learning algorithm. Materials Today: Proceedings.

Nguyen, T. and Meesad, P., 2021, December. A Study of Predicting the Sincerity of a Question Asked Using Machine Learning. In 2021 5th International Conference on Natural Language Processing and Information Retrieval (NLPIR) (pp. 129-134).

Nolte, J., Hanoch, Y., Wood, S.A. and Reyna, V.F., 2021. Compliance with mass marketing solicitation: The role of verbatim and gist processing. Brain and behavior, 11(11), p.e2391.

Orakzai, T., 2022. Botnet Propagation: An Analysis. Available at SSRN 4022576.

Packard, M., Stubbs, J., Drake, J. and Garcia, C., 2021. Real-World, Self-Hosted Kubernetes Experience. In Practice and Experience in Advanced Research Computing (pp. 1-5).

Pais, S., Cordeiro, J. and Jamil, M., 2022. HULTIG-C: NLP Corpus and Services in the Cloud.

Peng, F., Schuurmans, D. and Wang, S., 2004. Augmenting naive bayes classifiers with statistical language models. Information Retrieval, 7(3), pp.317-345.

Peng, F., Schuurmans, D. and Wang, S., 2004. Augmenting naive bayes classifiers with statistical language models. Information Retrieval, 7(3), pp.317-345.

Petersohn, D., Macke, S., Xin, D., Ma, W., Lee, D., Mo, X., Gonzalez, J.E., Hellerstein, J.M., Joseph, A.D. and Parameswaran, A., 2020. Towards scalable dataframe systems. arXiv preprint arXiv:2001.00888.

Pota, M., Ventura, M., Fujita, H. and Esposito, M., 2021. Multilingual evaluation of pre-processing for BERT-based sentiment analysis of tweets. Expert Systems with Applications, 181, p. 115119

Pullin, D.W., 2018. Cybersecurity: Positive changes through processes and team culture. Frontiers of Health Services Management, 35(1), pp.3-12.

Raharjana, I.K., Siahaan, D. and Fatichah, C., 2021. User stories and natural language processing: A systematic literature review. IEEE Access, 9, pp.53811-53826.

Ramsingh, J., 2022. PySpark toward Data Analytics. In Big Data Applications in Industry 4.0 (pp. 297-330). CRC Press.

Rathi, M. and Pareek, V., 2013. Spam mail detection through data mining-A comparative performance analysis. International Journal of Modern Education and Computer Science, 5(12), p.31.

Rauf, A.A., 2021. New moralities for new media? Assessing the role of social media in acts of terror and providing points of deliberation for business ethics. Journal of business ethics, 170(2), pp.229-251.

Raza, G.M., Butt, Z.S., Latif, S. and Wahid, A., 2021, May. Sentiment Analysis on COVID Tweets: An Experimental Analysis on the Impact of Count Vectorizer and TF-IDF on Sentiment Predictions using Deep Learning Models. In 2021 International Conference on Digital Futures and Transformative Technologies (ICoDT2) (pp. 1-6). IEEE.

Romli, I., Pardamean, T., Butsianto, S., Wiyatno, T.N. and bin Mohamad, E., 2021, March. Naive bayes algorithm implementation based on particle swarm optimization in analyzing the defect product. In Journal of Physics: Conference Series (Vol. 1845, No. 1, p. 012020). IOP Publishing.

Salahdine, F. and Kaabouch, N., 2019. Social engineering attacks: A survey. Future Internet, 11(4), p.89.

Sarica, S. and Luo, J., 2021. Stopwords in technical language processing. Plos one, 16(8), p.e0254937.

Savytska, L.V., Vnukova, N.M., Bezugla, I.V., Pyvovarov, V. and Sübay, M.T., 2021. Using Word2vec technique to determine semantic and morphologic similarity in embedded words of the Ukrainian language.

Shin, Y., Kim, K., Lee, J.J. and Lee, K., 2022. Focusing on the Weakest Link: A Similarity Analysis on Phishing Campaigns Based on the ATT&CK Matrix. Security and Communication Networks, 2022.

Singh, L.J. and Imphal, N.I.E.L.I.T., 2018. A survey on phishing and anti-phishing techniques. International Journal of Computer Science Trends and Technology (IJCST), 6(2), pp.62-68.

Singh, P., 2022. Manage Data with PySpark. In Machine Learning with PySpark (pp. 15-37). Apress, Berkeley, CA.

Sinthong, P. and Carey, M.J., 2019, December. Aframe: Extending dataframes for large-scale modern data analysis. In 2019 IEEE International Conference on Big Data (Big Data) (pp. 359-371). IEEE.

Spark, A., 2018. Apache spark. Retrieved January, 17(2018), p.1.

Subasi, A., Alzahrani, S., Aljuhani, A. and Aljedani, M., 2018, April. Comparison of decision tree algorithms for spam e-mail filtering. In 2018 1st International Conference on Computer Applications & Information Security (ICCAIS) (pp. 1-5). IEEE.

Subasi, A., Alzahrani, S., Aljuhani, A. and Aljedani, M., 2018, April. Comparison of decision tree algorithms for spam e-mail filtering. In 2018 1st International Conference on Computer Applications & Information Security (ICCAIS) (pp. 1-5). IEEE.

Suresh, P., Logeswaran, K., Keerthika, P., Devi, R.M., Sentamilselvan, K., Kamalam, G.K. and Muthukrishnan, H., 2022. Contemporary survey on effectiveness of machine and deep learning techniques for cyber security. In Machine Learning for Biometrics (pp. 177-200). Academic Press.

Tang, S., He, B., Yu, C., Li, Y. and Li, K., 2020. A survey on spark ecosystem: Big data processing infrastructure, machine learning, and applications. IEEE Transactions on Knowledge and Data Engineering.

Thangaraj, M. and Sivakami, M., 2018. Text classification techniques: a literature review. Interdisciplinary Journal of Information, Knowledge, and Management, 13, p.117.

Thangavel, M., Divyaprabha, M. and Abinaya, C., 2021. Threats and Vulnerabilities of Mobile Applications. In Research Anthology on Securing Mobile Technologies and Applications (pp. 560-580). IGI Global.

Torani, S., Majd, P.M., Maroufi, S.S., Dowlati, M. and Sheikhi, R.A., 2019. The importance of education on disasters and emergencies: A review article. Journal of education and health promotion, 8.

Varol, C. and Abdulhadi, H.M.T., 2018, December. Comparision of string matching algorithms on spam email detection. In 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT) (pp. 6-11). IEEE.

Vayansky, I. and Kumar, S., 2018. Phishing–challenges and solutions. Computer Fraud & Security, 2018(1), pp.15-20.

Williams, E.J., Hinds, J. and Joinson, A.N., 2018. Exploring susceptibility to phishing in the workplace. International Journal of Human-Computer Studies, 120, pp.1-13.

Williams, E.J., Hinds, J. and Joinson, A.N., 2018. Exploring susceptibility to phishing in the workplace. International Journal of Human-Computer Studies, 120, pp.1-13.

Yu, B. and Xu, Z.B., 2008. A comparative study for content-based dynamic spam classification using four machine learning algorithms. Knowledge-Based Systems, 21(4), pp.355-362.

Yu, L., 2018. Review of the Classification of Massive Chinese Texts Based on Spark. In MATEC Web of Conferences (Vol. 232, p. 01039). EDP Sciences.

Zaware, S., Patadiya, D., Gaikwad, A., Gulhane, S. and Thakare, A., 2021, June. Text summarization using tf-idf and textrank algorithm. In 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 1399-1407). IEEE.

Zečević, P., Slater, C.T., Jurić, M., Connolly, A.J., Lončarić, S., Bellm, E.C., Golkhou, V.Z. and Suberlak, K., 2019. Axs: A framework for fast astronomical data processing based on apache spark. The Astronomical Journal, 158(1), p.37.

Zheng, Z. and Sieber, R., 2022. Putting humans back in the loop of machine learning in Canadian smart cities. Transactions in GIS, 26(1), pp.8-24.