ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 7, Issue 7, July 2021

# Comprehensive Analysis of Spam Classification models using Machine Learning and Deep Learning

Srinishaa Prabhakaran, Sneha B S

Student, Computer Science Engineering,Dayananda Sagar University,Bangalore,India

Student, Computer Science Engineering,Dayananda Sagar University,Bangalore,India

***Abstract*—Spam emails have become a huge problem on the internet today. These unsolicited emails are sent by spammers across the world with different motives such as increasing the network traffic, phishing attacks, spreading viruses, and so on. The computer networks face adverse effects on the memory space of email servers, communication bandwidth, CPU power, and user time due to huge amounts of spam mails flooding the . This paper discusses a few of the most popular machine learning approaches such as Naive Bayes, Random Forest, Support Vector Machine and Deep learning algorithms like the Multiple Layer Perceptron, and the Recurrent Neural Networks that can be deployed to build a spam classification model and compares the working of these approaches using various performance measures. Apache's spam assassin dataset is taken and processed and it has been found that the SVM algorithm in the machine learning domain and the MLP in the Deep learning domain perform the best.**

***Keywords*—Spam, Ham, SVM, Random Forest, Naive Bayes, MLP, RNN, Pre-processing, Training, Testing, Accuracy metrics**

## I. INTRODUCTION

Emails and SMS modes of communication have become a very integral part of most people's everyday life. However, this has also resulted in the accretion of spam emails. Spam emails unnecessarily clutter the inbox which leads to wastage of storage, network bandwidth and is useless to the receiver. They can also be very distracting and may be a hindrance, making it very difficult to focus on the important mails. Humans learn to identify if a mail is a ham or spam based on their previous experiences with such mail, but can a machine be automated to do the same? This is where the role of spam classification systems comes into play. These spam detection techniques are very popular these days as they help in identifying these unwanted or sometimes even malicious mails from reaching the inbox.

This research paper documents a comprehensive analysis of various machine learning and deep learning techniques used for the classification of spam emails and provides a clear-cut comparison between all the models. The models used are the Naive Bayes, Random Forest, Support Vector Machine, Multiple Layer Perceptron, and the Recurrent Neural Networks.

## II. RELATED WORK

Due to the risk factor involved in the classification of spam emails manually, gradually many automated techniques have been accomplished to identify these emails. Many spam classification models using machine learning algorithms are designed for the purpose of checking whether a mail is spam or non-spam, and if it is spam, prevent them from reaching the inbox. Many research works have been taken up to find the best classification algorithm, but it is also dependent on the data fed. The five algorithms we have used are Naive Bayes, Random Forest, and Support Vector

Machine from Machine Learning and Multi-Layer Perceptron and RNN from Deep Learning. To differentiate the performances of these algorithms, many accuracy measures have been used and based on their performance, the algorithms are given grades. The performance metrics included are accuracy, recall, precision, and F-score. Spam Classifiers are modeled and evaluated on publicly available datasets. The datasets for all the models are taken from the assassin corpus site. In this paper, a thesaurus of some of the machine learning approaches to Spam filters is presented, where a quantitative analysis of the use of feature selection also known as variable selection algorithms was conducted on the dataset.

## III. METHODOLOGY

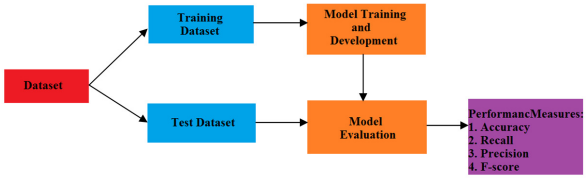



Fig.1 Model Schema

### A. *Pre Processing*

Dataset

The dataset is obtained from Apache's SpamAssassin public mail corpus.[6] It contains a collection of 5799 ham

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 7, Issue 7, July 2021

and spam messages with a 33% spam ratio suitable for training and testing spam classification models.

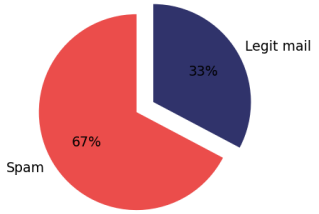



Fig.2 Pie Chart Spam-Legit mail

. Data cleaning

The dataset is checked for any missing or anomalous entry and is removed.

b. Natural Language Processing

The emails are deciphered to a suitable form that can be fed into the models using NLP.[7]

- The words in each email are converted to lower case.
- The email body is tokenized i.e. the sentences in the emails are split into smaller units such as words known as tokens.
- The stop words like 'as', 'the', 'is', 'a', etc. are eliminated.
- The HTML tags in the email are removed.
- The words are replaced by their root or base form, meaning the words are stemmed.
- The punctuations are removed.

c. Word Embedding

Some words often come in pairs, like good and bad or pros and cons. So the co-occurrence of words in a corpus can teach us something about their meaning. Word Embedding is the approach of finding this co-occurrence of words.

This approach represents words in a way that captures their meanings, semantic relationships, and the contexts they are used in.

Various word embedding techniques such as the CountVectorizer, TF-IDF, and Glove embedding are used for each of the models.

CountVectorizer - This technique converts each word into a vector depending on the frequency of each word in the dataset. It creates a matrix where each unique word is stored as a vector in the form of columns and the emails in the dataset are represented by rows. The value in each cell of a row is the frequency of each word in the email body.

TF-IDF - TF-IDF resizes the frequency of the common words based on, how frequently they appear in a given dataset so that the scores for words like "at", "in", that are frequent across all other mails in the dataset are penalized, for which it uses the concept of Term-frequency- Inverse document frequency, abbreviated as TF-IDF.[9]

Glove Embedding - Glove is a word embedding method. The approach behind it is that a certain word generally co-occurs more often with one word than another. The word hot is more likely to occur alongside the word fire for instance.[8]

B. *Training*

The data is trained using Machine Learning models SVM, Random Forest, and Naive Bayes and Deep Learning models such as MLP and RNN. An in-depth study of each model is performed in the further sections.

C. *Testing*

The confusion matrix is employed to classify the results, which consists of four parts: true positives, false positives, true negatives, and false negatives. Accuracy metrics such as Recall, Precision, and F-score are used for measuring the performance of the different models used for comparison.

## IV. PERFORMANCE MEASURES

*A. Accuracy :*

This tells how correctly the classification is performed. It is calculated by:

$$Accuracy = \frac{(True Positives + False Negatives)}{(True Positives + True Negatives + False Positives + False Negatives)}$$

*B. Recall Score :*

This is also known as sensitivity. It is a measure of the quantity of how many spam emails are identified vs how many emails were missed. It is calculated by:

$$RecallScore = \frac{True Positives}{True Positives + False Negatives}$$

*1. C. Precision Score :*

It gives the measure of how precise the classification is. It is given by using the formula:

$$PrecisionScore = \frac{True Positives}{True Positives + False Positives}$$

*D. F-Score :*

This is also known as F1-score. An inverse relationship exists between the precision score and the recall score, but to obtain maximum values for both the metrics we need to blend both of them to get a maximum score, that's where F-score comes into the picture. To obtain a balanced classifier we need to maximize the F-score. It is calculated by:

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 7, Issue 7, July 2021

$$F - Score = \frac{2 \cdot Precision \cdot RecallScore}{Precision + RecallScore}$$

## V. MACHINE LEARNING MODELS

A branch in artificial intelligence which utilises statistical models to learn from examples is Machine Learning. The examples are called data and the computer learns from data. Output is the number of categories called classes. Machine learning is about creating statistical programs called models and a lot of data is given to training those models. To develop a model it goes through a series of steps,i.e gathering data, pre-processing, exploring and visualizing, training, and testing the algorithm. This section deals with the comparison between the different models such as Naive Bayes, Random Forest, and Support Vector Machine.

### *A. Naive Bayes*

One of the simplest supervised algorithms is the Naive Bayes algorithm which is based on the Bayes theorem.[14] It includes probability concepts such as joint probability, conditional probability.

Naive Bayes classifier assumes each class of a particular feature is independent of other features.

The probability of each token or individual word is calculated by :

$$P(Spam|Token) = \frac{P(Token|Spam) \cdot P(Spam)}{P(Token)}$$

In this, probability of each token is calculated using the Bayes theorem and finally, the joint probability is applied to them.[2] If the outcome's probability of spam is more than non-spam then the mail is classified as spam or else non - spam. For training, the pre-built Naive Bayes classifier from the Scikit-learn is used to train the dataset. The component count vectorizer is used to generate the vocabulary, very quickly and efficiently. The final result is depicted through the confusion matrix.

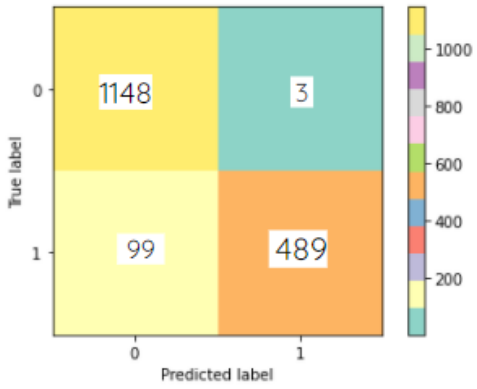


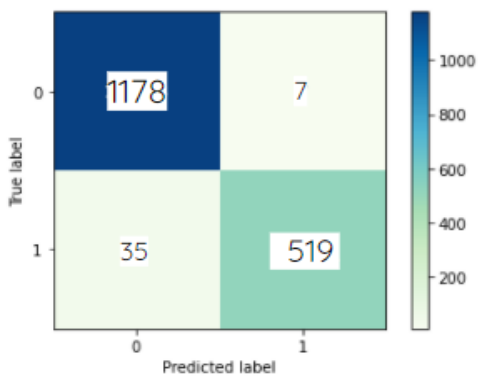

Fig.3 Naive Bayes Confusion matrix

### *B. Random Forest*

Random forest, a supervised learning algorithm, is used for both regression and classification. A forest is composed of trees, in the same way the random forest algorithm is composed of decision tree.[15]To create these decision trees, data samples are selected randomly and a prediction from each tree is obtained, and the best solution is selected through voting. An attribute selection indicator such as the Gini index is applied to each attribute to generate individual decision trees, where each tree depends on an individual random sample. In a classification problem, each tree votes, and the best solution is declared as the final result.

Model Description

The method used to generate the features, for training the model is TF-IDF vectorizer . For training, the pre-built Random Forest model from the Scikit-learn is used to train our dataset. The number of decision trees in the forest is the n_estimators, a parameter and is given a value of 32. The final result is depicted through the confusion matrix.



ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
**Available online at www.ijarmate.com**

IJARMATE

***International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)***
***Vol. 7, Issue 7, July 2021***

Fig.4 RF Confusion matrix

*C. Support Vector machine*

Support Vector Machine is very simple and highly accurate when compared to decision tree classifiers. It is best recognized for its kernel trick to manipulate the nonlinear input spaces. SVM classifier is also known as the discriminative classifier as the hyperplane is used to classify and separate the data points[1].To separate or classify the dataset into classes we use the concept of Maximum Marginal Hyperplane(MMH).[16]
SVM searches for the best classes and chooses the hyperplane with minimum classification error.
For non-linear planes, the SVM algorithm uses kernel trick to transform a lower dimensional input data space into the required higher dimensional data space, where it is transforming the non-separable problem to separable problems by adding more dimension to it. This helps to build accurate classifiers.

Model Description

The TF - IDF and Count vectorizer are used to generate the features employed for training the model. It has been observed that the accuracy is significantly higher when the TF-IDF approach is used as compared to Count Vectorisation. For training, the pre-built Support Vector Machine model from the Scikit-learn is used to train our dataset. The parameter, kernel which is used in the implementation is set to the sigmoid kernel.
The sigmoid kernel is given by the equation
$k(x,y) = \tanh(\gamma * x^T y + c)$, where $\gamma$ is the slope and c is intercept constant.[17]
The kernel coefficient for sigmoid is gamma and given set to the float value of 1.0
The final result is depicted through the confusion matrix.

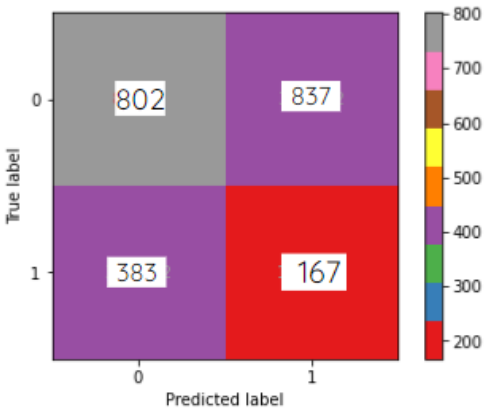



Fig. 5 SVM(CountVec) Confusion matrix

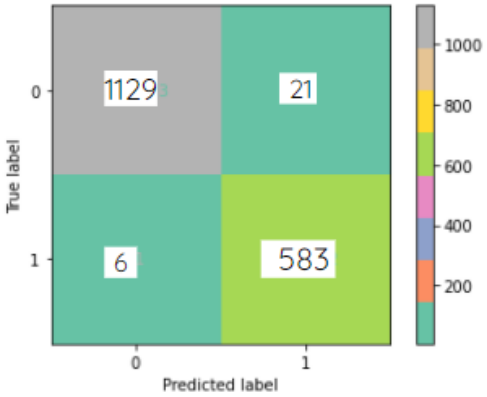



Fig. 6 SVM(TF-IDF) Confusion matrix

## VI. DEEP LEARNING MODELS

The birth of Deep Learning comes from the inquisitive thought of making computers think like humans and perform actions without having to explicitly specify what has to be done in a particular situation. This inspired the development of a model based on biological neurons. Inputs are taken by each neuron, weighed separately, summed up, and passed into a nonlinear function to produce output. Any machine-learning problem that requires classification can be solved using Neural Networks. Due to their efficacy, they are emerging as a major field within machine learning. What takes deep learning up another notch is its ability to detect input features from data ie. recognize patterns in input and make intelligent decisions on its own. This section demonstrates the pros and cons of using a deep learning model for spam classification.

*. Multilayer perceptron*

The Multilayer perceptron model is primarily composed of:
Input Unit: Here the input data is fed into the model.
Output Unit: Here the calculated output is channeled outside the network.
Hidden Unit: Here the data is accepted and signals are transmitted within the network with the help of activation function, weights, and bias. The input for each perceptron layer is the output of the previous perceptron layer.[4]

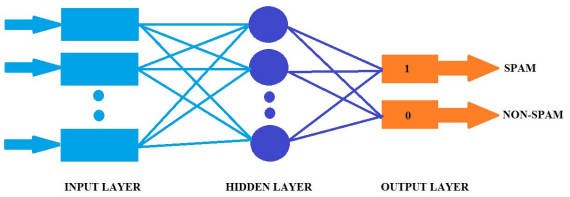



Fig.7 MLP Structure

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 7, Issue 7, July 2021

- Each neuron in a neural network will be activated based on a mathematical formula called the activation function.
- The activation function determines how strong the neuron will fire through trial and error.
- The neural network generates input features from input data. It can solve both linear and non-linear problems.
- The more the combination of input features, the deeper ie. the more the layers involved in the model, and more complex features are generated at each level.
- It makes a prediction and adjusts its parameters based on how far off the prediction was. It adjusts weights between neurons.
- The process through which error gets sent back down through the network so that each node can adjust its weight is called backpropagation.

Model Description

TF-IDF will be used to generate the features employed for training the model.

Algorithm:

- First layer L1 layer with Relu activation
- Second layer L2 layer with Relu activation
- Output Layer with Sigmoid activation
- Regularization: L1, L2 uses a dropout with probability 0.5 to prevent overfitting.
- Loss Function used is Binary cross-entropy
- Optimizer used is Adam Optimizer

The final result is depicted through the confusion matrix.

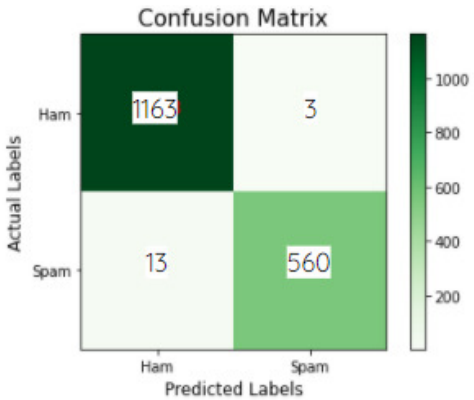



Fig.8 MLP Confusion matrix

*B. Recurrent Neural Networks*

In a multiple layer perceptron model, each one of the hidden layers has its own weights and bias. So each of these layers behaves independently. To combine the hidden layers, the same weights and biases are rolled together in a single recurrent layer. This forms the rudimentary Recurrent Neural Networks model.

A recurrent neuron combines current input with the stored state of previous input and hence maintains the relationship between the current input and previous inputs.

RNNs are mainly used for NLP sequence modeling problems i.e., where the sequence of input data is important. This approach may help in the classification of spam and ham messages as the relationship between the words in the sentences of each mail is taken into consideration, based on which the output is predicted. [11]

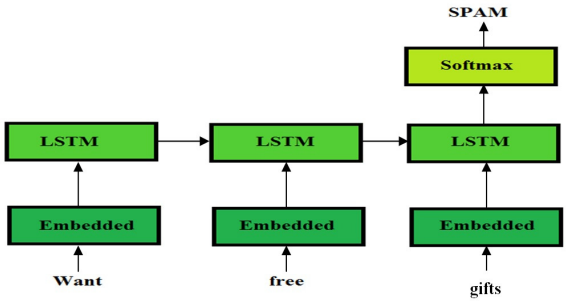



Fig.9 RNN Structure

- Single time step of input is supplied to the network. Here, 'time step' refers to each successive input that is fed to the model.
- The current input and previous input is combined and the current state is calculated.
- The number of time steps depends on the problem and in each step information is combined from the previous states.
- The current state that is obtained finally is used to calculate the output once time steps are completed.
- The error is generated after it is compared with the actual output.
- The weights are updated and the network is trained based on error that is back propagated through the network.
- The sum of errors at each time step gives the total error.
- One limitation is the vanishing gradient problem due to the short memory of RNN. To overcome this LSTM and GRU (specialized RNNs) are added to the network.

Model Description

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 7, Issue 7, July 2021

Glove embedding is used to generate the features for training the models.

Algorithm:

- Embedding layer using Glove embedding matrix
- LSTM layer with dropout=0.2
- 2 Fully-Connected output neurons with softmax activation
- Regularization: Dropout with a probability of 0.3
- Loss Function used is Categorical cross-entropy
- Optimizer used isAdam Optimizer

The final result is depicted through the confusion matrix.

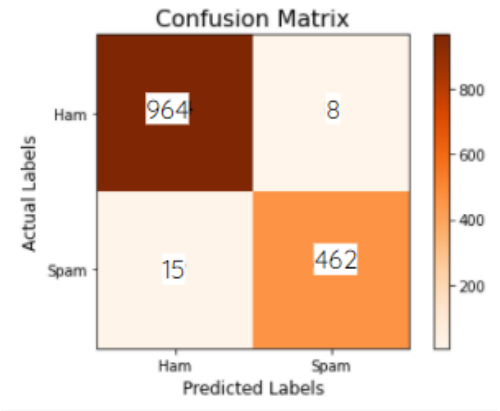



Fig.10 RNN Confusion matrix

## VII. Result Analysis

| Model | Precision | Recall | F1 | Samples | Training time |
|---|---|---|---|---|---|
| SVM(CountVec) | 56% | 57% | 57% | 1739 | 10.2s |
| SVM (TF-IDF) | 98% | 98% | 98% | 1739 | 5.91s |
| Random Forest | 97% | 97% | 97% | 1739 | 2.11s |
| Naive Bayes | 99.39 % | 83.17 % | 90.55 % | 1739 | 30□s |
| MLP | 99% | 98.73 % | 98.95 % | 1739 | 10.9s |
| RNN | 97.16 % | 99.55 % | 98.19 % | 1739 | 3min 52s |

### B. TABLE I: Results Table

The observations drawn from the results:

- SVM using TF-IDF performs the best among the Machine Learning algorithms[3], it is closely tied with the Random Forest algorithm.
- SVM using CountVectorizer shows a very low accuracy rate.
- Naive Bayes shows a precision score of 99% but a rather poor recall score showing it finds it difficult to classify the spam emails correctly but does a good job identifying the ham mails.
- Naive Bayes takes a significantly lesser amount of time as compared to other algorithms.
- In Deep Learning algorithms, MLP and RNN both show good results.
- RNN consumes the most time.

## VIII. Conclusion

The Deep Learning Models have the capability of generating the input features on their own, which saves a considerable amount of time and work. This may also be a disadvantage if there is a requirement to know exactly, on what basis the input features are selected, deep learning is like a “Black box” approach where a lot of computations happen inside the network and the features are magically spit out. Hence, this is not a tractable model. The MLP shows an impressive accuracy rate and is simple to construct as compared to RNN. RNN is rather superfluous for spam classification, it also consumes more time for training the model.

On the other hand, Machine Learning algorithms are computationally efficient and inexpensive as they do not require a high GPU. But the input features have to be generated manually to be fed into the model. Among all algorithms, Naive Bayes is considered the gold standard for spam classification because of its ease of implementation and its short training period, but this study shows SVM and Random Forest also perform well, in fact, better than the Naive Bayes Model.

### C. Acknowledgment

The authors would like to thank Dr.Srinivas, Dean of Dayananda Sagar University, Dr. Sanjay Chitnis, Chairman, Department of Computer Science, and Prof. Cauvery Raju for their intuitive ideas and insightful discussions with respect to the paper’s contribution.

### D. References

[1] R. Kishore Kumar, G. Poonkuzhali, P. Sudhakar, Member, IAENG.: Comparative Study on Email Spam Classifier using Data Mining Techniques In Proceedings of the International MultiConference of Engineers and Computer Scientists 2012 Vol I, IMECS 2012, March 14 - 16, 2012, Hong Kong

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 7, Issue 7, July 2021

[2] Emmanuel Gbenga Dada, Joseph Stephen Bassi, Haruna Chiroma, Shafi'i Muhammad Abdulhamid, Adebayo Olusola Adetunmbi, Opeyemi Emmanuel Ajibuw: Machine learning for email spam filtering: review, approaches and open research problems. In: 2019 The Authors. Published by Elsevier Ltd. It is an open-access article under the CC BY-NC-ND license

[3] Dima Suleimana, Ghazi Al-Naymat: SMS Spam Detection using H2O Framework In The 8th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2017)

[4] Ahmed, Kh: An overview of content-based spam filtering techniques. Informatica 31(3), 269– 277 (2007)

[5] Aakanksha Sharaff, Naresh Kumar Nagwani, and Abhishek Dhadse.: Comparative Study of Classification Algorithms for Spam Email In Springer India 2016 N.R. Shetty et al. (eds.), Emerging Research in Computing, Information, Communication, and Applications

[6] PublicCorpus Dataset https://spamassassin.apache.org/old/publiccorpus/

[7]https://www.dummies.com/programming/big-data/data-science/deep-learning-and-natural-language-processing/

[8]https://towardsdatascience.com/light-on-math-ml-intuitive-guide-to-understanding-glove-embeddings-b13b4f19c010

[9]https://medium.com/the-programmer/how-does-bag-of-words-tf-idf-works-in-deep-learning-d668d05d281b

[10]https://machinelearningmastery.com/how-to-configure-the-number-of-layers-and-nodes-in-a-neural-network/

[11]https://www.analyticsvidhya.com/blog/2017/12/introduction-to-recurrent-neural-networks/?utm_source=blog&utm_medium=cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning

[12]https://www.datacamp.com/community/tutorials/naive-bayes-scikit-learn

[13]https://www.datacamp.com/community/tutorials/random-forests-classifier-python?utm_source=adwords_ppc&utm_campaignid=1455363063&utm_adgroupid=65083631748&utm_device=c&utm_keyword=&utm_matchtype=b&utm_network=g&utm_adpostion=&utm_creative=332602034358&utm_targetid=aud-299261629574:dsa-429603003980&utm_loc_interest_ms=&utm_loc_physical_ms=9062014&gclid=Cj0KCQjwse-DBhC7ARIsAI8YcWIWN8oblnONeInh8LlHCf7Cb64evJDmSV2y0QD8Q6Hh8MkKi-J4ajwaApYTEALw_wcB

[14]https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python

[15]http://crsouza.com/2010/03/17/kernel-functions-for-machine-learning-applications/#:~:text=The%20Sigmoid%20Kernel%20comes%20from,%2Dlayer%2C%20perceptron%20neural%20network.