INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

www.ijprems.com
editor@ijprems.com

Vol. 03, Issue 04, April 2023, pp : 90- 99

e-ISSN : 2583-1062

Impact Factor : 5.725

EMAIL SPAM DETECTION USING GATED RECURRENT NEURAL NETWORK

S.Mani[1], Dr.G.Gunasekaran[2], Dr.S.Geetha[3]

[1]Final Year M.Tech CFIS, Department of Computer Science and Engineering, Dr.M.G.R Educational and Research Institute, Chennai 600 095, Tamilnadu, India.

[2]Professor, Department of Computer Science and Engineering, Dr.M.G.R Educational and Research Institute, Chennai 600 095, Tamilnadu, India.

[3]Head of department, Department of Computer Science and Engineering, Dr.M.G.R Educational and Research Institute, Chennai 600 095, Tamilnadu, India.

ABSTRACT

A rise in email triage as a result of a high volume of spam emails results in losses of USD 355 million annually. Sorting spam emails into categories like fraudulent or promotions from unwelcome parties is one approach to minimize this loss. Simple techniques, including word filters, served as the foundation for the early stages of the development of spam message classification. More sophisticated techniques are now being used, like machine learning-based language modeling. Networks using Recurrent Neural Units are among the most popular approaches to the text classification problem (GRU). GRU approaches were utilized because the purpose of this research is on the categorization of phishing emails. The findings of this investigation demonstrate that GRU achieved a high precision rate in the dropout-free situation. The constrained mailbox capacity is impacted by the sizeable volume of SPAM mail that is generated globally from numerous botnets. They have an impact on communication space loss as well as the safety of personal mail. They have an impact on the amount of time needed to recognize and respond to spam emails. Email spam identification is still regarded as a difficult task nowadays. This is due to the continued prevalence of email spam. It's because there is still much room for improvement in the identification. In order to detect spam emails, the author of this research creates a GRU-RNN. Using the Spam basis dataset, the new technique was evaluated. The method displays a 98.7% accuracy rate. The researcher comes to the conclusion that the suggested approach demonstrates an exceptional ability to recognize spam emails after completing extensive testing.

Keywords: Spam detection, Phishing messages, Email spam Recognition, GRU, RNN.

1. INTRODUCTION

Email spam, often known as electronic email spam, is the practice of sending malicious mails or commercial emails to a list of subscribers. Unsolicited emails signify that the receiver has not given consent to receive them. Throughout the last generation, using spam messages has grown in popularity. Spam has grown to be a significant online problem. Spam waste of space, time, and message delivery. Although automatic email filtration may be the best way to stop spam, modern spammers may quickly get around all of these apps. Prior to a few years ago, the majority of spam that came from particular email addresses could be manually stopped. For spam detection, an ML approach will be utilized. "Content analysis, white and bans of web addresses, and community-based procedures" are three major strategies that have been adopted closer to junk mail filtering. Text analysis of email content is a widely used spam prevention technique.

There are numerous solutions that can be used based on server and buyer considerations. Users and organizations would typically not require any important messages to get lost. The boycott strategy was most likely the first one used for the separation of spam. The strategy is to respond to every sender, excluding those from local or digital mail ids. This method no longer functions as well as it formerly did since more modern regions entered the category of spamming domain names. The white category strategy is the method of admitting emails from website addresses and names that have been publicly registered in the system while placing all other mails in a much lower-priority queue. This method works best when the sender responds to a confirmation request issued by a "junk mail filtration system".

2. REVIEW OF THE LITERATURE

This study evaluates the efficacy of two distinct aimless projection algorithms for resilient and effective spam identification when they are taught using a limited sample size. While effectiveness has to do with I the difficulty of the detection technique utilized; and (ii) the quantity of training material used for retraining and training again, robustness relates to learning and adaptation leading to a high degree of performance regardless of data unpredictability. Whereas the second approach, Random Boost, uses various feature selections to improve the efficiency of the Logic Boost technique, the first way, Random Project, uses a random transformation matrix to

construct linear consolidation of input values [1]. Since spam now frequently contains spyware and virus packages that might destroy the recipient's system, it has grown increasingly important to detect spam. Several machine learning-based spam detection methods have been proposed. Spam detection techniques should be used to address the issue because the volume of spam has increased significantly as a result of bulk mailing programs. Parameter optimization and extraction of features were suggested for spam identification to lower processing costs while ensuring an excellent rate of detection [2-3]. Spam is defined as unsolicited or unwanted communications received on mobile devices under an SMS. The cell phone customers are truly irritated by these Text spam messages. Service providers are also concerned about this marketing technique because it aggravates their customers or possibly makes them lose users. Researchers have put out a number of ideas for the recognition and filtration of SMS spam as a strategy to mitigate this activity. In this work, we evaluate the existing approaches, problems, and future perspectives in the fields of fake news detection, filtration, and reduction of mobile SMS spam [4-5]. The messages that hackers create and transmit to people via smartphones in an effort to obtain their sensitive information are known as SMS spam. For those who are illiterate, the hacker may obtain personal information if they comply with the message's instructions and enter sensitive data, including the username and password for their online banking account, into a phony website or application. Their money could be lost as a result. Effective spam detection is a crucial tool for assisting users in determining whether an SMS is a spam or not. In this article, we suggest a unique method for detecting SMS spam using Deep Convolutional and Natural Language Processing, based on an examination of English language spam messages [6-8]. Due to the localized content and frequent use of abbreviated phrases, SMS spam identification algorithms are more difficult than phishing mail detection methods. Unfortunately, none of the available research tackles these difficulties. Future study in this area has a vast amount of potential, and this survey can serve as a guide for its course [9-10]. This paper discusses the subject of SMS phishing detection and threads recognition and presents a cutting-edge clustering-based solution. Two phases of the work are planned. In the initial stage, a binary classifier method is used to divide Text messages between two groups, namely spamming and non-spamming messages.

Then, in the second phase, non-negative matrices factoring and K-means clustering methods are used to build Short messaging clusters for non-spamming text messages [11-12]. The amount of people who use mobile phones has increased, which has resulted in a sharp rise in SMS spam emails. Recent investigations have made it very evident that the amount of spam sent to smart phones is skyrocketing. Fighting such a scourge, in reality, is challenging due to a number of variables, such as the lower SMS usage rate that has enabled many consumers and network operators to ignore the problem and the scarcity of smart phone spam-filtering technology [13-14]. Being able to enter a field in its initial stages is one of the luckiest conditions a scientist may experience. There is a wide range of themes to choose from, and a lot of the problems go beyond simple technical ones. Regarding the field of supporting vector machines, we were granted the privilege of being in this role for the last seven years (SVMs) [15]. We use localized, dispensed, real-valued, and noisy pattern descriptions in our studies with synthetic data. LSTM produces far more good runs and trains considerably more quickly when compared to real-time recurrent lessons, back propagation through space, recurrent cascading correlation, Elman-nets, and neuronal sequence data compression. Also, LSTM resolves difficult, fake longer lead tasks that no existing recurrent network techniques have ever been able to [16-17]. In this work, we present a hybrid Nave Bayes classification and Apriori algorithm-based Text classification system to identify spam and ham. Despite the fact that this method is entirely reasoning based, the database's analytical nature will determine how well it performs. One of the most effective and important learning techniques for ML and data mining, Nave Bayes is additionally viewed as a fundamental method in information extraction [19-20].

3. PROBLEM STATEMENT

Spam is the unsolicited text that is delivered over the internet, especially to large numbers of online users, with the intention of advertising, soliciting business, communicating, or disseminating viruses. Nearly 85% of emails contain spam, which has increased significantly since 2007. The accuracy of existing approaches is displayed below. The number of spam emails has increased to 95% today. Statistics from Spam police show that 85,734,997 spam reports were made last year, with an average of 2.7 per second and a maximum of 4.7 per second. Getting rid of uncertain texts can often be difficult because they might take up a lot of room in your mailbox and need a lot of work to delete. These mails may contain viruses and malware, and fraudsters may steal the company's private information. It is crucial in this condition to install spam filters on the email servers of every organization. In the sense of NLP, text categorization is carried out for binary classification (for the ENRON dataset) and multi-class categorization (for the 20 Message board dataset), and is applied by deep learning methods that have achieved improved testing and verification accuracy than the conventional predictive methods.

www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

Vol. 03, Issue 04, April 2023, pp : 90- 99

e-ISSN : 2583-1062

Impact Factor : 5.725

4. PROPOSED METHODOLOGY

The researcher suggests a technique for identifying spam emails in this section. Depending on a detector that is an anomaly, the authors decided on the RNN-GRU as the core of the module. A trained model is loaded in the next module. It is given the statistics. It decides whether or not an email qualifies as an anomaly message. With regard to the proposed approach, it is evaluated using the Spam mails based dataset, which is taken from the UCI ML repository. There must be two phases—training and testing—for the neural network to successfully identify spam emails. The method used to identify spam and identify phishing scams using RNN is shown in Figure 1. The suggested models have three phases. Feature extraction and RNN-GRU with SVM are among these stages.

5. SYSTEM ARCHITECTURE

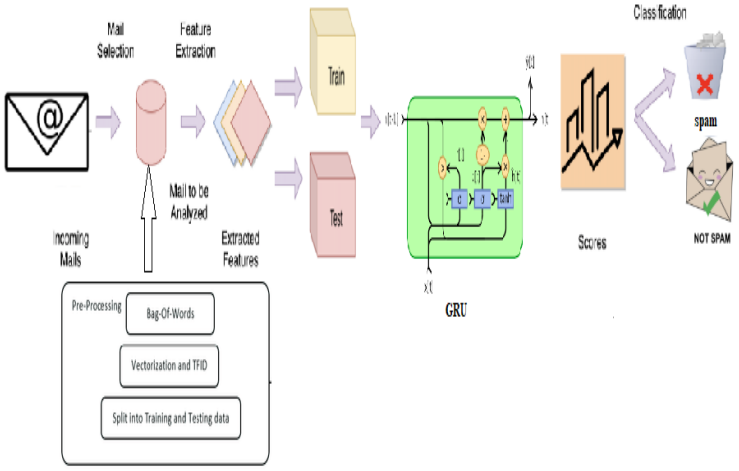


Figure 1. System Architecture

6. MODULE DESCRIPTION

In this system contains several modules, there are listed below,

- Dataset Preparation
- Data Preprocessing
- Tokenization
- Stop Word Removal
- Stemming
- Feature Extraction and Selection
- Modeling
- Performance

6.1 Dataset Preparation

ML models will be trained using the raw data that was gathered, which was acquired from the website Kaggle. We eventually acquired a total of a few emails, which were listed in 2 columns. Emails were to be categorized using the first column, titled "type," which had the potential values of spam or messages. The second column, under "e-mail Content," had several types of email content. Up to 80% of the mail will be utilized to prepare the models for use, and the leftover 20% will be utilized to test each model separately.

6.2 Data Preprocessing

ML models will be trained using the raw data that was gathered, which was acquired from the website Kaggle. We eventually acquired a total of a few emails, which were listed in 2 columns. Emails were to be categorized using the first column, titled "type," which had the potential values of spam or messages. The second column, under "e-mail Content," had several types of email content. Up to 80% of the mail will be utilized to prepare the models for use, and the leftover 20% will be utilized to test each model separately.

6.3 Tokenization

The process of turning the words of a phrase into indexes denoted by a number is called word tokenization. In this procedure, we generate a word tokenizer using a specified amount of interesting lexical terms. A word tokenizer is used to turn the words of a phrase into sequence information after being created. Tokenization converts words into indexes and sets the index to 0 for terms that are unknown. Additionally, we develop a tokenizer with a 10,000-word vocabulary limit. Additionally, we categorize text data by sequencing a tokenizer's index word count.

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
www.ijprems.com
editor@ijprems.com
Vol. 03, Issue 04, April 2023, pp : 90- 99
e-ISSN : 2583-1062
Impact Factor : 5.725

6.4 Stop Word Removal

Once the data has also been converted into distinct tokens, the following step is removing all superfluous words and punctuation, such as white spaces, commas, punctuation, colons, and semi-colons. The technique of removing pointless words is known as stop word deletion. NL Toolkit (NLTK), a built-in library for Python, is frequently used in language comprehension. Here, we employed the NLTK toolkit's stop phrases removal technique to get rid of any extraneous words and spaces.

6.5 Stemming

The next stage is to stem the token once they have been generated. Probably stem is the process of returning the derived terms from the data to their initial form. Then, all prefixes and suffixes are removed from the underlying phrase. The stemming process is then used to convert both modified and misspelled words to their basic or stem words. We used the NLTK Python Library to perfectly complete the stemming procedure for this phase as well. After mail content has been stemmed, spamming words can be quickly found.

6.6 Feature Extraction and Selection

A big raw dataset is transformed into an easier accessible format through the procedure of extracting features. Depending on the initial dataset, any variable, feature, or category can be extracted from the set of data throughout this step.

6.7 Modeling

In this experiment, using two algorithms for deep learning, including GRU methodologies, we create spam categorization models. By utilizing update gates and reset gates, the GRU is a deep learning method that is an improvement over the LSTM method in terms of reducing the complication of the system's structure. The updating gate acts to regulate how much of the concealed state is sent on to the following state. The importance of the prior hidden state data is determined by the reset gate.

6.8 Testing

To create models, we'll use open-source and free Keras neural network software. Since the Keras only accepts actual data as input, data must be vectored (converting characters into integers). To begin, we tokenize the data and used the 5000 most commonly occurring words in text reviews of movies. The block chain-enabled words are subsequently vectorized, as shown in the sample below.

6.9 Performance

By using the test dataset, we assess the models' effectiveness after training. The outcomes show that the model based LSTM and GRU have similar levels of accuracy.

7. WORKING

RNN is utilized in this work to process a sizable dataset and categorize emails as spam or not. A supervised machine learning method called RNN imitates short term memory. The frontal lobe of the brain is where short-term memory is housed. RNNs operate on the principle that they retain the knowledge they have acquired via prior observations. It then makes use of this information as it continues. In the instance of RNN, the hidden units not only create outputs but also provide feedback to the system. In order to process the data, the RNN makes use of poor memory.

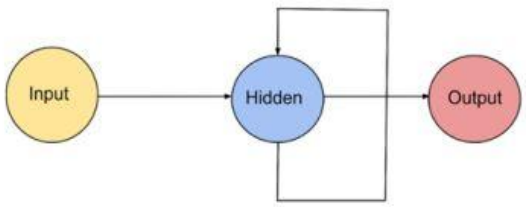


Figure 2. Working Methodology

Over time, the neurons become interconnected with one another. This illustrates the idea of having an STM, a type of memory. The neurons can recall what was already inside of them. The neuron gains ideas from earlier observations and transmits them to the following neurons. For instance, when an RNN is used to translate a phrase, the network must remember every phrase's translation as it moves forward in order to comprehend the sentence's meaning and produce an appropriate translation. RNN gains knowledge from the results produced by earlier neurons. Time step is the amount of time it takes for the output to change into the input. The amount of attributes constitutes the inputs. Before using the activation operation in an RNN, the input values and the prior output are determined. The input for the following layer is going to be this outcome. The sum of the epochs presents the total number of practice instance

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
www.ijprems.com
editor@ijprems.com
Vol. 03, Issue 04, April 2023, pp : 90- 99
e-ISSN : 2583-1062
Impact Factor : 5.725

iterations with one forward pass and one reverse pass. The hidden layer includes an assortment of neurons produced by employing the one. The dropout rate is another parameter that may be utilized to improve how well the NN performs. According to the data, this is because it can prevent the over fitting issue in NN and varies between the output.

7.1 K-Nearest Neighbor Algorithm

- The K-Nearest Neighbor is a simplest machine learning algorithm and it is also called a supervised learning algorithm.
- The K-NN algorithm implies that the new model and the previous cases are related, and it places the new instance in the area which looks most like the previous categories.
- The K-NN algorithm maintains all the information that is accessible and categorizes new data points according to similarity. This implies that by utilizing the K-NN method, fresh data can be quickly and accurately sorted into a suitable category.
- Although the K-NN approach is capable of solving regression and classification queries, classification challenges are where it is most frequently applied.
- Being a non-parametric approach, K-NN makes no assumptions about the underlying data.
- It is also known as a slow learning algorithm since it saves the training data rather than learning from it instantly. Instead, it uses the data to execute a task when classifying data.
- The KNN approach basically saves the dataset during the training process, and when it receives new data, it divides it into a category that is quite related to the new info.
- Consider the following scenario: We have a photograph of a species that resembles both cats and dogs, but we are unsure of its identity. However, since the KNN algorithm is based on a similarity metric, we can utilize it for this detection. Our KNN algorithm will look for similarities between the new data set's features and those in the photos of cats and dogs, and based on those similarities, it will classify the additional data set as either cat- or dog-related.

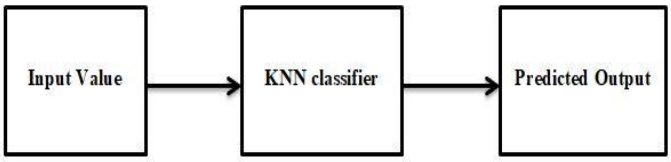


Figure 3. K-Nearest Neighbor Algorithm

7.2 KNN Working Methodology

In order to understand K-NN, let's look at the below details:

- **Step-1:** Choose the Kth neighbor from the list.
- **Step-2:** Determine the Euclidean distance between K neighbors.
- **Step-3:** Choose the Nearest k neighbors based on the Euclidean distance calculation.
- **Step-4:** Count the amount of data in every category among such k neighbors.
- **Step-5:** Put the additional data points towards the area where the neighbor count is at its highest.
- **Step-6:** The model is complete.

Let's assume we need to classify new data in order to use it. Think about the below graph:

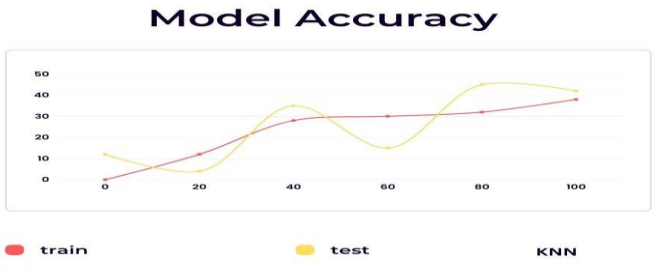


Figure 4. Model Accuracy

Calculating the percentage of the right forecasts is the most logical way to assess the success of any classification model. And even the Accuracy score was determined using precisely this reasoning. The Accuracy range (or simply Accuracy) in ML is a classification parameter that includes a percentage of the forecasts that a model correctly predicted. The metric is widely used since it is simple to calculate and understand. Additionally, it uses a single value to assess the model's accuracy. So, in order to assess a Classification algorithm utilizing the Accuracy range, you must have:

- The ground truth classes;
- And the model's predictions

7.3 Accuracy Score Formula

Moreover, accuracy is a relatively intuitive parameter, so understanding it shouldn't be difficult for anyone. By dividing the entire amount of forecasts by the number of right predictions, the accuracy rate is determined.

Accuracy = Number of correct Predictions/Total Number of Predictions

The more formula is the following one.

Accuracy = (True Negatives + True Positive)/TP + FP + TN + FN

It is evident that concepts from the Confusion matrix, such as True Positive, True Negative, False Positive, and False Negative, are useful for describing accuracy. However, as stated on the Confusion matrix site, the majority of the time, these words is deployed for binary classifier jobs.

Thus, the following is the accuracy rating method for the binary categorization assignment:

- Get forecasts from your model;
- The total number of True Positives, True Negatives, False Positives, and False Negatives should be determined.
- Employ the binary case accuracy equation;
- Assess the value that was obtained.
- Yep, it really is that easy. How about the multiclass scenario, though? As there is no explicit formula, we advise utilizing the metric's fundamental logic to calculate the outcome. The following describes the multi-classification task's accuracy rate algorithm:
- Get forecasts from your model;
- Determine the percentage of accurate forecasts;
- Split it by the overall amount of predictions;
- And evaluate the value you just got.

7.4 Gated Recurrent Unit (GRU)

The GRU neural network is a unique variation of the recurrent neural network, or RNN, which has found widespread use in industries and can maintain a piece of longer-term data dependent. GRU still suffers from the drawbacks of slow convergence and ineffective learning, though. Hence, we suggested neural network architecture with an optimized gated recurrent unit (OGRU). The OGRU model improves the learning strategy of GRU through the application of the reset gate, resulting in enhanced learner performance and prediction performance. As likened to lengthy short-term memory, the Gated Recurrent Unit (GRU) is a form of recurrent neural network (RNN) (LSTM). GRU is rapid and utilizes less memory than LSTM, however LSTM is more efficient when working with datasets that include longer sequences. The vanishing gradient problem, which concerns standard recurrent neural networks, is also resolved by GRUs (information used it to update network weights). Grading may become too little to have an impact on learning if it reduces with time as it back propagates, rendering the neural network inefficient. RNNs can essentially "forget" longer sequences if a layer in a neural net fails to learn. The updated gate and reset gate are two gates that GRUs uses to address this issue. Such gates can be trained to retain the information from far back and decide what data is accepted to make it to the output. As a result, it can transfer important information along in an event chain to enhance its predictions.

According to the GRU network's essentially sequentially training phase, parallelizing GRU networks is a difficult task. Previous efforts to parallelize GRU mainly focused on the use of traditional parallelization techniques like information concurrent and model-parallel training algorithms. Existing approaches are still unavoidable performance constrained in regard to instruction duration whenever the provided sequences are extremely long, though. In this research, we present a unique parallel training technique for GRU using a multi-grid reduction in time (MGRIT) solution. A sequence is split up into numerous smaller sub-sequences by MGRIT, which trains the sub-sequences concurrently on different processors. The key to speeding it up is a hierarchical change of the hidden layer to promote

end-to-end communication also during the forward and reverse propagation phases of gradient descent. The dataset's experimental results, where each video is an image sequence, show that the new parallel training methodology speeds up training by up to 6.5 times faster than a sequential method. Our unique parallel GRU algorithm significantly improves in efficiency as the sequence increases since the effectiveness of our unique parallel processing method is correlated with sequence length.

Repeated Unit with such a Gate Even though the disappearing gradients issue has been addressed by LSTM, Cho et al. presented the Gated Recurrent Unit (GRU), a generalized version of LSTM, in 2014. The GRU has gated units that influence the information that moves within the unit, much like the Lstm does, but it lacks separate memory cells. Each hidden unit's flow of information is controlled by two gates known as updates and resetting gates, that are calculated by a gated recurrent unit (GRU). The following formulas are employed to calculate each hidden layer at time-step t:

Update gate: $z_t = \sigma (W_z \cdot [h_{t-1}, x_t])$

Reset gate: $r_t = \sigma (W_r \cdot [h_{t-1}, x_t])$

New memory: $\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$

Final memory: $h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$

Where W denotes weight vector, $*$ denotes element wise multiplication and σ is the sigmoid function.

The process shows the GRU's internal environment at time-step t.

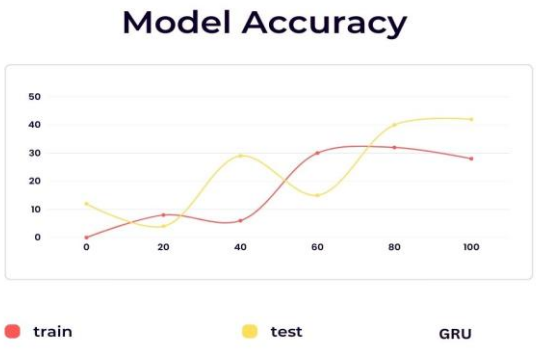
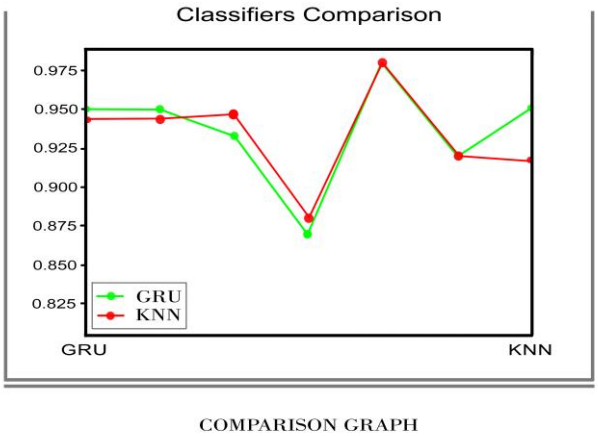


Figure 5. Model Accuracy

7.5 KNN vs. GRU Algorithm



COMPARISON GRAPH

Figure 6 Classifiers Comparison

The above mentioned comparison graph explains about the difference and accuracy ranges between the K- Nearest Neighbor Algorithm and Gated Recurrent Unit. We firstly used The KNN algorithm for the study of Email spam detection and it gives the accuracy rate almost 90 percentage. In this study we used pre-trained datasets that is collected from kaggle. The data set contains thousands of images or data's that based spam messages and spam mails. We used Jupiter note book that is used to train the data sets.

The K-Nearest Neighbor algorithm is a Machine Learning algorithm that contains following steps to implement the algorithm in application.

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

www.ijprems.com
editor@ijprems.com

Vol. 03, Issue 04, April 2023, pp : 90- 99

Impact Factor : 5.725

- Firstly, load the data.
- Select the K value.
- For every data points in the data
- Discover the Euclidean distance to all training data samples
- Save the distances on an ordered list and align the data samples.
- Select the top K entries from the data list
- Mark the test point based on the majority of areas presented in the selected points

End

To verify the accuracy of the KNN categorization, in this study we used confusion matrix in KNN algorithm and used other statistical methods like the likelihood ratio test. In the study of KNN regression, the primary steps are repetitive, rather than allocating the class with the greater votes, the average of the K value is estimated and allocated to the undetermined data point. But the KNN algorithm has some disadvantages, there are

- Storage space requirements are high
- It is necessary to determine K's value
- In the presence of high N, prediction is slow
- Observes irrelevant features with great sensitivity

In the case of KNN disadvantages we used Gated Recurrent Unit (GRU). The GRU algorithm is provided more than 90 percentage of accuracy in this project. In this study we used pre-trained datasets that is collected from kaggle. The data set contains thousands of images or data's that based spam messages and spam mails. We used Jupiter note book that is used to train the data sets.

The structure of the GRU is simpler than KNN algorithm it contains following advantages

- The training time of GRU algorithm is shorter
- It requires less data points to capture the properties of the data

8. EXPERIMENTAL RESULTS

8.1 Dataset

The author used the spam-based dataset, which contains 57 data variables relating to the frequency of specific terms in the text of the email, to analyze and quantify the abnormalities. The dataset consists of 4601 mails and 57 attributes, of which 1813 (or 39%) are tagged as spam emails. The non-spam text must contain 2788 (61%).\

8.2 Experimental Setup

In this study, we build a neural network representation that primarily uses GRU utilizing deep learning KERAS architecture with ADAM optimizing operation. Tensor flow is used as the backend of Keras. Furthermore, a Boolean cross-entropy is used to calculate the cost function. The laptop used for this study's investigations included an Intel Core(TM) i7-3632QM processor running at 2.20GHz, 8GB of memory, and an NVIDIA GeForce GTX 4GB GPU.

8.3 Performance evaluation and results

The TPR, TNR, FPR, median validation accuracy rate, and F-Score were estimated for the chosen datasets. According to the findings, which are displayed in Table, the deep RNN exhibits the technique's maximum precision and recognition rate at roughly 98.65%. Also, the proposed method has the greatest F- score rates, AUC, and TPR, which are approximately 97.93%, 98.61%, and 98.65%, correspondingly. In this below mentioned figure 7 shows that the mail sent to the another user in text format through this application

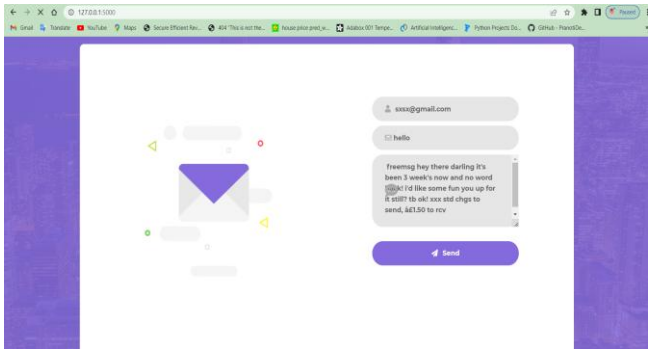


Figure 7. Performance evaluation

After sent the mail to the user the mail shows this is the spam mail or not, and also shows is contain any bullying content or not. This detail explained in figure 8.

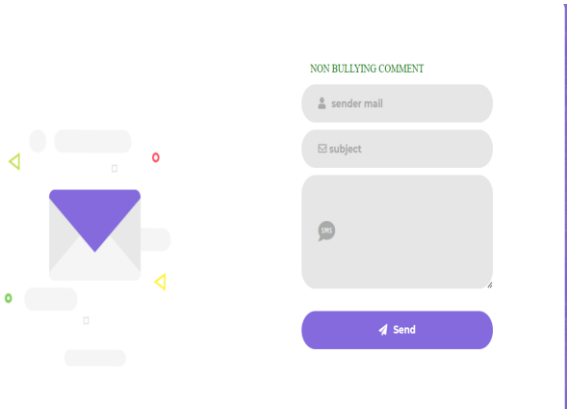


Figure 8. Performance evaluation

With this outcome, it is clear that the GRU-RNN produces the greatest results, but it is constrained by class-conditional freedom, which causes the machine to mistakenly categorize some tuples. On the opposite hand, ensemble approaches have been shown to be effective because they combine several classifications to predict classes.

9. CONCLUSIONS

Nowadays, a large number of messages are being sent and received, which makes it challenging because our project can only analyze mails using a small corpus. Our solution, which enables spam detection, is capable of filtering emails based on the mail's content rather than the web addresses or any other factors.

As a result, the mail's body is now somewhat brief. Our project has many opportunities for improvement. The following enhancements are possible: On the basis of reputable and validated web addresses, spam can be filtered. "The fake email classification is particularly crucial in identifying spam from non-spam emails and in categorizing emails." "The big body can use this strategy to distinguish acceptable emails from emails they just want to receive.

10. REFERENCES

[1] D. DeBar and H. Wechslar, "Spam identification applying random improvement," Pattern Classification Letters, vol. 33, no. 10, pp. 1237-1244, 2012.

[2] S. M. Lee, D. S. Kim, J. H. Kim, and J. S. Park, "Spam recognition using feature extraction and parameters utilization," in 2010 Global Meeting on Complex, Smart and s/w In-depth Systems, 2010: IEEE, pp. 883-888.

[3] A. Narayanan and P. Saxina, “The curse of 140 characters: Calculating the efficiency of SMS spam recognition on android,” in Proc. Third ACM workshop on Safety and Privacy in Mobile phones & Devices, 2013, pp. 33–42.

[4] M. A. Shafi’I, M. S. A. Latief, H. Chroma , et al., “A study on mobile SMS text spam reduction methods,” IEEE Access, vol. 5, pp. 15650-15666, 2017.

[5] S. Bhatia, “A study on spam recognition techniques for security SMS communication,” Global Article of Engineering & Technology, vol. 7, pp. 790–792, 2018.

[6] P. Poomika, W. Pongsana, N. Kerdprasoop, and K. Kerdprasoop, “SMS spam detection based on long shortterm memory and gated recurrent unit,” Global Article of Future CS and Communication vol. 8, no.1, pp. 11-15, 2019.

[7] E. G. Dada, J. S. Basia, H. Chromab, S. M. Abdullhamidc, A. O. Adetunimbid, and O. E. Ajibuwa, “ML for mail spam reduction: Study, methodologies and open research problems,” 2019.

[8] K. Yaadav, S. K. Sahaa, P. Kumaraguru , and R. Kumara, “Keep control of your SMS or Text: Developing an runnable SMS spam reduction system,” in Proc. IEEE 13th Global Meeting on Mobile Data Management, 2012, pp. 352– 355.

[9] L. N. Lotas and B. M. M. Hosain, “A systematic literature review on SMS spam detection techniques,” Global Article of IT and CS, vol. 9, no. 7, pp. 42-50, 2017.

[10] T. Islam, S. Latif, and N. Ahmed, “Utilizing social media networks to identify malicious bangala text content,” in Proc. 1st Global Meeting on Advances in Science, Engineering and Robotics Technology, May 2019, pp. 1-4.

[11] Q. Xu, E. W. Xiang, and Q. Yang, "SMS Mail spam Recognition utilizing non-content features," IEEE Smart Systems, vol. 27, no. 6, pp. 44-51, 2012.

[12] F. Olsson, "A literature review of responsive ML in the context of natural language processing," Swedish Institute of CS, 2009.

[13] T. Almeda, J. M. G. Hidalgeo, and T. P. Silvia, "Towards SMS Mail spam reduction: Outputs under a new data cluster," Global Article of ISS, vol. 2, no. 1, pp. 1-18, 2013.

[14] Pedregosa, Fabian, et al., "Scikit-learn: ML in Python," Article of ML Research, vol. 12, pp. 2825-2830, 2011.

[15] B. Scholcopf and A. J. Smola , Learning with Kernels: SVM, Regularization, Escalation, and Beyond, MIT Press, 2001.

[16] R. Pascan, T. Mikoluv, and Y. Bengiou, "On the difficulty of tutoring recurrent neural networks," in Proc. Global Meeting on ML, February 2013, pp. 1310-1318.

[17] M. T. Nuruz zaman and C. Lee "Individual and Private Text Spam Reduction," presented at 11th IEEE Global Meeting on CS and IT, 2011.

[18] S. Hosch Reitar and J. Scihmidhubeer, "Long and short-term memory," Neural Calculation, vol. 9, no. 8, pp. 1735- 1780, 1997.

[19] I. Ahamed, D. Guan , and T. C. Chung, "SMS classification based on navïe bayes classifier and apriori methodology frequent data set," Global Article of ML and CS, vol. 4, no. 2, p. 183, 2014.

[20] T. Mikoluv, M. Karafiáet, L. Burget , J. Černoký, and S. Kudanpur, "Recurent neural network based language model ," in Proc. 11th Annual Meeting of the Global Voice Transmission Union, 2010