Spam Detection of SMS Messages Using Random Forest Classifier Algorithm

K Ranjith Reddy[1], Dr Ganpat Joshi[2]
[1]Research Scholar,
Dept of Computer Science and Engineering, Madhav University,
Abu Road, Sirohi, Rajasthan 307026, India
[2]Professor,
Dept of Computer Science and Engineering, Madhav University,
Abu Road, Sirohi, Rajasthan 307026, India

Abstract-: As the popularity of cell phones has increased in recent years, SMS i.e., short message service has arisen as a multi-billion-dollar industry. Reduced messaging costs have led to an increase in unsolicited (spam) mobile ads. In a study in 2012, it is found that a total of 31% of text sent was spam in some Asian countries. The email filtering algorithms may not achieve the quality they are supposed to given the results. In this paper, we use efficient random forest algorithm for the classification of the SMS spam database from the Machine Learning UCI repository is taken, which contains around 5572 samples. After preprocessing, we create the embeddings for the dataset, we then pass in the vectors to our random forest algorithm for the classification task. The results are given considering the data imbalance problem and achieving an accuracy of about 96%. The experiment results aim to differentiate between spam and ham messages by creating a sensitive and efficient classification model that provides good accuracy with fewer false positives. And finally, we have concluded our experiment with high accuracy with the Random Forest classifier.

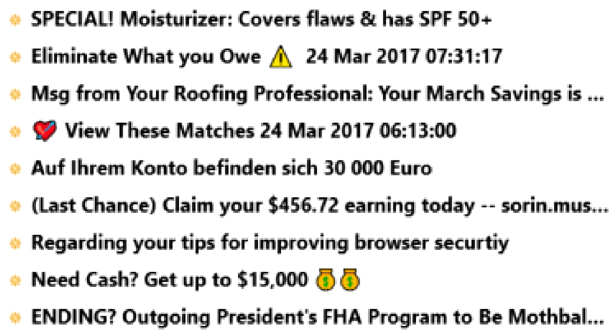
Keywords: Random Forest Classifier, SMS Spam, Spam Filtering, Natural Language Processing, Text Mining

1. Introduction:

The market for mobile phones has grown significantly in recent years [7, 8]. In the second quarter of 2013, over 432 million cell phones were supplied, which is a 6% increase year over year. Short Message Service, or SMS, is developing into a commercial sector with a multi-billion-dollar market value as cell phone use has become more widespread. Cell phone users can exchange brief messages via SMS, a messenger-like platform, often limited to 160 7-bit characters. With around 3 billion active users in 2010, or almost 80% of all cell phone subscribers, it was the most widely used data application.

As the platform's growing in popularity, the increase in the number of unsolicited notifications sent to cell phones via SMS can be seen [3,4]. SMS spam is even rarer than email spam, with about 90% of emails being spammed in 2010, and still not large in North America. In China, text messages can cost less than $0.001 per message for adults. In addition, in some Asian parts, up to thirty percent of text messages were sent. In the Middle East, there are telephone companies that send marketing text messages. In addition, especially SMS spam [5,9] is more exciting than spam in the email as in some countries they also add to the cost of the recipient. we would like to consider the problem of finding spam, especially for text messages, due to these factors and the limited availability of mobile spam software.

In spam classification [1,2], the spam emails are identified using the spam filters and prevent those emails from going to the mailbox. These filtering techniques are used to overcome form the negative effects of spamming which serve as a reliable predictable tool for eliminating unwanted emails. However, there is little risk of legitimate emails being sorted or removed incorrectly. Below picture clearly shows examples of spam messages:



- SPECIAL! Moisturizer: Covers flaws & has SPF 50+
- Eliminate What you Owe ⚠ 24 Mar 2017 07:31:17
- Msg from Your Roofing Professional: Your March Savings is ...
- 💔 View These Matches 24 Mar 2017 06:13:00
- Auf Ihrem Konto befinden sich 30 000 Euro
- (Last Chance) Claim your $456.72 earning today -- sorin.mus...
- Regarding your tips for improving browser securtiy
- Need Cash? Get up to $15,000 💰💰
- ENDING? Outgoing President's FHA Program to Be Mothbal...

Fig. 1 Example of Spam messages

There is a big difference between text and email spam. In contrast to email, where a large database is available, a real SMS spam database is very limited [3,4]. Because of the text messages' short length, the quantity of features that can be used to classify them is much lesser than the corresponding number in an email. In addition, text messages are full of acronyms and much less formal language than you would expect from an email. All these factors can significantly reduce the performance of the main spam filtering algorithms used on short text messages. This paper focuses on applying several mechanical algorithms [10,11] to spam sorting problems and comparing their results in order to gain insight and further investigate the problem. It uses a database made up of 5572 messages that were taken from the UCI Machine Learning repository [13,14]. We have a collection of 747 SMS spam messages that have been manually taken from the UK forum Grumble text, where users of mobile phones can request SMS spam in public. A sizable text file makes up the dataset. The message tag and text message string are the first two characters on each line. The random forest algorithm is used once the data is provided for property analysis and extraction.

2. Literature Review and methodologies

Several papers [9-13] were studied for understanding the existing research work and technologies used for performing the task of Machine Learning based Spam Detection using various techniques. Some of the noteworthy papers are mentioned here. In a paper, Gomez Hidalgo et al. [2] have used some classifiers-based Bayesian techniques to detect the mobile scam. The authors have proposed two famous SMS spam datasets and have tested a few machine learning approaches and some methods based on message portrayal on these English and Spanish datasets. Finally, they concluded that for the SMS spam classification, the Bayesian filter can easily be adopted.

In another paper, Cormack et al. suggested that for short text messages, some spam filtering based on content can be used. These text messages broadly happen in three different perspectives: blob comments, Short Message Services, and email summary information [3]. The ending of their work was that SMS is limited to fewer words in order to adequately support words or spam classifiers based on bigrams of words, and thus, expanding the feature set to bigrams of words. sparse orthogonal words and character trigrams and bigrams, the efficiency was increased. The popularity of Bayesian methods is growing and their use in spam and text classification applications is enhancing. It offers advantages in price-sensitive assessment because it’s able to provide a huge level of classification confidence. For research Sahami et al. [1], a Naive Bayes classifier and a word representation Houshmand Shirani-Mehr applied various algorithms on SMS spam problems and used these different algorithms to compare the overall performance to further investigate and to gain insight about the problem, and designed a program based on one of the used machine learning algorithms that can detect SMS spam filtering with great performance and accuracy[6]. Database with 5574 text messages was used by them bag were used for the email dataset. This article shows an enhanced performance level which was demonstrated by adding advanced features and a few non - alphabetical features to the featured bag.

The functionality of filtering messages on standalone cell phones using text sorting methods[4]. Independent mobile phone processing was carried out, which involved training, screening, and updating. Their well-known results depict that the predictive model was capable of distillation between ham and spam messages with moderate efficiency, consumed less memory, and spent considerable time operating without machine assistance. In another paper, on an SMS corpus [5], Sarah Jane Delany worked on an experiment based on clustering. For accessing SMS spam behavior, they gathered 1353 spam messages and used them as a data package that did not understand duplication. A k-way spectral cluster with orthogonal starters was applied. By using spectrum clusters to their own composite database, some clusters were obtained, with ten of them with the best 8 terms and proposed explanations.

Houshmand Shirani-Mehr applied various algorithms on SMS spam problems and used these different algorithms to compare the overall performance to further investigate and to gain insight about the problem, and designed a program based on one of the used machine learning algorithms that can detect SMS spam filtering with great performance and accuracy [6]. Database with 5574 text messages was used by them.

2.2 Methodologies

i) Random Forest

Random forests are ensemble learning-based algorithms that can be used to address data classification issues, as discovered by Akinyelu and Adew. Breiman and Cutler proposed the RFs algorithm in 2007 [1,15]. Using decision trees, the algorithm divides the data into several classes. A few decision trees are created during the training stage, and these trees are later used for categorization. This is made possible by taking into account the class that each individual tree has elected, with the class receiving the most votes being the final outcome. The RF algorithm has gained a lot of notoriety over time and is now used to address related issues in a variety of human endeavors. In comparison to other machine learning methods, random forests have a number of benefits, including high f scores and low classification error. Additionally, they typically perform just as well as or even better than SVMs. Unbalanced records with missing values can be dealt with by it successfully. It functions as an effective technique for estimating the value of missing data and maintaining data accuracy in the presence of vast amounts of data. RFs typically require less training time than SVMs and neural networks, though individual implementation may affect this. The ability to perform RF is superior to the accuracy of other machine learning algorithms. It has a very good performance in large databases.
It can efficiently handle a wide variety of input variables. Random forest (RF) produces an internal unbiased prediction of collective error during forestry. Provides an approach to reducing errors in population classes with skewed records. Random forests are simple and use fewer parameters compared to the number of observations. The steps for cultivating trees are described in the outlined text:

1. Suppose, T, be the number of training instances, randomly representing T instances that can be substituted from the existing data. These situations are used as a guide to growing the tree.
2. Suppose that if Y input variables are taken into consideration, then for each of the nodes, a value x << Y is selected corresponding to them, x variables are randomly selected from Y and to partition the node, the finest portion on x is used and thus, now x have a fixed value all through the period of growing the forest.

ii) AdaBoost

Adaboost is an ensemble method that takes sequential classifications that have been modified in favor of cases of misclassification by previous classifiers. The classifiers it uses may be as weak as a little better than a random guess, but it will still improve the final model. This method can be used in combination with other methods to enhance the final ensemble model. In each version of Adaboost, certain weights are applied to the training examples. These weights were evenly distributed before the first version. Then, after each iteration, the weights for your multi-ranked tags increase according to the current model, and the weights for properly sorted samples decrease. This means that the new prediction focuses on the weaknesses of the previous classifier. Like random forests, while the complexity is much higher, Naive Bayes' algorithm is still beating Adaboost with better performance of decision tree.

iii) K-Nearest Neighbors

K-NN is basically a distance calculation-based simple machine learning classifier. The
new data point is classified by identifying the k-nearest neighbors based on the
neighbor’s class count.

2.3 Evaluation parameters

i). Precision: Precision is the ratio of system-generated results that are correctly predicted positive (truly positive) observations to the total number of predicted positive system observations, both true (true positives) and false (false positives).

ii). **Recall:** Recall is the fraction of the model-generated results that correctly predicted the positive (truly positive) observations compared to all true (truly positive) malignancy class observations.

$$\Pr ecision = \frac{True\ Positives}{True\ Positives + False\, Positives}$$

$$\mathrm{Re}\, call = \frac{True\ Positives}{True\, Positives\ +\ False\, Negatives}$$

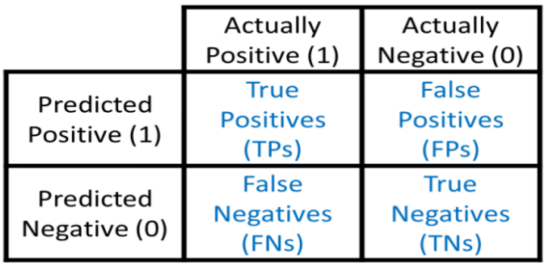
iii). F-Value: The F score is a measure of the precision of a test. Both accuracy and memory are taken into account when scoring.

$$F\ Score = \frac{2 \times (\Pr ecision \times \mathrm{Re}\, call)}{\Pr ecision + \mathrm{Re}\, call}$$

iv). Accuracy: Accuracy is the ratio of the total number of predictions that were correct (both true positives and true negatives) to that of the total number of samples.

$$Accuracy = \frac{(TP\ +\ TN)}{(TP + FP + TN + FN)}$$

v). Confusion Matrix: A confusion matrix is a way of presenting the summary performance of a ranking algorithm. Essentially, this shows how confusing the classification model is in making predictions. Here, each column present in the confusion matrix depicts an actual class whereas the predicted class is represented by the corresponding columns.



	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Fig. 2 Confusion matrix

3. Proposed Method

Machine learning tools are employed in the proposed workflow of our experiment for the classification and analysis of the dataset. Data is gathered from a variety of sources at the first level to produce a good set of spam and ham radio data in text format, which is then fed as input into the model. We change the dataset from text format to a csv formatted file in the second stage of the experiment. Preprocessing is then used to improve input quality by putting different feature extraction strategies into practice. Next, the data set we are using is subjected to a classifier. As a result, the data set is used to train the data. To obtain results, the data is tested. The confusion matrix is obtained using the random forest technique and is then examined and discussed in the last stage of the experiment.

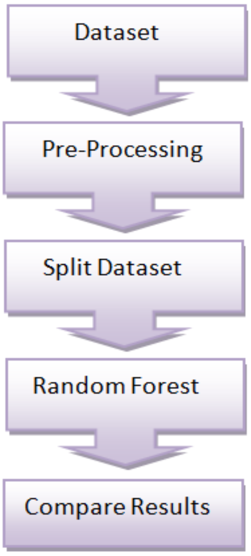


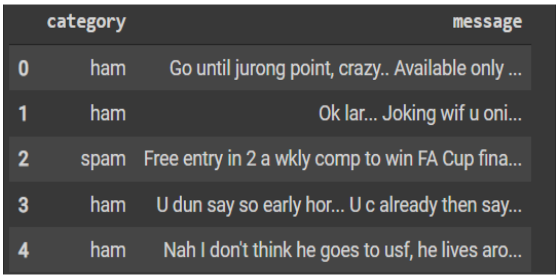
Fig. 3 Flowchart of the Proposed Algorithm

i). Dataset: In this project, we have used the spam dataset which consists of 747 spam messages and 4825 non-spam messages. Considering this ratio of spam to non-spam, the classification of a particular message into spam category is a big challenge, as most of the time the ML model would be biased towards the majority category. So, to tackle this problem, we increased the existing data by augmenting the minority category data with the help of state-of-the-art transformer-based models.

Table 1. Dataset Details

Dataset	**Training Samples**	**Testing Samples**	**Total**
UCI Dataset	6739	293	7032

This in turn increased the minority category data to 4 times the original sample size, i.e. training data comprises around 6739 samples after augmentation and 292 samples for the test set.



	category	message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

Fig. 4 Glimpse of the Dataset

ii). Data Preprocessing: We have pre-processed the data by stemming, lemmatizing, and finally removing stop words, and punctuations in the corpora. Once this is done, we move on to data augmentation, where we have used T5 based pre-trained model [1].

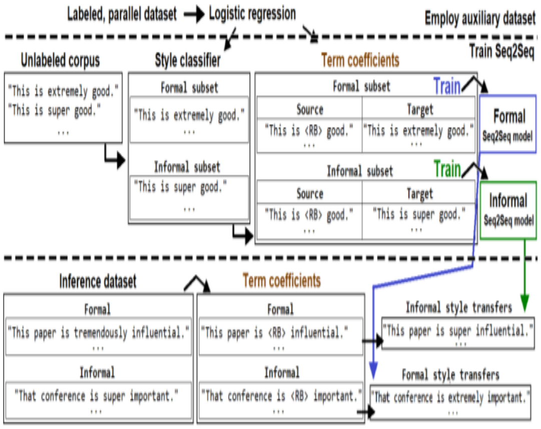
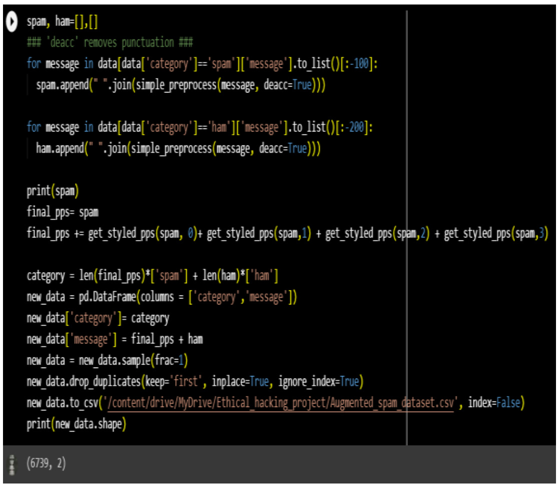


Fig. 5 Training architecture of pre-trained style transfer model

After augmentation was completed, we then moved to perform data cleaning, i.e. removing punctuations from the data, and then for feature extraction, we have used Sentence Transformer (SBERT) for creating a less dimensional sized vector (768D), for each of the sentences present in the corpus. Once we create the embeddings for the dataset, we then pass in the vectors to our random forest algorithm for the classification task.



```
spam, ham=[],[]
### 'deacc' removes punctuation ###
for message in data[data['category']=='spam']['message'].to_list()[:-100]:
  spam.append(" ".join(simple_preprocess(message, deacc=True)))

for message in data[data['category']=='ham']['message'].to_list()[:-200]:
  ham.append(" ".join(simple_preprocess(message, deacc=True)))

print(spam)
final_pps= spam
final_pps += get_styled_pps(spam, 0)+ get_styled_pps(spam,1) + get_styled_pps(spam,2) + get_styled_pps(spam,3)

category = len(final_pps)*['spam'] + len(ham)*['ham']
new_data = pd.DataFrame(columns = ['category','message'])
new_data['category']= category
new_data['message'] = final_pps + ham
new_data = new_data.sample(frac=1)
new_data.drop_duplicates(keep='first', inplace=True, ignore_index=True)
new_data.to_csv('/content/drive/MyDrive/Ethical_hacking_project/Augmented_spam_dataset.csv', index=False)
print(new_data.shape)
```

```
(6739, 2)
```

Fig. 6 Augmentation script

After augmentation, the training data comprises around 6739 samples after augmentation and 293 samples for the test set.

iii). Random Forest Classifier: The algorithm below is very concise and outlines the steps required for the creation of forest trees.

```
 Algorithm Start:

  Input:   A: number of nodes
               F: number of features
               B: number of trees to be grown

  Output:   H: the largest number of votes containing class
  While the stopping criteria are not true
    do
        From the training data B, pick a random self-starting sample S
        Create the $Ri$ tree from the selected auto start example S
        By performing the following operations:

        (1) Pick the characteristics f at random from F; where $f \ll F$
        (2) For the node y, find the best division point between the properties f
        (3) Parent node is divided into two descending nodes by the optimal division
        (4) Follow steps 1 to 3 until the maximum number of nodes is created
        (A) Repeat steps 1 to 4 to create your forest for B many times.
   End While
```

Create outputs for each tree that was built in step 1B.

For every tree that is made, start a fresh example at the root node.

Assign the example to the leaf node's respective class.

Gather the results from each tree's votes.

Enter the category (H) with the most votes to finish.

Algorithm End

4. Results and discussions

4.1 Results

We applied a Random Forest classifier for this SMS spam detection task, and we achieved the best results when compared to other research works. Below Table (2) shows the results of our random forest experiment.

Table 2. Results of Random Forest Classifier

Algorithm	**F1-score**	**Avg Recall**	**Avg Precision**	**Accuracy**
Our proposed classifier	0.955	0.95	0.96	0.96
G. V. Cormack et.al [3]	0.90	0.91	0.92	0.92
M. Taufiq Nuruzzaman et.al [4]	0.87	0.90	0.91	0.92

We have plotted the confusion matrix for our classifier below. Basically, the confusion matrix shows how much the classification model is confused during making the predictions. Therefore, the results of the confusion matrix obtained are also shown below

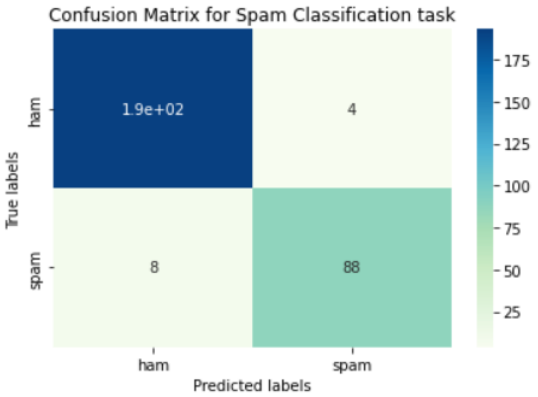
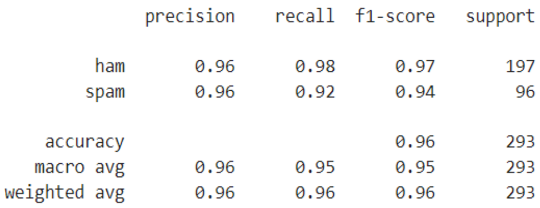


Fig. 8 Resultant Confusion Matrix

In the above confusion matrix, as we can see, the false positives and false negatives are very minute when compared to the total true positives and true negatives. Out of a total of 96 spam messages, our model had recognized it 88 times correctly (i.e ~92% accurate), and coming to the majority category (197 samples), 193 samples were predicted right as ham (~98%).



```
              precision    recall  f1-score   support

         ham       0.96      0.98      0.97       197
        spam       0.96      0.92      0.94        96

    accuracy                           0.96       293
   macro avg       0.96      0.95      0.95       293
weighted avg       0.96      0.96      0.96       293
```

Fig. 9 In-depth metrics

In figure 9, we can see the in-depth metrics analysis corresponding to each and every category. The ham category being the majority (197 samples), had around 96% precision, 98% recall, and an f1-score of 97%. For the minority category (96 samples), we had around 96% precision, 92% recall, and 94% for f1-score. Support metric provides us with the details regarding the number of samples present in each and every category.

5. Conclusion

In this paper, we have used the spam dataset which consists of 747 spam messages and 4825 non-spam messages. Considering this ratio of spam to non-spam, the classification of a particular message into spam category is a big challenge, as most of the time the ML model would be biased towards the majority category. So, to tackle this problem, we increased the existing data by augmenting the minority category data with the help of state-of-the-art transformer-based models. This in turn increased the minority category data to 4 times the original sample size, i.e training data comprises around 6739 samples after augmentation and 293 samples for the test set. After augmentation was completed, we then moved to perform data cleaning, i.e removing punctuations from the data, and then for feature extraction, we have used Sentence Transformer (SBERT) for creating a less dimensional sized vector (768D), for each of the sentences present in the corpus. Once we create the embeddings for the dataset, we then pass in the vectors to our random forest algorithm for the classification task. Our experiment shows that with high accuracy with the Random Forest classifier gives an efficient result.

References:

[1] Sayar Ul Hassan, Jameel Ahamed, Khaleel Ahmad, Analytics of machine learning-based algorithms for text classification, Sustainable Operations and Computers, Volume 3, 2022, Pages 238-248 https://www.sciencedirect.com/science/article/pii/S2666412722000101

[2] Jose Maria Gomez Hidalgo, Guillermo Cajigas Bringas, Enrique Puertas, Francisco Carrero García, "Content-Based SMS Spam Filtering", Proceedings of the 2006 ACM Symposium on Document Engineering, pp. 107-114, 2006. Available from: https://doi.org/10.1145/1166160.1166191.

[3] G. V. Cormack, J. M. G. Hidalgo, and E. P Sanz, Spam Filtering for Short Messages, Proceedings of the 16th ACM Conference on Information and Knowledge Management, pp. 313-320, 2007. Available from: https://doi.org/10.1145/1321440.1321486

[4] M. Taufiq Nuruzzaman, C. Lee, M. F. A. Bin Abdullah and D. Choi, Simple SMS spam filtering on independent mobile phones, Security and Communication Networks,vol. 5, no. 10, pp. 1209-1220, 2012. Available from: https://doi.org/10.1002/sec.577

[5] S. J. Delany and M. Buckley, SMS spam filtering: Methods and data, Expert Systems with Applications, Vol. 39,pp. 9899-9908, 2012. Available from: https://doi.org/10.1016/j.eswa.2012.02.053

[6] Roger Alan Stein, Patricia A. Jaques, João Francisco, An analysis of hierarchical text classification using word embeddings, Information Sciences, Volume 471, January 2019, Pages 216-232. https://www.sciencedirect.com/science/article/abs/pii/S0020025518306935

[7] A.Pumsirirat and L. Yan, Credit card fraud detection using deep learning based on auto-encoder and restricted Boltzmann machine, International Journal of Advanced Computer Science and Applications, Vol. 9, No. 1, pp. 18-25,2018 Available from (DOI) : 10.14569/IJACSA.2018.090103

[8] Dingkun Zhu, Chungang Yan, Mingjian Guang,Yu Xie, A Novel Information-Entropy-Based Feature Extraction Method for Transaction Fraud Detection, 2021, 4th International Conference on Intelligent Autonomous Systems (ICoIAS),pp.129-133,2021,Available from: https://doi.org/10.1109/ICoIAS53694.2021.00031

[9] Zhaohui Zhang, Ligong Chen, Qiuwen Liu, Pengwei Wang, A Fraud Detection Method for Low-Frequency Transaction, IEEE Access, vol.8, pp.25210-25220, 2020. Available from: Digital Object Identifier 10.1109/ACCESS.2020.2970614 https://web.archive.org/web/20201108180902id_/https://ieeexplore.ieee.org/ielx7/6287639/8948470/08977544.pdf

[10] Lutao Zheng, Guanjun Liu, Chungang Yan, Changjun Jiang, Mengchu Zhou, Maozhen Li, Improved TrAdaBoost and its Application to Transaction Fraud Detection, IEEE Transactions on Computational Social Systems, vol.7, no.5, pp.1304-1316, 2020. https://doi.org/10.1109/TCSS.2020.3017013

[11] Chao Fan Yang, Guan Jun Liu, Chun Gang Yan, A k-means-based and no-super-parametric Improvement of AdaBoost and its Application to Transaction Fraud Detection, 2020 IEEE

International Conference on Networking, Sensing and Control (ICNSC), pp.1-5, 2020. Available from :https://doi.org/10.1109/ICNSC48988.2020.9238121

[12] B.Branco,P.Abreu, A.S.Gomes, M.S.C. Almeida, J.T. Ascensão and P. Bizarro, Interleaved sequence RNNs for fraud detection, Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, pp. 3101-3109, 2020.Available from : https://doi.org/10.1145/3394486.3403361

[13] A.O.Balogun, S.Basri, S.J. Abdulkadir and A.S.Hashim,Performance analysis of feature selection methods in software defect prediction: A search method approach, Appl. Sci., vol. 9, no. 13, pp. 2764, Jul. 2019. Available from : https://doi.org/10.3390/app9132764

[14] Błaszczyński, A.T.de Almeida Filho, A.Matuszyk, M.Szelński and R.Słowiński, Auto loan fraud detection using dominance-based rough set approach versus machine learning methods, Expert Syst. Appl., vol. 163, Jan. 2021. Available from : https://doi.org/10.1016/j.eswa.2020.113740