[1] Rajshree Singh*

[2] Dr. Reena Srivastava

BALANCING SARCASTIC HINGLISH SHORT TEXT DATA USING AUGMENTATION TECHNIQUES WITH HANDLING SPELLING VARIATIONS

Abstract: - In the real world, there is a significant presence of imbalanced data due to the fact that the classes that make up the datasets are not evenly distributed. Even when using methods that are traditionally used to achieve class balance, such as re-sampling & re-weighting, current deep learning still faces a significant obstacle because of the class imbalance. This study's major objective is proposing a data augmentation technique to balance the data to improve the sample sizes for the minority classes. Python, a well-known programming language, & multiple methods of machine learning are being employed in the execution of this study. Classification models like Logistic Regression, Naïve Bayes, Support Vector Machine, Decision Tree, Random Forest, Extra Trees Classifier, AdaBoost classifier, Gradient Boost classifier was used to implement this study. Precision, recall, & F-score were used to determine which model would be the most effective. According to the findings of this study's analysis, the Naive Bayes approach, which has a F1-Score of 95.85% & has Wn = 3, Cn = 3, & CWn =3 as its parameters, is the technique that yields the most accurate results.

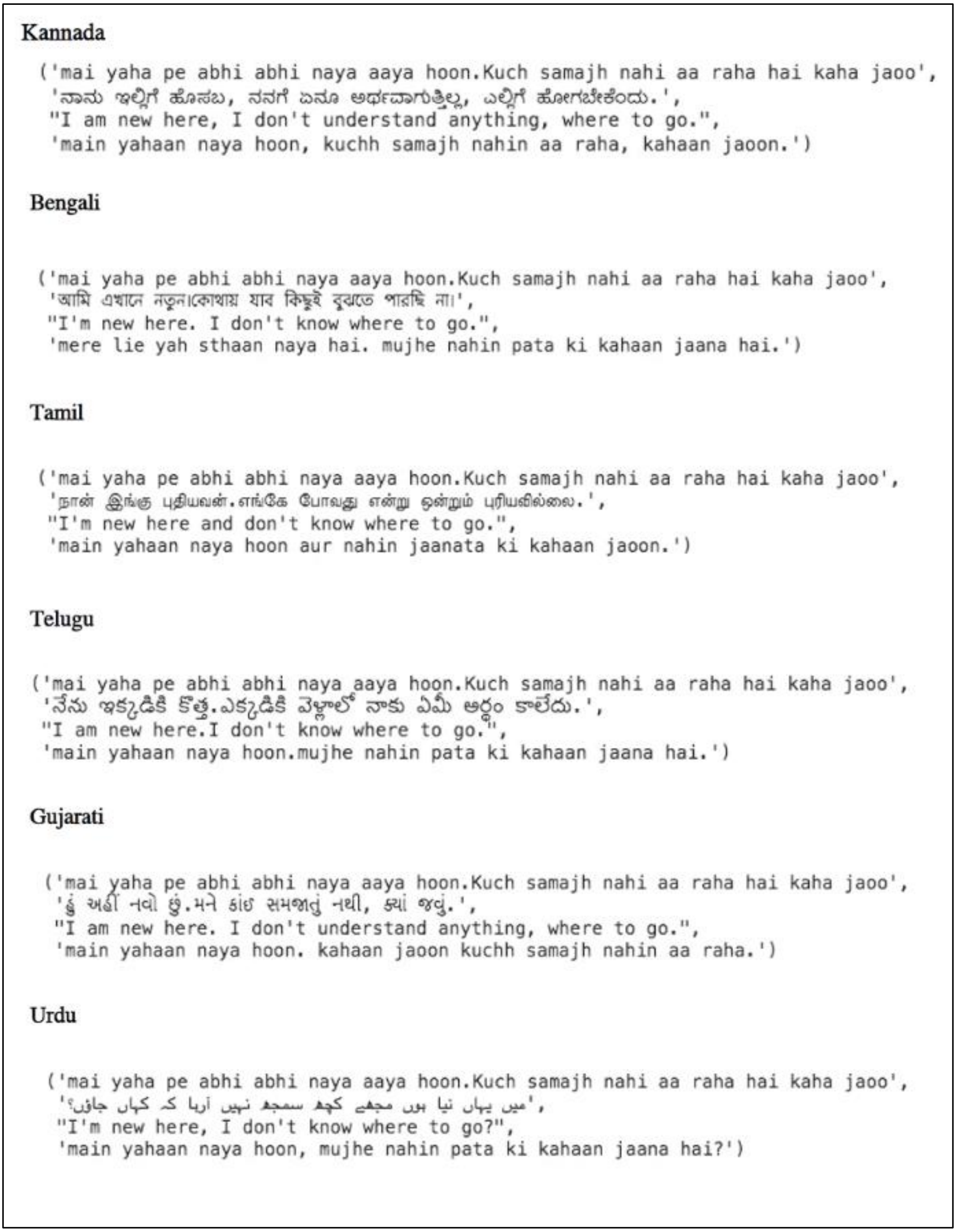
Keywords: Data Balancing; Data augmentation; Machine Learning Techniques; Hinglish

1. INTRODUCTION

Real-world data frequently have substantial class imbalance & long-tailed distributions (Liu et al., 2019). When trained on unbalanced datasets, where certain categories include much more examples than others, machine learning models have a tendency to learn effectively on majority groups' samples while generalising poorly on minority classes (Cao et al., 2019). In situations with high stakes, such as the identification of a rare disease or discrimination against minority groups, learning from such seminars is essential. For pre-trained language models (LMs) in natural language processing (NLP), data imbalance is particularly difficult. In this study, it has become increasingly necessary to analyse social content produced in "Hinglish" (Hindi + English) on platforms like Twitter, WhatsApp, Facebook etc. due to a fast increase in fluency & users in the linguistically diverse nation of India. In this study, the classification of social information written in Hindi-English is tackled utilising deep learning techniques (Gupta, 2019). For example: Consider this sentence "mai yaha pe abhi abhor naya aaya hoon. Kuch samajh nahi aa raha hai kaha jaoo". Converting this sentence into different languages will produce the following outcomes:

[1] Research Scholar, School of Computer Application, Babu Banarasi Das University, Lucknow. rajshree.singh07@bbdu.ac.in

[2]Dean, School of Computer Application, Babu Banarasi Das University, Lucknow. dean.soca@bbdu.ac.in



Kannada

```
('mai yaha pe abhi abhi naya aaya hoon.Kuch samajh nahi aa raha hai kaha jaoo',
 'ನಾನು ಇಲ್ಲಿಗೆ ಹೊಸಬ, ನನಗೆ ಏನೂ ಅರ್ಥವಾಗುತ್ತಿಲ್ಲ, ಎಲ್ಲಿಗೆ ಹೋಗಬೇಕೆಂದು.',
 "I am new here, I don't understand anything, where to go.",
 'main yahaan naya hoon, kuchh samajh nahin aa raha, kahaan jaoon.')
```

Bengali

```
('mai yaha pe abhi abhi naya aaya hoon.Kuch samajh nahi aa raha hai kaha jaoo',
 'আমি এখানে নতুন।কোথায় যাব কিছুই বুঝতে পারছি না।',
 "I'm new here. I don't know where to go.",
 'mere lie yah sthaan naya hai. mujhe nahin pata ki kahaan jaana hai.')
```

Tamil

```
('mai yaha pe abhi abhi naya aaya hoon.Kuch samajh nahi aa raha hai kaha jaoo',
 'நான் இங்கு புதியவன்.எங்கே போவது என்று ஒன்றும் புரியவில்லை.',
 "I'm new here and don't know where to go.",
 'main yahaan naya hoon aur nahin jaanata ki kahaan jaoon.')
```

Telugu

```
('mai yaha pe abhi abhi naya aaya hoon.Kuch samajh nahi aa raha hai kaha jaoo',
 'నేను ఇక్కడికి కొత్త.ఎక్కడికి వెళ్లాలో నాకు ఏమీ అర్థం కాలేదు.',
 "I am new here.I don't know where to go.",
 'main yahaan naya hoon.mujhe nahin pata ki kahaan jaana hai.')
```

Gujarati

```
('mai yaha pe abhi abhi naya aaya hoon.Kuch samajh nahi aa raha hai kaha jaoo',
 'હું અહીં નવો છું.મને કાંઈ સમજાતું નથી, ક્યાં જવું.',
 "I am new here. I don't understand anything, where to go.",
 'main yahaan naya hoon. kahaan jaoon kuchh samajh nahin aa raha.')
```

Urdu

```
('mai yaha pe abhi abhi naya aaya hoon.Kuch samajh nahi aa raha hai kaha jaoo',
 'میں یہاں نیا ہوں مجھے کچھ سمجھ نہیں آرہا کہ کہاں جاؤں؟',
 "I'm new here, I don't know where to go?",
 'main yahaan naya hoon, mujhe nahin pata ki kahaan jaana hai?')
```

Figure 1. Text that converts into different languages

This study's major goal is to suggest a data augmentation technique to balance the datasets which consist of Hinglish data. Data augmentation is an approach that is utilised in computer vision & NLP to deal with a lack of data diversity & a paucity of data. The processing of natural language, on the other hand, is notoriously difficult due to the intrinsic complexity of the language itself. Creating augmented visuals, on the other hand, is a pretty straightforward procedure. It is unable to replace every term with its synonym, & even if it did, the context would change as a result. Also, the performance of the model is improved as a result of data augmentation, which increases the training data size. Also, the greater performance it can get out of the model, the more data that need to collect. The distribution of the newly created enhanced data shouldn't be overly similar or dissimilar from the distribution of the original data. This could result in overfitting or poor performance through effective Data Augmentation (DA) techniques, both of which should strive for equilibrium. The figure that follows provides a taxonomy of the kinds that can be found in the textual domain.

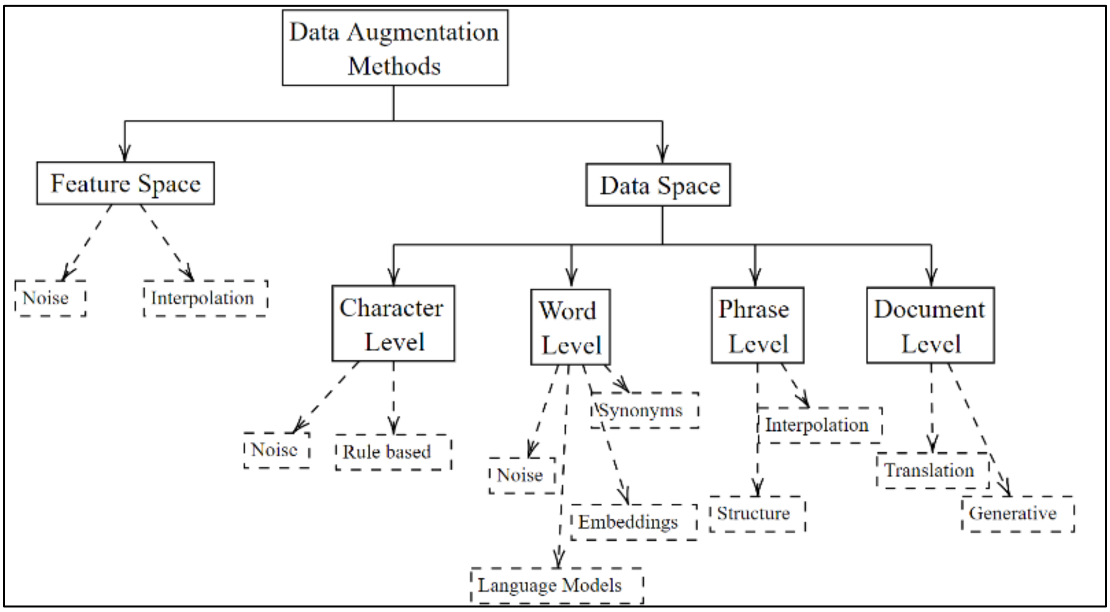


Figure 2. Taxonomy & grouping for different data augmentation methods (Bayer et al., 2022)

In the following section, this study will elaborate on the prior investigations that were conducted on this concept.

2. LITERATURE REVIEW

Table 1: Literature review

AUTHOR AND YEAR	METHOD	FINDINGS
(Dai et al., 2023)	An AugGPT-based text data augmentation method was suggested by the authors	The suggested AugGPT strategy outperforms state-of-the-art text data augmentation techniques in experiments on few-shot learning text classification tasks in terms of testing accuracy & distribution of the supplemented samples.
(Bayer et al., 2022)	This survey, which is concerned with data augmentation approaches for textual categorization, intends to give academics & practitioners a clear & thorough overview by describing the objectives & uses of data augmentation in detail & by using a taxonomy for existing studies.	Divide more than 100 methods into 12 separate groups based on the taxonomy & provide state-of-the-art references that explain which ways are particularly promising by connecting them.
(Wang et al., 2021)	A context extension framework & a data augmentation technique are both proposed based on such logical knowledge. With the former, the context is expanded to include implicit logical expressions that abide by logical equivalence rules. The ReClor dataset was used for the studies in this study.	The results demonstrate that the method provides state-of-the-art performance, & both the data augmentation algorithm & the logic-driven context extension framework can help to increase accuracy.

(Karimi et al., 2021)	Introduced the AEDA (An Easier Data Augmentation) technique to help with text categorization task performance. Only sporadic punctuation mark insertions into the original text are included by AEDA. Compared to the EDA method **(Wei & Zou, 2019),** which may be used to compare the outcomes, this method is simpler to implement for data augmentation. Additionally, it maintains the word order while shifting where they are in the phrase, which improves overall performance. Moreover, the deletion operation in EDA might result in information loss, which in turn causes the network to be misled, whereas AEDA retains all of the input data.	According to this study, the models perform better when trained on AEDA-augmented data than when trained on EDA-augmented data across all five datasets.
(Guo, 2020)	This study adopts a nonlinear interpolation policy for both the input & label pairs, in contrast to Mixup, where the input & label pairs share the same, linear, scalar mixing policy.	The cited empirical investigations further demonstrate that the strategy's out-of-manifold samples encourage training samples in each class to form a small, distant representation cluster.

It is evident from literature evaluations that there is a lack of a perfect approach for data balancing with a high accuracy level. Therefore, the primary goal of this research is to provide a data augmentation approach to balance the data set.

3. METHODOLOGY

The current dataset given in the study is comprised of 5250 tweets, of which 504 tweets have been labelled as being sarcastic & the remaining 4746 tweets have not been labelled as being sarcastic. At the start of the process, the sarcasm class had 504 samples while the non-sarcasm class had 4746 samples. To augment the dataset, language translation techniques were used, resulting in the creation of 3024 new samples. Although some of these new sentences were duplicates, they were identified & removed to create a set of unique samples. This process reduced the total number of samples in the sarcasm class to 3502 (504 original + 2998 new & unique). Despite this, the non-sarcasm class still had a higher number of samples, with 4746 samples. In order to balance the dataset, additional samples were created using Random Insertion (RI), Random Deletion (RD), Random Swap (RS), & Synonym Replacement (SR) techniques. These techniques were applied to both classes to maintain their relative proportions. As a result, both the sarcasm & non-sarcasm classes now contain an equal number of samples, with 4746 samples in each class.

In this research, it is using the data augmentation technique to increase the samples of the minority class. The Data Augmentation (DA) Technique is a process that allows us to artificially enhance the number of training data by producing various versions of genuine datasets without actually collecting the data. This allows us to train more accurately & efficiently. It is necessary to modify the data in order to maintain the class categories in order to achieve improved performance in the classification task. Using the data augmentation technique, it is possible to differentiate between words with the same sound, which are referred to as homophones. Also takes into account the order of characters, for example "apna" and "paan." Strategies followed to augment the data are as follows:

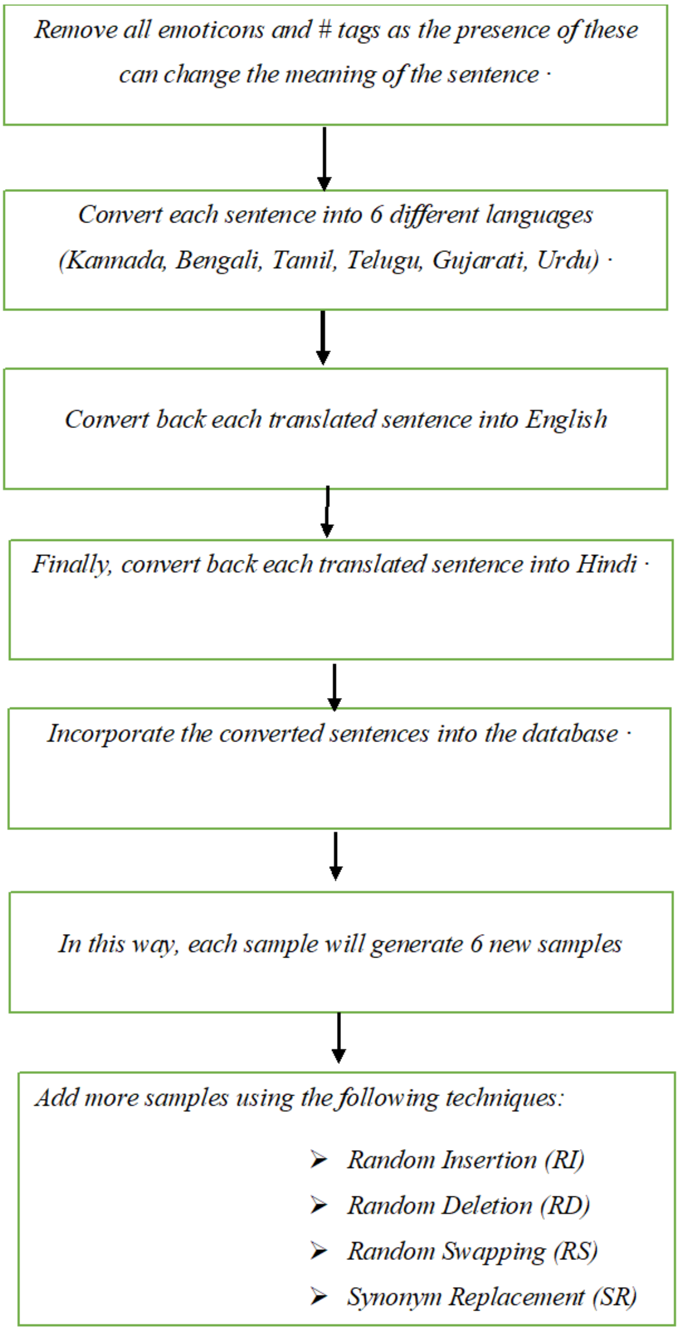

Remove all emoticons and # tags as the presence of these can change the meaning of the sentence ·
Convert each sentence into 6 different languages (Kannada, Bengali, Tamil, Telugu, Gujarati, Urdu) ·
Convert back each translated sentence into English
Finally, convert back each translated sentence into Hindi ·
Incorporate the converted sentences into the database ·
In this way, each sample will generate 6 new samples
Add more samples using the following techniques:
Random Insertion (RI)
Random Deletion (RD)
Random Swapping (RS)
Synonym Replacement (SR)

Figure 3. Strategies used to augment the data

The following is an illustration of Data Augmentation Techniques for Text Classification:

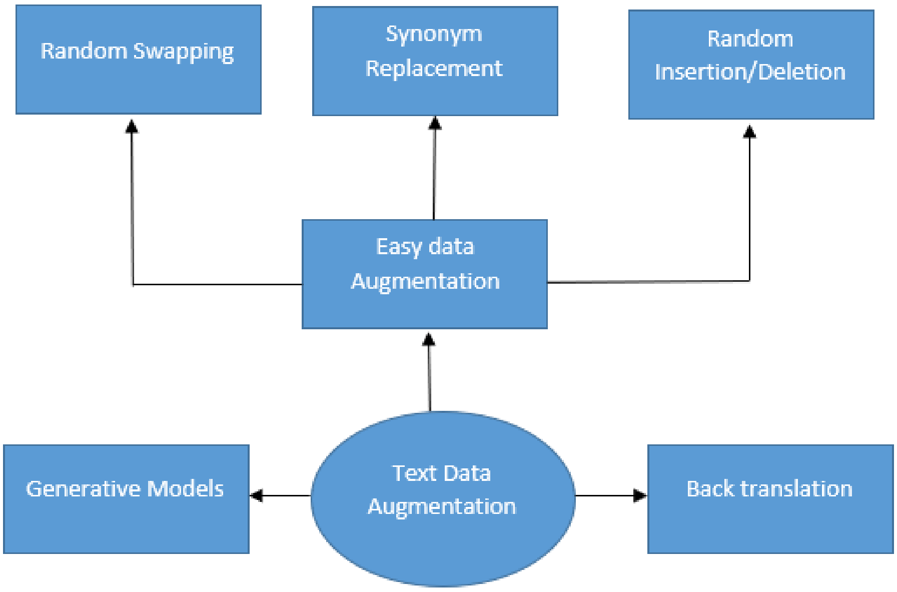
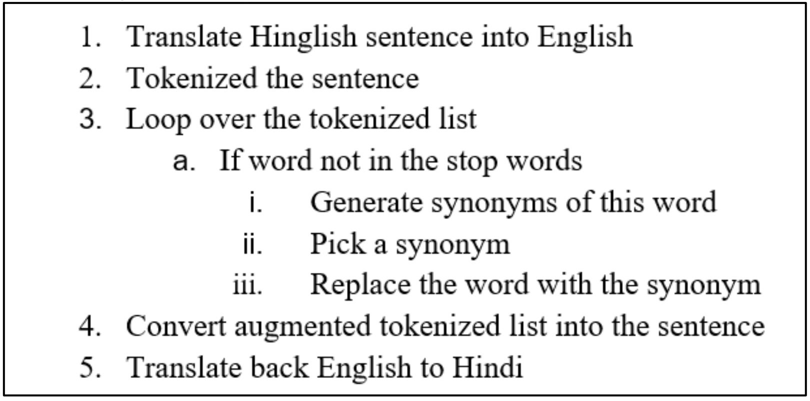


Figure 3: Data Augmentation Techniques for Text Classification

A word is randomly selected from the phrase and replaced with one of these word synonyms in this Easy

Data Augmentation (EDA) technique, or two words are selected and switched in the statement.

a. **Synonym Replacement:** Technique entails replacing a term with one of its synonyms. WordNet, a big linguistic database, is used to find pertinent synonyms. This function, used in this study, locates and pre-processes a word's synonyms. Following are the steps:



```
1. Translate Hinglish sentence into English
2. Tokenized the sentence
3. Loop over the tokenized list
    a. If word not in the stop words
        i. Generate synonyms of this word
        ii. Pick a synonym
        iii. Replace the word with the synonym
4. Convert augmented tokenized list into the sentence
5. Translate back English to Hindi
```

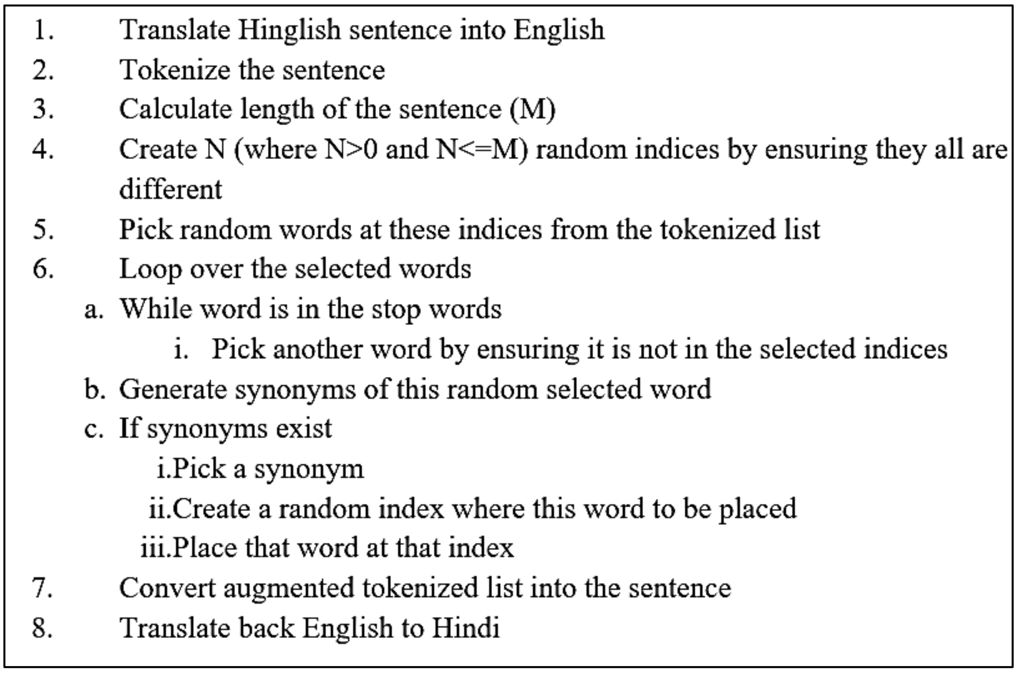
Example:

Bollywood ke writers se savinay nivedan hai ki wo hindi mein sochne Ka kasht Karen. Taaki unki filmein jyaada logon ko samajh aa sake. (**Hinglish Sentence**)
It is a humble request to the writers of Bollywood that they should try to think in Hindi. so that more people can understand their films. (**English Translated Sentence**)

Words humble, understand are synonym replaced by words lowly, empathize respectively

It is a lowly request to the writers of Bollywood that they should try to think in Hindi. so that more people can empathize their films. (**Synonyms Replaced English Sentence**)
boleevud ke lekhak se vinamr nivedan hai ki ve hindee mein sochane ka prayaas karen. taaki adhik se adhik log unakee philmon se sahaanubhooti rakh saken. (**Translated Back To Hindi**)

b. **Random Insertion**: Technique inserts synonyms of a word into a document in an unpredictable order. Following steps are implemented:



```
1.  Translate Hinglish sentence into English
2.  Tokenize the sentence
3.  Calculate length of the sentence (M)
4.  Create N (where N>0 and N<=M) random indices by ensuring they all are
    different
5.  Pick random words at these indices from the tokenized list
6.  Loop over the selected words
    a. While word is in the stop words
        i. Pick another word by ensuring it is not in the selected indices
    b. Generate synonyms of this random selected word
    c. If synonyms exist
        i.Pick a synonym
        ii.Create a random index where this word to be placed
        iii.Place that word at that index
7.  Convert augmented tokenized list into the sentence
8.  Translate back English to Hindi
```

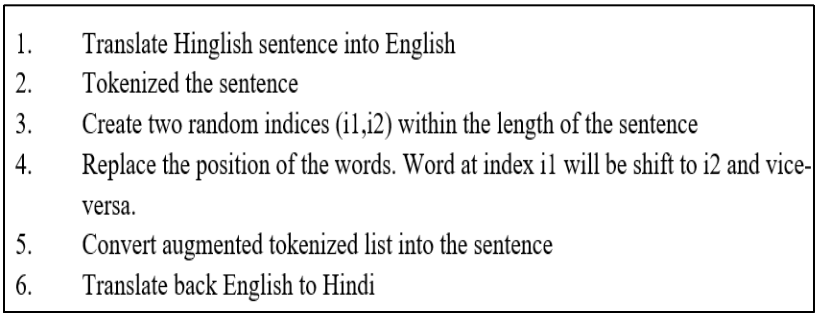
Example:

Sports ko nayi disha dijiye aur cricket hi nahi har khel n khilaadi ko baraabari ka samman dilaayiye **(Hinglish Sentence)**
Give a new direction to sports and give equal respect to every player, not just cricket (**English Translated Sentence**)

Word <u>abide by</u> is Randomly Inserted

Give a new direction abide by to sports and give equal respect to every player, not just cricket (**Randomly Inserted English Sentence**)
khel ko naee disha den aur sirph kriket ko hee nahin, har khilaadee ko samaan sammaan den (**Translated Back To Hindi**)

c. **Random Swap**: Technique involves changing the sequence of two words in a sentence in a way that is completely random.



```
1.  Translate Hinglish sentence into English
2.  Tokenized the sentence
3.  Create two random indices (i1,i2) within the length of the sentence
4.  Replace the position of the words. Word at index i1 will be shift to i2 and vice-
    versa.
5.  Convert augmented tokenized list into the sentence
6.  Translate back English to Hindi
```

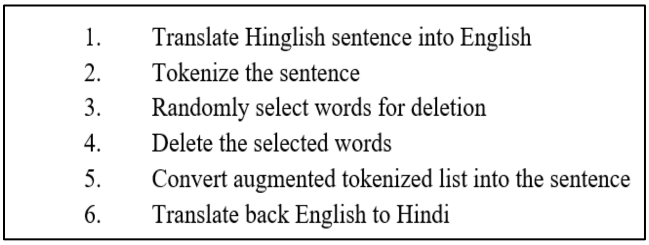
Example:

Sports ko nayi disha dijiye aur cricket hi nahi har khel n khilaadi ko baraabari ka samman dilaayiye (**Original Sentence**)

Words <u>nayi disha</u> are randomly swapped by <u>disha nayi</u>

Sports ko disha nayi dijiye aur cricket hi nahi har khel n khilaadi ko baraabari ka samman dilaayiye (**Randomly Swapped Sentence**)

d. **Random Deletion**, Technique removes each word in the sentence randomly using probability. Following are the steps:



1. Translate Hinglish sentence into English
2. Tokenize the sentence
3. Randomly select words for deletion
4. Delete the selected words
5. Convert augmented tokenized list into the sentence
6. Translate back English to Hindi

Example:

Sports ko nayi disha dijiye aur cricket hi nahi har khel n khilaadi ko baraabari ka samman dilaayiye (**Original Sentence**)

Word <u>nayi</u> is randomly deleted here

Sports ko disha dijiye aur cricket hi nahi har khel n khilaadi ko baraabari ka samman dilaayiye (**Randomly Deleted Sentence**)

PREPROCESSING: There are 5250 samples in the database out of which 4746 samples are non-sarcastic which comprised 90.4%. Since it is creating 6 new instances of any sample of the sarcastic class. So after augmentation total samples in the minority class will reach up to 3024 which makes the distribution fair. After augmentation, the contribution of the no-sarcasm class will be 61.08%.

EVALUATION MATRICES: In this study, Precision, Recall, & Fscore are the three evaluation matrices that are utilised. The amount of correct positive predictions that were made can be quantified using the precision metric. Therefore, precision is the calculation that determines the accuracy for the minority class. To determine it, take the ratio of accurately anticipated positive examples & divide it by the total number of positive examples that were forecasted. This will give you the accuracy rate. In this context, "recall" is defined as the number of components that truly belong to the positive class divided by the total number of true positives. The harmonic mean of a system's precision & recall values is what constitutes an F-score for that system. Precision, Recall & F1-Score are calculated to evaluate performance of each classification model.

$$\text{Precision (P)} = \frac{\text{TP}}{\text{TP+FP}} \tag{1}$$

$$\text{Recall (R)} = \frac{\text{TP}}{\text{TP+FN}} \tag{2}$$

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{3}$$

$$\text{FScore} = \frac{2*(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})} \tag{4}$$

4. RESULTS AND DISCUSSIONS

Classification models like Logistic Regression, Naïve Bayes, Support Vector Machine, Decision Tree, Random forest, Extra Trees Classifier, AdaBoost classifier, Gradient Boost classifier was used to implement this study. All the models are performed by using different combination of Word N Grams (Wn); Char N Grams (Cn); Char Word N Grams (CWn). The following are the results obtained after applying data augmentation:

Table 2: Wn = 3; Cn =3; WCn =3

	Accuracy	**Precision**	**Recall**	**FScore**
Logistic Regression	89.89	84.60	94.48	89.27
Naive Bayes	95.93	94.56	97.17	95.85
Support Vector Machine	94.10	93.22	94.83	94.02
Random Forest	92.49	92.02	92.81	92.41
Decision Tree	85.25	87.99	83.29	85.58
Extra Trees Classifier	94.42	93.71	94.99	94.35
Gradient Boosting Classifier	84.69	81.85	86.62	84.17
Adaboost Classifier	85.18	83.19	86.49	84.81

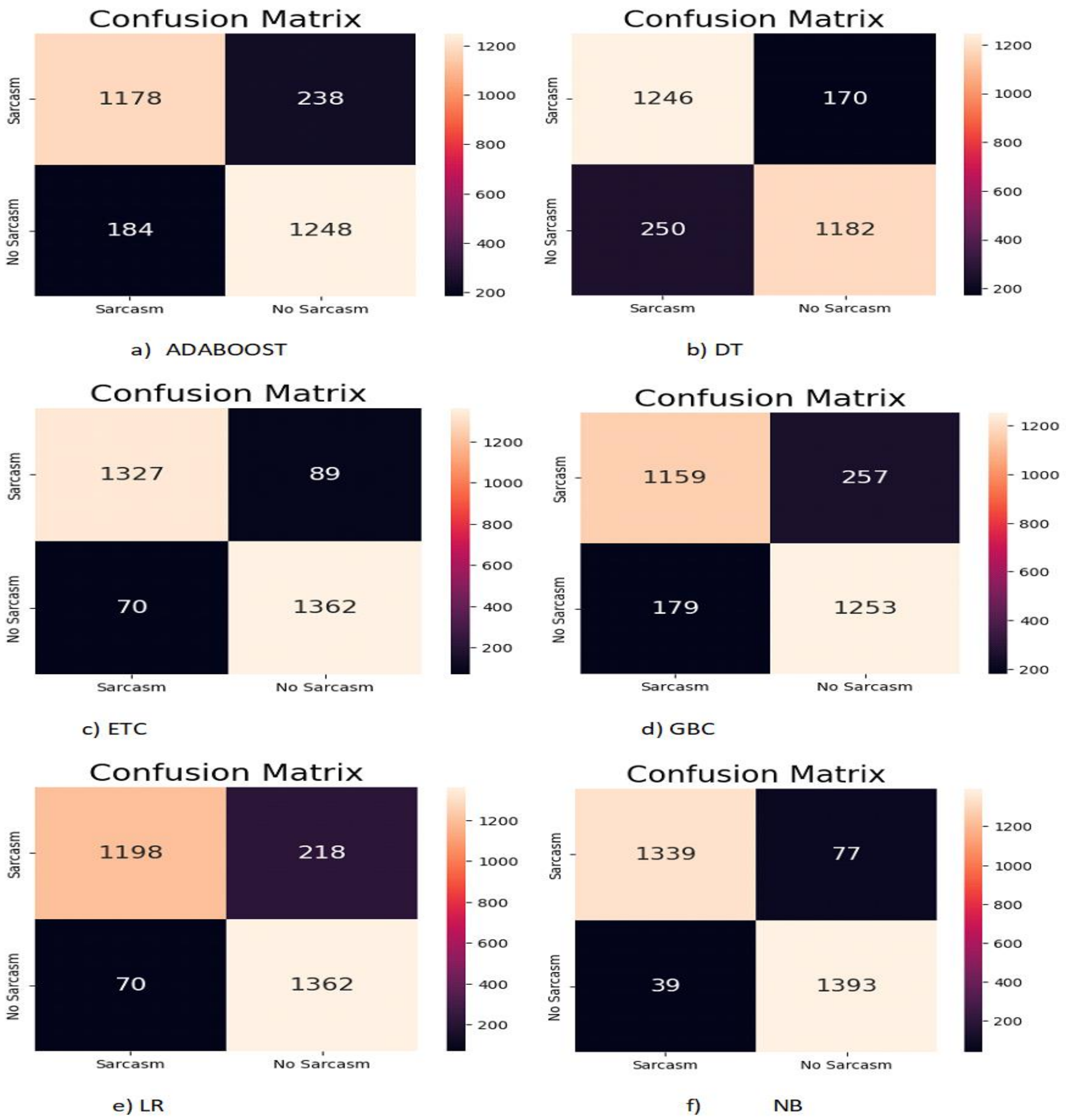
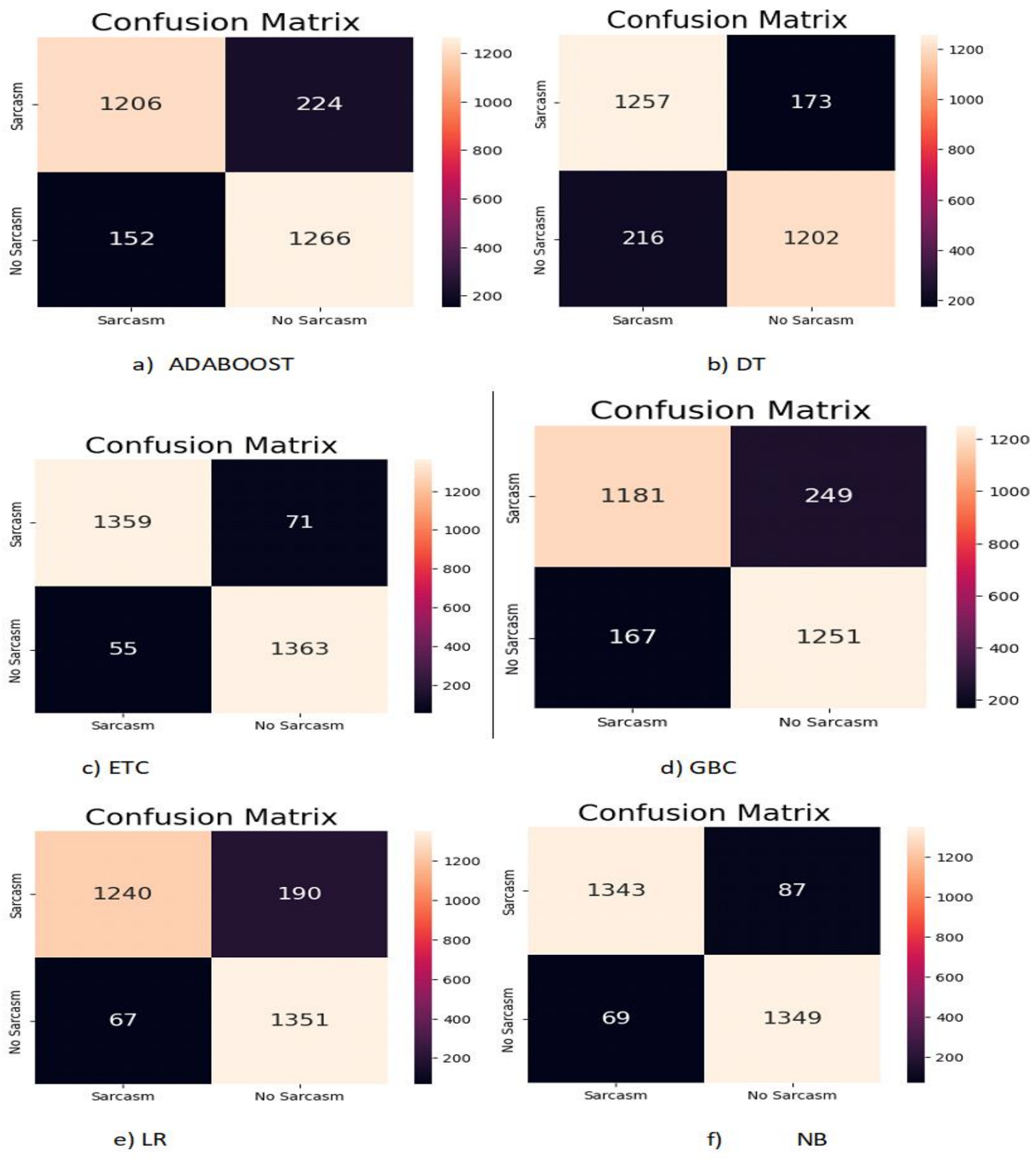


Figure 4. Confusion Matrix of **Wn = 3; Cn =3; WCn =3**

Table 3. Wn = 3 Cn =3 WCn =4

	Accuracy	**Precision**	**Recall**	**FScore**
Logistic Regression	90.98	86.71	94.87	90.61
Naive Bayes	94.52	93.92	95.11	94.51
Support Vector Machine	95.37	94.76	95.96	95.36
Random Forest	93.71	93.57	93.89	93.73
Decision Tree	86.34	87.9	85.34	86.6
Extra Trees Classifier	95.58	95.03	96.11	95.57
Gradient Boosting Classifier	85.39	82.59	87.61	85.03
Adaboost Classifier	86.80	84.34	88.81	86.52



a) ADABOOST

b) DT

c) ETC

d) GBC

e) LR

f) NB

Figure 5: Confusion Matrix of **Wn = 3; Cn =3; WCn =4**

Table 4. Wn = 3 Cn =4 WCn =3

	Accuracy	**Precision**	**Recall**	**FScore**
Logistic Regression	89.85	84.32	94.29	89.03
Naive Bayes	94.66	94.39	94.66	94.52
Support Vector Machine	95.70	94.24	94.86	94.55
Random Forest	92.70	92.73	92.34	92.53
Decision Tree	86.13	88.35	84.05	86.15
Extra Trees Classifier	94.87	94.46	95.01	94.73
Gradient Boosting Classifier	84.30	81.80	85.42	83.57
Adaboost Classifier	85.74	84.53	86.02	85.27

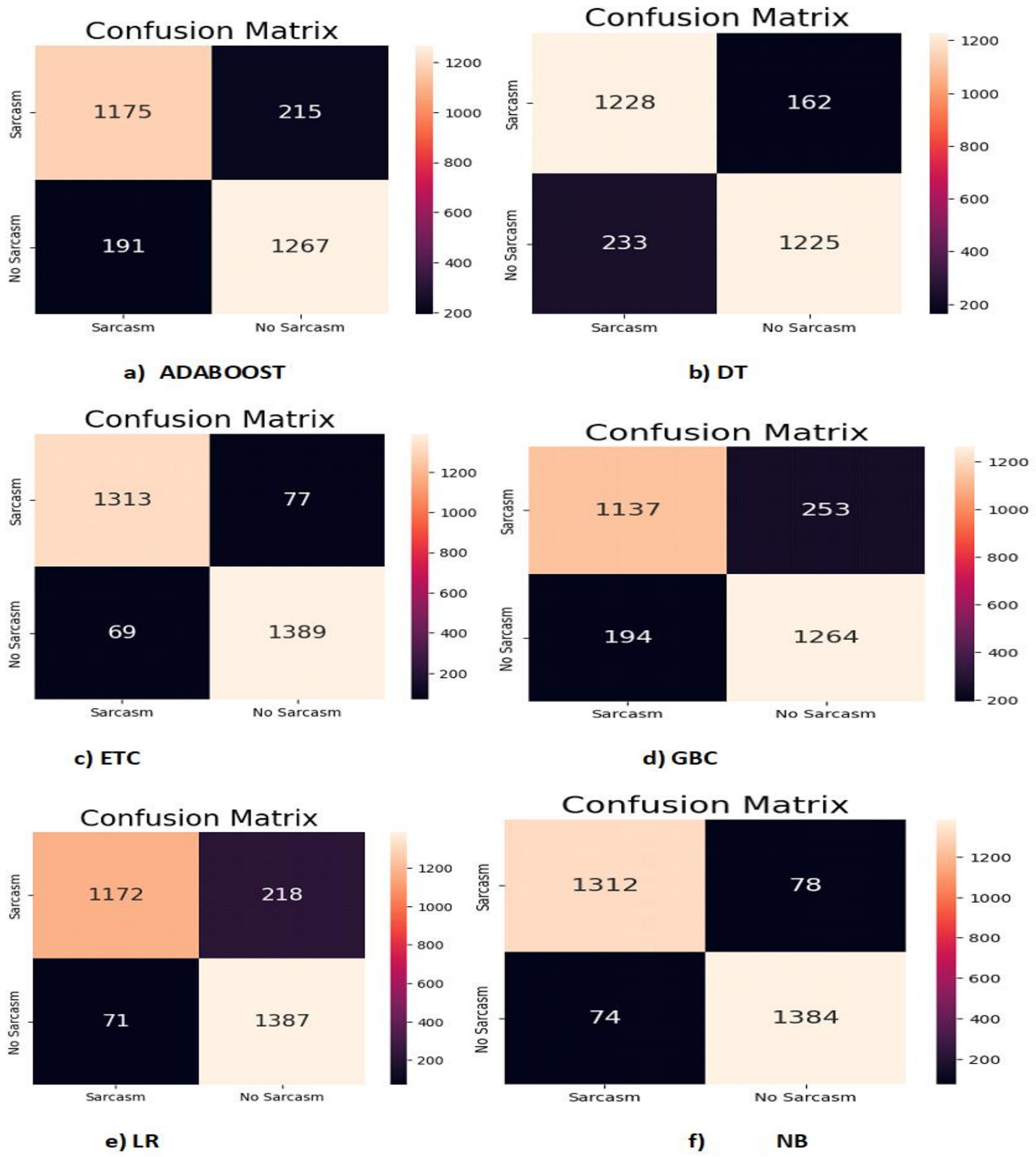
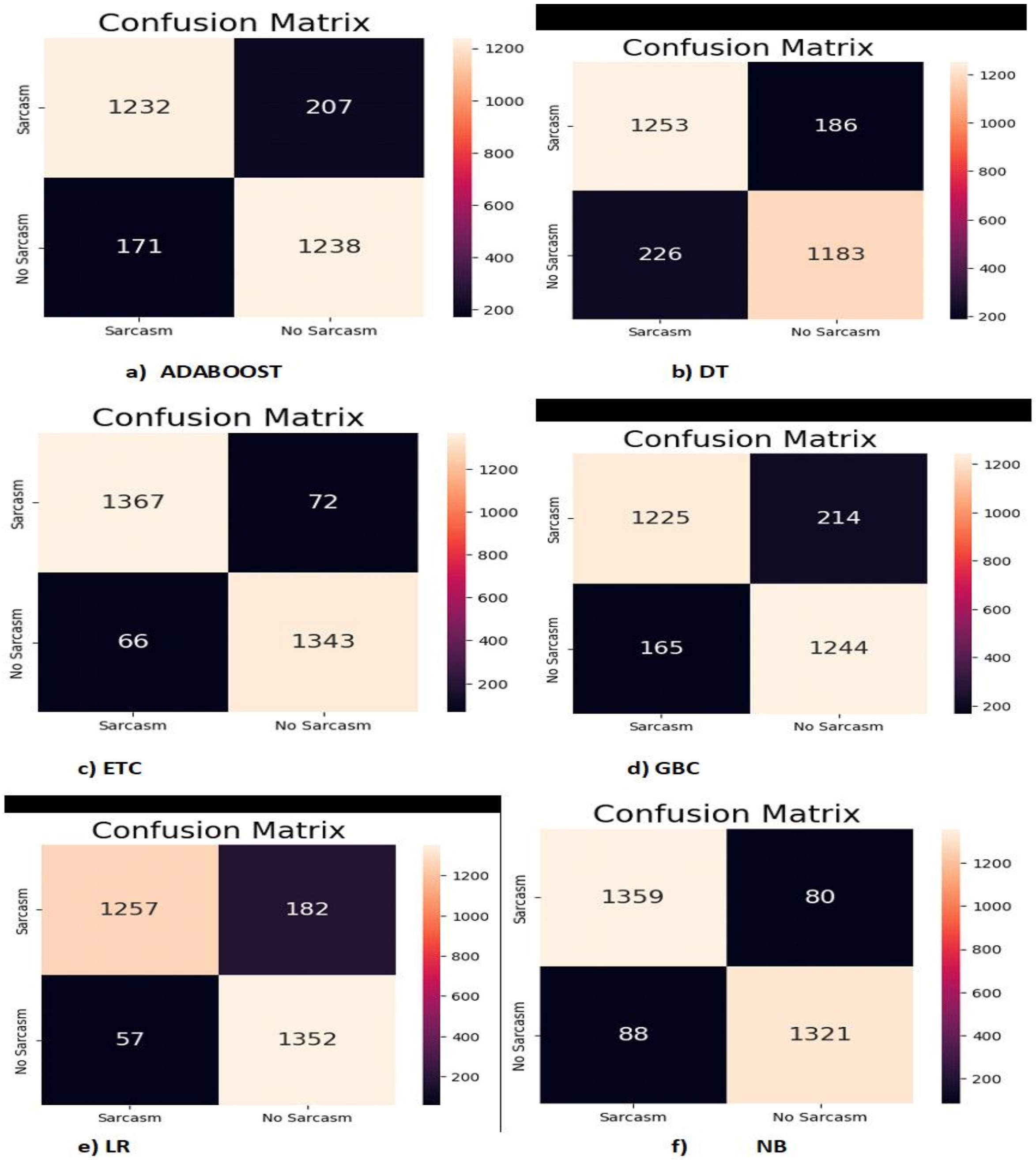


Figure 6. Confusion Matrix of **Wn = 3; Cn =4; WCn =3**

Table 5. Wn = 3 Cn =4 WCn =4

	Accuracy	**Precision**	**Recall**	**FScore**
Logistic Regression	91.61	87.35	95.66	91.30
Naive Bayes	94.10	94.44	93.92	94.18
Support Vector Machine	95.05	94.93	95.26	95.09
Random Forest	93.64	93.61	93.80	93.70
Decision Tree	85.53	87.07	84.72	85.88
Extra Trees Classifier	95.15	95.00	95.39	95.19
Gradient Boosting Classifier	86.69	85.13	88.13	86.60
Adaboost Classifier	86.73	85.62	87.81	86.60



a) ADABOOST

b) DT

c) ETC

d) GBC

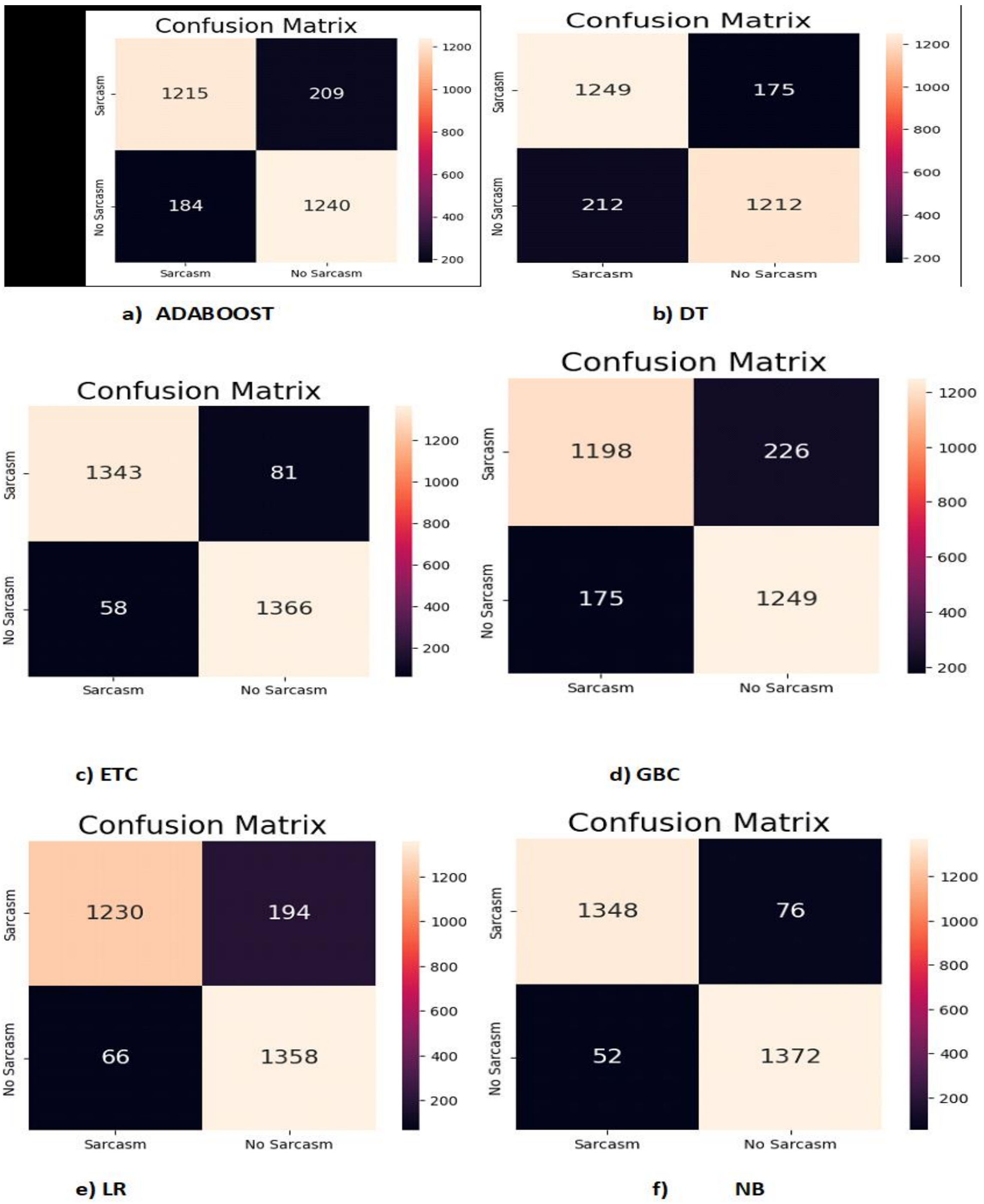
e) LR

f) NB

Figure 7. Confusion Matrix of Wn = 3; Cn =4; WCn =4

Table 6. Wn = 4 Cn =3 WCn =3

	Accuracy	**Precision**	**Recall**	**FScore**
Logistic Regression	90.87	86.38	94.91	90.44
Naive Bayes	95.51	94.66	96.29	95.47
Support Vector Machine	95.08	94.38	95.73	95.05
Random Forest	93.47	93.47	93.47	93.47
Decision Tree	86.41	87.71	85.49	86.59
Extra Trees Classifier	95.12	94.31	95.86	95.08
Gradient Boosting Classifier	85.92	84.13	87.25	85.66
Adaboost Classifier	86.20	85.32	86.85	86.08



a) ADABOOST

b) DT

c) ETC

d) GBC

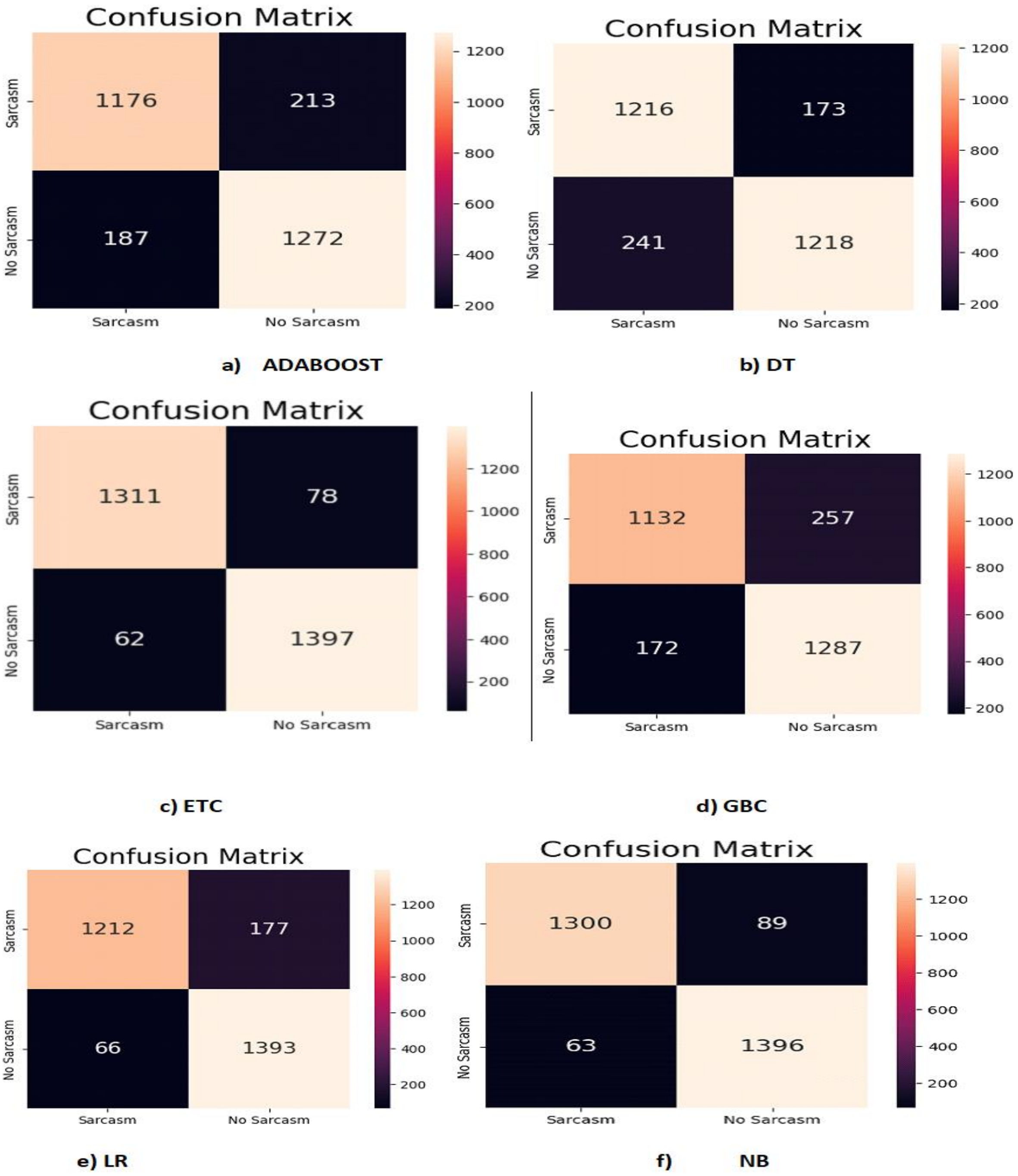
e) LR

f) NB

Figure 8. Confusion Matrix of Wn = 4; Cn =3; WCn =3

Table 7. Wn = 4 Cn =3 WCn =4

	Accuracy	**Precision**	**Recall**	**FScore**
Logistic Regression	91.47	87.26	94.84	90.89
Naive Bayes	94.66	93.59	95.38	94.48
Support Vector Machine	95.05	94.53	95.28	94.90
Random Forest	93.50	93.09	93.56	93.32
Decision Tree	85.46	87.54	83.46	85.45
Extra Trees Classifier	95.08	94.38	95.48	94.93
Gradient Boosting Classifier	84.94	81.50	86.81	84.07
Adaboost Classifier	85.96	84.67	86.28	85.47



a) ADABOOST

b) DT

c) ETC

d) GBC

e) LR

f) NB

Figure 9. Confusion Matrix of **Wn = 4; Cn =3; WCn =4**

Table 8. Wn = 4 Cn =4 WCn =3

	Accuracy	Precision	Recall	FScore
Logistic Regression	91.01	86.47	95.24	90.64
Naive Bayes	95.19	94.14	96.22	95.17
Support Vector Machine	94.59	94.98	94.32	94.65
Random Forest	92.91	93.58	92.42	93.00
Decision Tree	86.90	89.68	85.11	87.34
Extra Trees Classifier	94.70	95.26	94.27	94.76
Gradient Boosting Classifier	85.25	81.87	88.01	84.83
Adaboost Classifier	85.96	85.50	86.46	85.98

[redacted]

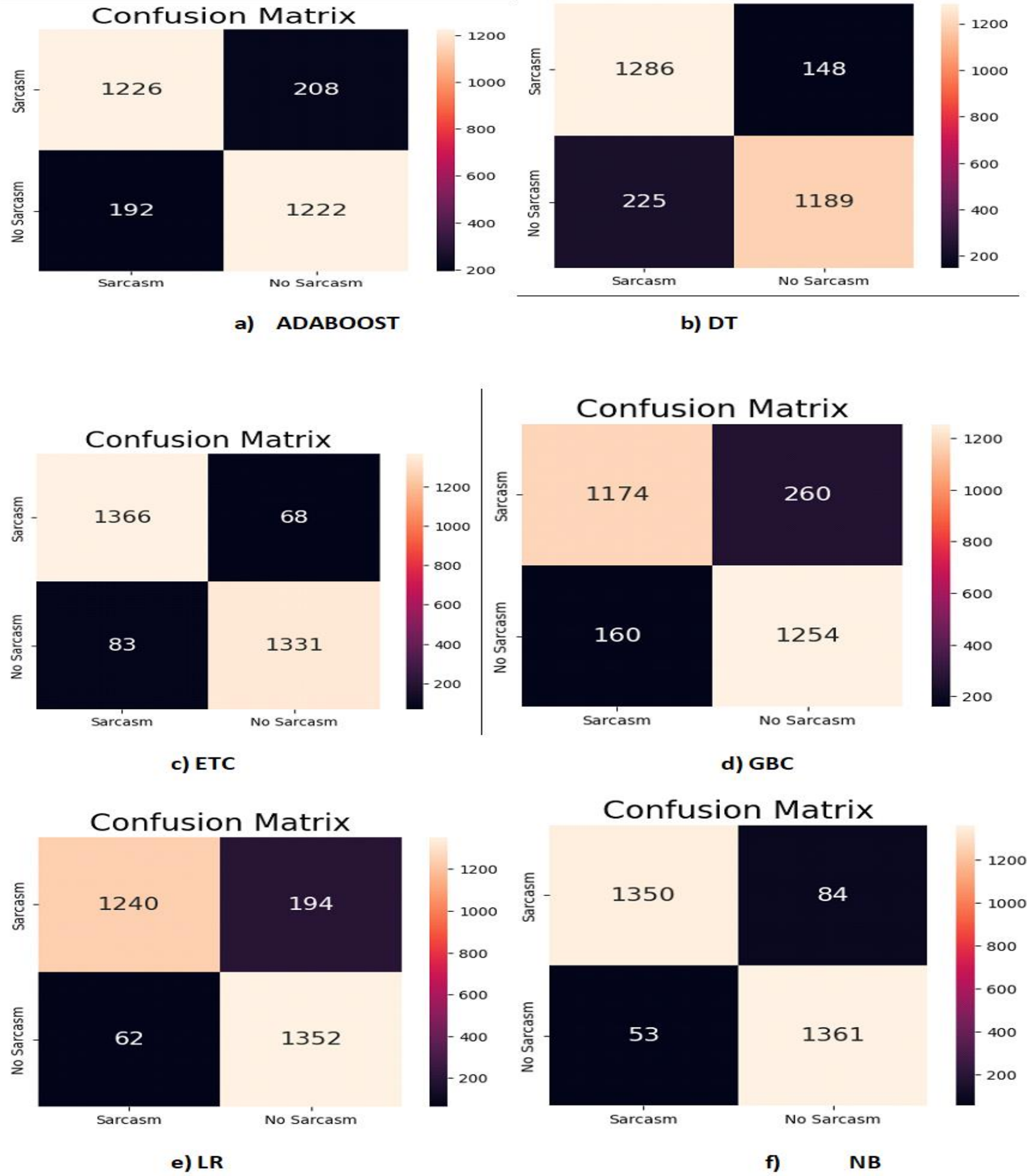
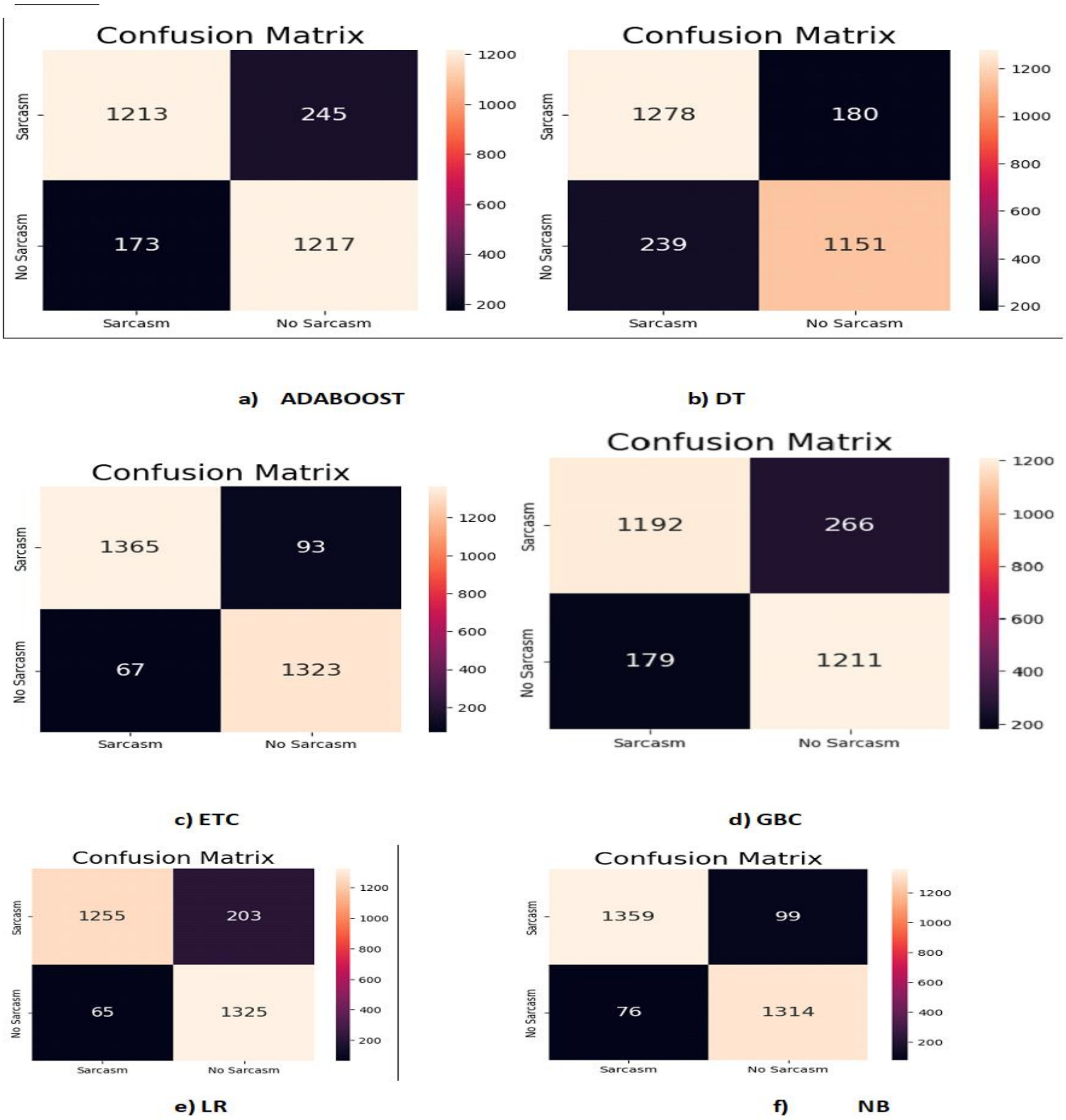


Figure 10. Confusion Matrix of **Wn = 4; Cn =4; WCn =3**

Table 9. Wn = 4 Cn =4 WCn =4

	Accuracy	Precision	Recall	FScore
Logistic Regression	90.59	86.08	95.08	90.36
Naive Bayes	93.86	93.21	94.70	93.95
Support Vector Machine	94.35	93.76	95.13	94.44
Random Forest	92.87	93.48	92.66	93.07
Decision Tree	85.29	87.65	84.25	85.92
Extra Trees Classifier	94.38	93.62	95.32	94.46
Gradient Boosting Classifier	84.38	81.76	86.94	84.27
Adaboost Classifier	85.32	83.20	87.52	85.31



a) ADABOOST

b) DT

c) ETC

d) GBC

e) LR

f) NB

Figure 11. Confusion Matrix of **Wn = 4; Cn =4; WCn =4**

This study considered the following cases for dimensionality reduction because accuracy are almost the same for lower values: (Wn=3,Cn=3,CWn=3), (Wn=3,Cn=3,CWn=4), (Wn=3,Cn=4,CWn=3),(Wn=3,Cn=4,CWn=4),(Wn=4,Cn=3,CWn=3),(Wn=4,Cn=3,CWn=4),(Wn=4,Cn=4,CWn =3),(Wn=4,Cn=4,CWn=4). Table 2 was represented with (Wn=3,Cn=3,CWn=3), & this has the greatest Fscore, which is for Naive Bayes & it is 95.85%. With the values in Table 3 (Wn=3,Cn=3,CWn=4), the Extra Trees Classifier has the highest Fscore (95.57%). Table 4 shows (Wn=3,Cn=4,CWn=3) & contains the Extra Trees Classifier's maximum Fscore of 94.73%. The greatest Fscore for Naive Bayes (95.47%) & Extra Trees Classifier (94.93%) is shown in Tables 5 & 6. Finally.

5. CONCLUSION

Using the approach of data augmentation, this study primarily focuses on achieving a balanced set of data. In order to accomplish this, the study expanded the dataset without compromising the conceptual meaning, & it also expanded the feature set for minority classes while maintaining an awareness of the essence of the sentence that was being written. After applying data augmentation, it is clear from Figure 4 & Table 1 that Naive Bayes with Fscore 95.85% with Wn = 3, Cn = 3, & CWn =3 is the most accurate method.

References:

1. Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., & Yu, S. X. (2019). Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF conference on computer vision & pattern recognition* (pp. 2537-2546).
2. Cao, K., Wei, C., Gaidon, A., Arechiga, N., & Ma, T. (2019). Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, *32*.
3. Bayer, M., Kaufhold, M. A., & Reuter, C. (2022). A survey on data augmentation for text classification. *ACM Computing Surveys*, *55*(7), 1-39.
4. Shorten, C., Khoshgoftaar, T. M., & Furht, B. (2021). Text data augmentation for deep learning. *Journal of big Data*, *8*, 1-34.
5. Wang, S., Zhong, W., Tang, D., Wei, Z., Fan, Z., Jiang, D., ... & Duan, N. (2021). Logic-driven context extension & data augmentation for logical reasoning of text. *arXiv preprint arXiv:2105.03659*.
6. Guo, H. (2020, April). Nonlinear mixup: Out-of-manifold data augmentation for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, No. 04, pp. 4044-4051).
7. Karimi, A., Rossi, L., & Prati, A. (2021). Aeda: An easier data augmentation technique for text classification. *arXiv preprint arXiv:2108.13230*.
8. Wei, J., & Zou, K. (2019). Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.
9. Gupta, V. K. (2019). " Hinglish" Language--Modeling a Messy Code-Mixed Language. *arXiv preprint arXiv:1912.13109*.
10. Dai, H., Liu, Z., Liao, W., Huang, X., Wu, Z., Zhao, L., ... & Li, X. (2023). Chataug: Leveraging chatgpt for text data augmentation. *arXiv preprint arXiv:2302.13007*.

Rajshree Singh
School of Computer Application,
Babu Banarasi Das University,
Lucknow
India
rajshree.singh07@bbdu.ac.in

Dr. Reena Srivastava
School of Computer Application,
Babu Banarasi Das University,
Lucknow
India
dean.soca@bbdu.ac.in