A Sampling-Based Sentiment Analysis of Imbalanced Streamed Movie Reviews

Ary Mazharuddin Shiddiqi
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
ary.shiddiqi@if.its.ac.id

Reza Wahyu Ramadhan
Department of informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
6025222007@mhs.its.ac.id

Gehad Adel Ali Dahman
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
gehaddahman@gmail.com

Zulchair Asyari
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
6025222005@mhs.its.ac.id

Muhammad Rafi' Ramadhani
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
6025222004@mhs.its.ac.id

2023 IEEE Industrial Electronics and Applications Conference (IEACon) | 979-8-3503-4751-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/IEACON57683.2023.10370207

***Abstract*—Sentiment analysis has gained significant importance in analyzing individuals' attitudes and perceptions toward various products, services, and entertainment mediums, including movies. Evaluating the sentiment expressed in movie reviews can provide valuable insights into how users interpret and react to specific films. However, Movie review datasets often suffer from an imbalance in the distribution of positive and negative sentiment labels, which presents challenges for accurate sentiment classification. We propose a framework that harnesses streaming data for enhancing sentiment analysis algorithms. First, we create an initial model using an IMDB movie review dataset to categorize real-time review streams. To address the issue of imbalanced streamed data in movie reviews, we apply diverse sampling techniques, mitigating bias toward the dominant sentiment. This method bolsters the sentiment classifier's effectiveness. Additionally, we iteratively improve the initial model using recorded classification outcomes. We conducted comprehensive experiments on varied movie review datasets to assess our approach's effectiveness. Evaluation metrics were used for comparison, including accuracy, precision, recall, and F1-score. The results encompassed contrasting our sampling-driven method with baseline approaches. The SVC outperformed other algorithms in a native classification environment, whereas the extra tree excelled in a streamed classification environment. These outcomes underscored our framework's efficacy in enhancing sentiment analysis algorithm performance.**

***Index Terms*—sentiment analysis, imbalanced data classification, data streams**

I. Introduction

Sentiment analysis is an emerging field in natural language processing focusing on understanding and categorizing people's opinions and emotions. It is vital in analyzing individuals' attitudes and perceptions towards various products, services, and entertainment mediums like movies. The sentiment analysis process involves evaluating consumers' opinions to determine their positive or negative sentiment [1]. The Internet Movie Database (IMDb) is a popular platform where users share their opinions on movies. Examining the expressions in movie reviews can categorize these opinions as positive, negative, or neutral.

Analyzing the tone of movie reviews can provide valuable insights into how users interpret and respond to different films. Filmmakers, producers, and studios can utilize this information to identify the effective elements of their movies and make necessary modifications to appeal to audiences more effectively. However, there are several challenges in analyzing the sentiment of movies from the IMDb database, including language complexity, subjectivity, context, and dataset quality.

Finding the best accuracy and performance between algorithms has been done by Balakrishnan Gokulakrishnan [2] on a Twitter data stream [3]. They have improved the accuracy of the classifiers they were using, and they compared each algorithm based on their performances. They aim to try sampling and boosting techniques on different datasets for future work.

While their study focused on comparing algorithm performances, further investigation is needed to determine the optimal combination of sampling and boosting techniques for different datasets. This research gap presents an opportunity for future work to evaluate the effectiveness of these techniques and potentially uncover new approaches to improving sentiment analysis accuracy and performance.

To address the abovementioned problem, this study presents a framework that leverages streamed data to refine sentiment analysis algorithms. We compare the performances of several algorithms, namely Support Vector Classifier (SVC), Random Forest, Naive Bayes, Gradient Boosting, Extra Tree, and AdaBoost. These algorithms are applied to an IMDb dataset, which is simulated as a data stream. The primary focus of this research is to evaluate the accuracy and performance of these classifiers in classifying IMDb movie reviews.

II. Literature review

Sentiment analysis identifies opinions, emotions, and values in natural language, a type of subjectivity analysis used in natural language processing studies[4]. It determines whether words or phrases are negative or positive based on the topic and context given.

The study by Tun et al. (2010) [5] introduced a sentiment analysis approach that employed fine-grained analysis to ascertain the sentiment orientation and intensity of movie reviewers, considering multiple aspects of the movie. The authors implemented sentiment classification for various aspects such as overall movie sentiment, director sentiment, story sentiment, cast sentiment, music sentiment, and scene sentiment.

In recent studies, Motz [6] proposed a live sentiment analysis method that utilizes machine learning and text processing algorithms. The approach encompasses trend and sentiment analysis on Twitter API data streams to focus solely on relevant data. The results indicate there is still potential for improvement in this field, as the highest achieved accuracy was 68.29%. This research highlights the importance of combining algorithms to enhance the accuracy and effectiveness of live sentiment analysis.

Furthermore, Satrya [7] conducted a study demonstrating the benefits of combining multiple approaches in sentiment analysis. By combining Machine Learning and Ensemble Learning with a rule-based approach on an Indonesian police-related Twitter dataset, the researchers increased positive sentiment by 0.83% to negative sentiment. This finding emphasizes the advantages of leveraging various techniques in sentiment analysis to achieve more accurate and balanced sentiment classification.

Sai et. al. [8] conducted a comparative study to determine the most effective algorithm for sentiment classification. The study compared the accuracy of various algorithms using two different techniques, namely Bag-of-Words (BOW) and TF-IDF. Among the algorithms tested, it was found that the K-Nearest Neighbors (KNN) algorithm had the lowest accuracy of 54% when applied with the BOW technique. Similarly, for the TF-IDF technique, the Decision Tree classifier achieved the lowest accuracy of 71%.

The challenge of imbalanced datasets in movie reviews remains a significant obstacle in developing accurate classifiers. Dai [9] addressed this issue by employing an oversampling technique known as Word Embedding-based Synthetic Minority Oversampling Technique (WESMOTE). This technique synthesizes new training examples from text based on word embeddings, helping to mitigate the imbalanced data problem. In a related study, Krishna [4] proposed a hybrid Deep Learning approach for sentiment analysis using IMDb movie reviews. The research demonstrated that the hybrid CNN-BiLSTM technique achieved a precision rate of 95%. This highlights the effectiveness of combining Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) models in accurately classifying sentiments expressed in movie reviews.

Algorithm 1 Generating the classification model

```
Require: Dataset D: IMDB Movie Review
  t ← global variable batch process duration
  Split D → training data T and testing data S
  train M ← with T using a classification algorithm A
  while not end do
      if Batch == t then
          get R from Algorithm 3
          merge data TR ← T + R
          re-train M ← with TR using a classification algorithm A
          use M for Algorithm 3
      end if
  end while
```

Algorithm 2 Generating data streams

```
Require: Dataset D: IMDB Movie Review
  while not end do
      X ←get random movie review from dataset
      Remove the class label Y from X
      Feed X and Y into the classification algorithm A
  end while
```

III. Methodology

The research methodology (Figure 1) begins by splitting the IMDb dataset into two portions: 80% of the data is allocated for training purposes (Algorithm 1). A random subset of the data is selected for simulating movie review streams (Algorithm 2) [10], [11]. The model is used to classify upon receiving movie reviews (Algorithm 3). The labeled streamed movie reviews are recorded to enhance the initial model. Once the predictions are generated, the corresponding attributes and predicted labels are added to the training data (Algorithm 1). This augmented dataset is then used to evaluate the performance of the model. The evaluation uses a confusion matrix, which comprehensively assesses the model's performance in terms of correctly and incorrectly classified instances.

A. Data Preprocessing

An imbalance between the minority and majority classes is often observed in movie review datasets. This imbalance can lead to bias during training, causing certain algorithms to overlook the minority class and impacting their results. We employed an oversampling approach to balance the dataset to address this issue and prevent such bias. Oversampling is a technique that aims to rectify imbalanced data by increasing the number of samples in the minority class. This approach ensures that the data information from the majority class is retained while enhancing the representation of the minority class [12]. By utilizing oversampling, we create a more balanced dataset that enables algorithms to effectively capture patterns and make accurate predictions for minority and majority classes.

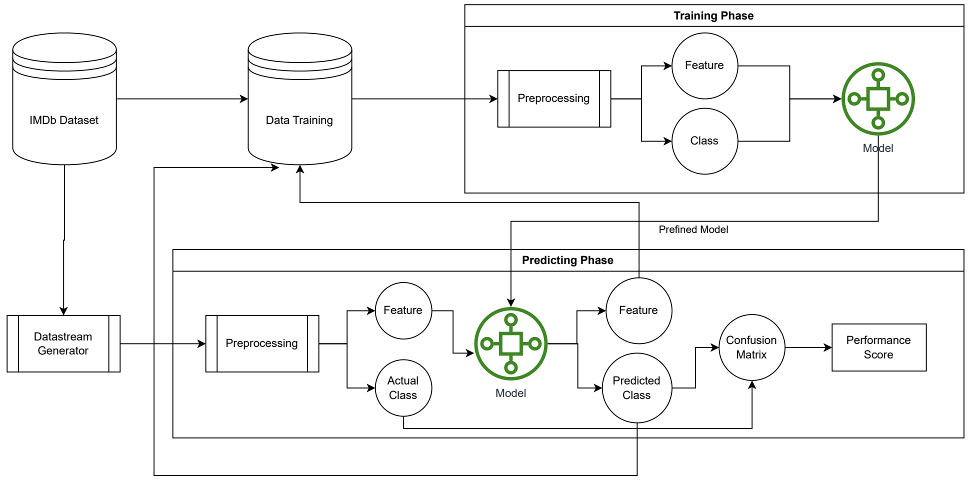

Fig. 1. Data streaming system architecture

Algorithm 3 Classifying data streams

Require: Model **M**
  Input: X and Y from Algorithm 2
  while not end **do**
    $Y' \leftarrow$ Predict class label of X using **M**
    if $Y == Y'$ **then**
      Record **R** $\leftarrow$ concat(X, Y)
      if Batch == t **then**
        pass **R** to Algorithm 1
      end if
    end if
    Evaluate the prediction using ground truth Y
      Calculate Accuracy using Equation 1
      Calculate Recall using Equation 2
      Calculate Precission using Equation 3
      Calculate F1 Score using Equation 4
  end while

B. Building Trained Model

In this paper, we conduct a comparative analysis of sentiment analysis algorithms to evaluate their accuracy and performance in classifying IMDb Movie Reviews. The algorithms that we compare include:

1) Support Vector Machine (SVM): Support Vector Machines (SVM) is a versatile machine learning algorithm for classification and regression tasks. Its strength lies in its ability to handle feature vectors of infinite dimensions, making it a highly powerful and reliable classifier in various applications [1].

2) Random Forest: Random forest is a machine learning algorithm that belongs to the family ensemble methods. The algorithm combines multiple decision trees to form a forest, where each decision tree makes predictions independently.

3) Naive Bayes: The Naive Bayes Classifier is a popular machine learning algorithm for text classification. It operates on the principles of Bayes' theorem and assumes that the features (such as words or terms) are independent given the class label.

4) Gradient Boosting: Gradient Boosting Classifier is a machine learning algorithm belonging to the boosting methods family. It is commonly used for text classification tasks, where the goal is to assign predefined categories or labels to text documents based on their content. Gradient Boosting combines multiple weak classifiers, typically decision trees, sequentially to create a strong classifier.

5) Extra Tree: Extra Trees Classifier, also known as Extremely Randomized Trees, is a machine learning algorithm that belongs to the ensemble learning family. It is a variant of the Random Forest algorithm and can be used for text classification tasks. Similar to Random Forest, the Extra Trees Classifier builds an ensemble of decision trees to make predictions.

6) AdaBoost: AdaBoost (Adaptive Boosting) is a machine learning algorithm belonging to the boosting methods family. The classifier works by iteratively training a sequence of weak classifiers, typically decision trees, and giving more weight to misclassified samples in each iteration.

C. Model refinement

The initial model was constructed using a balanced dataset randomly selected from the IMDB dataset. This balanced dataset constituted 80% of the entire IMDB movie review database. The aim of selecting a balanced dataset was to ensure an equal representation of positive and negative sentiments in the training data. This approach helped prevent any bias from an imbalanced dataset, allowing for a more robust and unbiased initial model.

Then, we stream the movie review by randomly selecting reviews from the database. The initial model was employed to conduct sentiment analysis on movie reviews from the streamed movie review. Throughout the sentiment analysis process, the streamed movie review data are recorded and utilized for the periodic refinement of the analysis model. This iterative refinement process aims to continuously update and improve the model based on the incoming movie review data stream.

Upon receiving a data stream containing movie reviews, it is common to encounter situations where the incoming data is highly imbalanced. This poses a significant challenge in effectively training models to handle such imbalances and make accurate predictions. To handle the imbalanced class problem, we utilized an oversampling, i.e. the Synthetic Minority Over-sampling Technique (SMOTE). This technique generated synthetic samples for the minority class, effectively increasing its representation and creating a more balanced distribution of positive and negative responses.

D. Performance measurement

We evaluate the effectiveness of classification performance by assessing its accuracy, precision, recall, and F1-score based on the classification outcomes. Accuracy represents the ratio of correct predictions and is computed by dividing the sum of true positives and true negatives by the total number of observations (Equation 1). While accuracy provides an overall understanding of the model's performance. It may not reveal potential issues such as a high misclassification rate for a specific class. Therefore, additional measurements such as recall and F1-score are necessary. Recall quantifies the number of correctly predicted positives (Equation 2), while the F1-score combines precision (Equation 3) and recall (Equation 2) to provide a single metric summarizing the model's performance (Equation 4).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$recall = \frac{TP}{TP + FN} \tag{2}$$

$$precision = \frac{TP}{TP + FP} \tag{3}$$

$$F1 - score = 2 \times \frac{precision \times recall}{precision + recall} \tag{4}$$

We also calculate the experiments' mean (Equation 5) and standard deviation (Equation 6). The mean/average of the data indicates the data's central tendency. On the other hand, the standard deviation measures how widely distributed or variable the data are. It assesses how dispersed from the mean the results are. While a high standard deviation means that the data points are more scattered or spread out, a low standard deviation shows that the data points are closely concentrated around the mean.

$$\overline{x} = \frac{1}{n} \sum_{i=i}^{n} x_i \tag{5}$$

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (x_i - \overline{x})^2} \tag{6}$$

IV. Experiment and analysis

This research aims to evaluate the performance of various classification algorithms through two types of experiments. The first experiment involves classification using a fixed set of training and testing data. The algorithms will be trained and tested on a predetermined dataset to assess their performance in this scenario. The second experiment focuses on classification with stream simulation. This involves simulating a data stream where the classification algorithms are continuously tested and updated as new data arrives. This experiment evaluates the algorithms' ability to handle dynamic and evolving data in real-time scenarios. We analyze the performance of each sentiment analysis algorithm based on evaluation metrics, including accuracy, precision, recall, and F1-Score.

A. Dataset

We utilized the IMDB Movie Ratings Sentiment Analysis dataset obtained from Kaggle. The dataset comprised 40,000 data points, with 20,019 labeled negative responses and 19,981 labeled positive responses. Thus, we can consider that this dataset is quite balanced. However, as explained in Subsection III-C, the effect of the streaming process could result in an extremely imbalanced dataset.

B. Experiment 1: Classical model classification

We generate the training dataset using two techniques, i.e. native random sampling and balanced random sampling.

1) Native random sampling: Native random sampling involves randomly selecting samples from the original dataset without considering class balance. This approach provides a representative sample of the data streams but may result in an imbalanced dataset with a skewed distribution of positive or negative sentiments. We collect random data consisting of 1000, 10000, and 20000 rows. The increased number of data allows us to investigate the performance of the classifiers across varying dataset sizes. Hence, we can observe how the classifiers' performance scales with the size and variety of the dataset. Subsequently, we employ a 10-fold cross-validation classification approach to evaluate the algorithms.

The experiment result indicates that the SVC performs better than other algorithms in all performance evaluations (Table I, Table II and Table III). This outcome is caused by the fact that SVC can efficiently divide classes by increasing the distance between instances and the separation hyperplane. Thus, SVC is particularly well-suited for handling unbalanced data where the minority class may have fewer instances. Additionally, SVC's fine-grained analysis enables it to detect subtle sentiment patterns and subtleties. This is crucial in sentiment analysis because the classification of movie reviews may depend on the specific words or phrases used.

2) Balanced random sampling: We employ the balanced random sampling technique to address the class imbalance issue. This technique ensures that the training dataset contains equal samples from each class, thereby creating a more balanced representation of positive and negative sentiments. By utilizing this technique, we aim to improve the model's

TABLE I
RESULT OF 1000 RANDOM SAMPLING DATA AS DATA TRAINING IN NATIVE CLASSIFICATION ENVIRONMENT

Algorithm	Mean				Standard Deviation			
	Accuracy	F1 Score	Recall	Precision	Accuracy	F1 Score	Recall	Precision
Naive Bayes	0.86295	0.862841062	0.862924112	0.864039066	0.003839149	0.003850872	0.003842825	0.003774514
Extra Trees	0.8613000001	0.861293304	0.86129439	0.861354712	0.002198366	0.002199767	0.002198871	0.002186039
Random Forest	0.84305	0.8430401	0.84304447	0.843120694	0.004302615	0.004309101	0.004306074	0.004257551
SVC	**0.899425**	**0.899420692**	**0.899431682**	**0.89950932**	**0.001971674**	**0.001972292**	**0.001970619**	**0.001961815**

TABLE II
RESULT OF 10000 RANDOM SAMPLING DATA AS DATA TRAINING IN NATIVE CLASSIFICATION ENVIRONMENT

	Accuracy	F1 Score	Recall	Precision	Accuracy	F1 Score	Recall	Precision
Naive Bayes	0.862950	0.862841062	0.862924112	0.864039066	0.003839149	0.003850872	0.003842825	0.003774514
Extra Trees	0.860000	0.85999513	0.85999591	0.860038516	**0.001934473**	**0.001937620**	**0.001935857**	**0.001906659**
Random Forest	0.843575	0.843565452	0.84356893	0.843642276	0.005172493	0.005176470	0.005174392	0.005148329
SVC	**0.899425**	**0.899420692**	**0.899431682**	**0.89950932**	0.001971674	0.001972292	0.001970619	0.001961815

TABLE III
RESULT OF 20000 RANDOM SAMPLING DATA AS DATA TRAINING IN NATIVE CLASSIFICATION ENVIRONMENT

Algorithm	Mean				Standard Deviation			
	Accuracy	F1 Score	Recall	Precision	Accuracy	F1 Score	Recall	Precision
Naive Bayes	0.86295	0.862841062	0.862924112	0.864039066	0.003839149	0.003850872	0.003842825	0.003774514
Extra Trees	0.858975	0.858970372	0.858971184	0.859011428	0.002231521	0.002233252	0.002232624	0.002218709
Random Forest	0.843725	0.84371565	0.843720056	0.84379438	0.002427705	0.002428700	0.002429551	0.002426946
SVC	**0.899425**	**0.899420692**	**0.899431682**	**0.89950932**	**0.001971674**	**0.001972292**	**0.001970619**	**0.001961815**

ability to classify sentiments across both classes accurately. We used a similar approach in this experiment as in the prior one. We randomly selected 1000, 10000, and 20000 rows from a simulated data stream to collect data samples from. These data samples were used as the dataset for our analysis. The experimental results demonstrate that the SVC outperforms other algorithms in all performance evaluations, reaffirming the findings from the previous experiment (Table I, Table II and Table III).

C. Experiment 2: Datastream classification

We begin this experiment by using 80% of the dataset to initialize the model. The dataset is randomly selected, ensuring balanced positive and negative classes. As the experiment progresses, new data is continuously added to the dataset, with the range of 1 to 100 data points being introduced every minute. After 15 minutes, the system is triggered to re-train the initial model based on each individual data point that we attempt to classify. This ensures that the model is continuously updated and refined to adapt to the evolving data stream. By remodeling the training data in this manner, we aim to enhance the model's accuracy and performance in sentiment classification.

We calculated the mean (Table VII), variance (Table IX), and standard deviation (Table VIII) of the accuracy, F1 score, recall, and precision for each classifier. The Extra Tree Classifier consistently achieved the highest accuracy, F1 score, recall, and precision values. However, the mean alone may not adequately represent the data distribution of accuracy. We also calculated the standard deviation and variance to better understand the data distribution for accuracy, F1 score, recall, and precision to address this limitation.

Based on the experiment results, the Extra Tree Classifier had the highest mean values. It showed relatively low standard deviation and variance, indicating a more stable and consistent performance across the evaluation metrics. These results conclude that the Extra Tree Classifier performs better in accuracy, F1 score, recall, and precision than other classifiers. Additionally, the low standard deviation and variation of its performance further reinforce its superiority in sentiment analysis tasks on the evaluated dataset.

V. CONCLUSION

Our study focuses on analyzing sentiment in movies based on imbalanced streamed reviews. We propose a technique that effectively refines the initial model to adapt to changing sentiment trends in the reviews. A comprehensive evaluation of various classification models shows that the Support Vector Classifier (SVC) performs superior in the native data environment. However, the Extra Trees classifier outperforms other models in a streamed environment. A future work of our proposed framework can explore methods to optimize the utilization of computational resources in sentiment analysis. This may involve developing lightweight algorithms or designing resource-aware models that balance accuracy and resource consumption. The enhancement of our proposed method can be developed based on the related works in the area.

TABLE IV Result of 1000 balanced random sampling data as data training in native classification environment

Algorithm	Mean				Standard Deviation			
	Accuracy	F1 Score	Recall	Precision	Accuracy	F1 Score	Recall	Precision
Naive Bayes	0.863525	0.863427076	0.86350045	0.86450621	0.003813422	0.003824676	0.003816954	0.003747296
Extra Tree	0.859375	0.85937381	0.85937594	0.859387791	0.006733777	0.006734824	0.006733774	0.00672407
Random Forest	0.842325	0.84231578	0.842319531	0.842389079	0.00380645	0.003815222	0.003810106	0.003744666
SVC	**0.90065**	**0.900640367**	**0.90065994**	**0.900829298**	**0.002868743**	0.002870403	**0.002866899**	**0.00284218**

TABLE V Result of 10000 balanced random sampling data as data training in native classification environment

Algorithm	Mean				Standard Deviation			
	Accuracy	F1 Score	Recall	Precision	Accuracy	F1 Score	Recall	Precision
Naive Bayes	0.863525	0.863427076	0.86350045	0.86450621	0.003813422	0.003824676	0.003816954	0.003747296
Extra Tree	0.860975	0.860971289	0.860972304	0.861006681	0.004139558	0.004137167	0.004139132	0.004165179
Random Forest	0.84515	0.845135646	0.845142327	0.845258638	0.005092427	0.005093073	0.005093051	0.005094881
SVC	**0.90065**	**0.900640367**	**0.90065994**	**0.900829298**	**0.002868743**	**0.002870403**	**0.002866899**	**0.00284218**

TABLE VI Result of 20000 balanced random sampling data as data training in native classification environment

Algorithm	Mean				Standard Deviation			
	Accuracy	F1 Score	Recall	Precision	Accuracy	F1 Score	Recall	Precision
Naive Bayes	0.863525	0.863427076	0.86350045	0.86450621	0.003813422	0.003824676	0.003816954	0.003747296
Extra Tree	0.860125	0.860123514	0.860123178	0.860134919	0.002489038	0.00248809	0.002488169	0.002497265
Random Forest	0.843725	0.843717217	0.843720313	0.84377834	0.004385167	0.004396243	0.004389004	0.004304175
SVC	**0.90065**	**0.900640367**	**0.90065994**	**0.900829298**	**0.002868743**	**0.002870403**	**0.002866899**	**0.00284218**

TABLE VII Mean of the accuracy in streamed classification environment

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-score**
svc	0,9481777	0,9518184	0,9481777	0,9483182
random forest	0,9681520	0,9702490	0,9681520	0,9681852
nb	0,8959449	0,9034277	0,8959449	0,8958279
gradient boosting	0,8009341	0,8166880	0,8009341	0,7996701
extra tree	**0,9717636**	**0,9736189**	**0,9717636**	**0,9717993**
adaboost	0,8091311	0,8212302	0,8091311	0,8088404

TABLE VIII Standard Deviation of the acccuracy in streamed classification environment

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-score**
svc	0,0497305	0,0455899	0,0497305	0,0491354
random forest	0,0365329	0,0341241	0,0365329	0,0368645
nb	0,0752644	0,0733693	0,0752644	0,0762665
gradient boosting	0,0878767	0,0883729	0,0878767	0,0902029
extra tree	**0,0313473**	**0,0280731**	**0,0313473**	**0,0312633**
adaboost	0,0879722	0,0893127	0,0879722	0,0894462

TABLE IX Variance of the accuracy in streamed classification environment

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-score**
svc	0,0024731	0,0020784	0,0024731	0,0024142
random forest	0,0013346	0,0011644	0,0013346	0,0013589
nb	0,0056647	0,0053830	0,0056647	0,0058165
gradient boosting	0,0077223	0,0078097	0,0077223	0,0081365
extra tree	**0,0009826**	**0,0007881**	**0,0009826**	**0,0009773**
adaboost	0,0077391	0,0079767	0,0077391	0,0080006

References

[1] Y. Al Amrani, M. Lazaar, and K. E. El Kadiri, “Random forest and support vector machine based hybrid approach to sentiment analysis,” *Procedia Computer Science*, vol. 127, pp. 511–520, 2018.

[2] B. Gokulakrishnan, P. Priyanthan, T. Ragavan, N. Prasath, and A. Perera, “Opinion mining and sentiment analysis on a twitter data stream,” IEEE Xplore, p. 182–188, 12 2012. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6423033/

[3] A. M. Shiddiqi, D. H. Fudholi, and A. Zahra, “A social media analytics on how people feel about their cancelled homecoming during covid-19 pandemic,” in *2021 Fourth International Conference on Vocational Education and Electrical Engineering (ICVEE)*, 2021, pp. 1–6.

[4] M. M. Krishna, B. Duraisamy, and J. Vankara, “Hybrid deep learning techniques for sentiment analysis on imdb datasets,” in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*. IEEE, 2022, pp. 689–693.

[5] T. T. Thet, J.-C. Na, and C. S. Khoo, “Aspect-based sentiment analysis of movie reviews on discussion boards,” *Journal of Information Science*, vol. 36, pp. 823–848, 11 2010.

[6] A. Motz, E. Ranta, A. S. Calderon, Q. Adam, F. Alzhouri, and D. Ebrahimi, “Live sentiment analysis using multiple machine learning and text processing algorithms,” *Procedia Computer Science*, vol. 203, pp. 165–172, 2022.

[7] W. F. Satrya, R. Aprilliyani, and E. H. Yossy, “Sentiment analysis of indonesian police chief using multi-level ensemble model,” *Procedia Computer Science*, vol. 216, pp. 620–629, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050922022554

[8] L. Sai, K. Yasaswi, V. Rakesh, M. Varun, M. Yeswanth, and J. S. Kiran, “Comparative study of algorithms for sentiment analysis on imdb movie reviews,” 03 2023.

[9] H.-J. Dai and C.-K. Wang, “Classifying adverse drug reactions from imbalanced twitter data,” *International Journal of Medical Informatics*, vol. 129, pp. 122–132, 09 2019.

[10] M. M. Gaber and A. M. Shiddiqi, “Distributed data stream classification for wireless sensor networks,” in *Proceedings of the 2010 ACM Symposium on Applied Computing*, ser. SAC ’10. New York, NY, USA: Association for Computing Machinery, 2010, p. 1629–1630. [Online]. Available: https://doi.org/10.1145/1774088.1774439

[11] A. Cuzzocrea, M. M. Gaber, and A. M. Shiddiqi, “Distributed classification of data streams: An adaptive technique,” in *Big Data Analytics and Knowledge Discovery*, S. Madria and T. Hara, Eds. Cham: Springer International Publishing, 2015, pp. 296–309.

[12] Z. Wei, L. Zhang, and L. Zhao, “Minority-prediction-probability-based oversampling technique for imbalanced learning,” *Information Sciences*, vol. 622, pp. 1273–1295, 2023.