SentiMLBench: Benchmark Evaluation of Machine Learning Algorithms for Sentiment Analysis

Anuradha Yenkikar[1], C. Narendra Babu[2]
[1]Department of Computer Science and Engineering, Ramaiah University of Applied Sciences, Bengaluru, Karnataka, India and Pune Institute of Computer Technology, Savitribai Phule Pune University, Pune, India
[2]Department of Computer Science and Engineering, Ramaiah University of Applied Sciences, Bengaluru, Karnataka, India

Article Info

Article history:

Received Dec 18, 2022
Revised Mar 12, 2023
Accepted Mar 26, 2023

Keywords:

Twitter
Sentiment Anaysis
Machine Learning
Ensemble model
Neural Network

ABSTRACT

Sentiment Analysis has been a topic of interest for researchers due to its increasing usage by Industry. To measure end-user sentiment., there is no clear verdict on which algorithms are better in real-time scenarios. A rigorous benchmark evaluation of various algorithms running across multiple datasets and different hardware architectures is required that can guide future researchers on potential advantages and limitations. In this paper, proposed SentiMLBench is a critical evaluation of key ML algorithms as standalone classifiers, a novel cascade feature selection (CFS) based ensemble technique in multiple benchmark environments each using a different twitter dataset and processing hardware. The best trained ensemble model with CFS enhancement surpasses current state-of-the-art models, according to experimental results. In a study, though ensemble model provides good accuracy, it falls short of neural networks accuracy by 2%. ML algorithms accuracy is poor as standalone classifiers across all three studies. The supremacy of neural networks is further stamped in study three where it outperforms other algorithms in accuracy by over 10%. Graphical processing unit provide speed and higher computational power at a fraction of a cost compared to a normal processor thereby providing critical architectural insights into developing a robust expert system for sentiment analysis.

Corresponding Author:

Anuradha Yenkikar,
Research Scholar at Department of Computer Science and Engineering,
Ramaiah University of Applied Sciences, Bengaluru-560058,
Karnataka, India.
Email: anu.jamkhande@gmail.com

1. INTRODUCTION

Social media has completely changed how people communicate. Social network data is useful for analyzing user perspectives, such as gauging user reactions to newly released products, examining how people reacted to a change in government policy, or examining how much people are enjoying an ongoing event. This data would be difficult and potentially expensive to manually sort through. For instance, Twitter is a well-known and rapidly expanding platform where users share text messages known as tweets. Tweets allow users to express their ideas and viewpoints on a specific subject.

A method for determining and categorising the polarity of a text is sentiment analysis (SA). Web content can be broken down into three levels based on its level of granularity: documents, sentences, and words. A character level feature extraction technique is applied at the fourth level, as illustrated in Table 1 [1].

Sentiment analysis is a validated technology with applications in e-commerce, healthcare monitoring, election campaigns, and social event planning, to mention a few. Interest in sentiment analysis has increased as a result of the requirement to analyze and organize unstructured data derived from social media that contains hidden information [2]. Businesses can profit from keeping track of consumer feedback on their goods, while consumers can profit from choosing the best product based on the public opinion. However, the following are the most significant obstacles in Twitter sentiment analysis:

- Tweets are typically written in a colloquial style;
- Short messages provide few cues about sentiment;
- Acronyms and abbreviations are commonly used on Twitter;
- There are no comprehensive benchmark comparison metrics on the various ML and DL algorithms; and
- As the number of tweets grows, there is a lack of robust and fast models for real-time sentiment processing.

Table 1. Classification of web text based on granularity

Level	Delimiter	Granularity Depth	Multiplicity of sentiments	Interpretation of sentiments
1. Document	‘\n’ Newline character	Overall perception at higher level	One opinion of numerous entities	A document's overall feeling
2. Sentence	‘.’ Period character	Polarity of facts in each sentence	Various opinions of various entities	Categorization based on subjectivity
3. Entity or aspect level	‘ ‘ or named entities	The target entities are words at the finest level.	A solitary entity's one opinion	Sentiment, target in a two-tuple
4. Character level	Special characters and ‘ ’ are not used.	Embedding of characters at the micro level	Various viewpoints around a single word entity	Word extraction via morphology

The two types of methodologies used in sentiment analysis are lexicon-based [3] and ML-based [4]. Lexicon or corpus-based techniques: These methods, which relate to methodologies of sentiment classification, are based on Decision Trees (DT) and include k-Nearest Neighbors (KNN), Conditional Random Field (CRF) and Hidden Markov Model (HMM).

Machine learning based techniques: The sentences and aspect levels are extracted to implement these kinds of approaches. Parts-of-speech (POS) tags, n-grams, bi-grams, unigrams, and Bag-of-Words (BoW) are among the features used.

In this paper, we implement and compare most popular unsupervised ML algorithms as base classifiers on the CPU in our first study. This includes an implementation of the ensemble technique to compare its performance with individual base classifiers using the Crowdflower dataset (Data world) sentiment analysis in text (D1.1) comprising of 40,000 tweets and 3-classes, the SemEval-2017 Task 4A (D1.2), 4B (D1.3) and 4C (D1.4). In the second study, we evaluate six popular ML algorithms using D2 (*#newsfeed*) dataset as the number of tweets are increased from 60,000 to 160,000 on CPU to study the scaling characteristics of the algorithms. The role of GPU on classifier performance is explored as part of third study which involves evaluation of six algorithms including a convolution neural network (CNN) using D3 (*#politics*) dataset as the number of tweets increase from 100,000 to 300,000 on both CPU and GPU. The research is towards recommending specifications for a robust expert system for real-time tweet sentiment analysis. The experiments and benchmark comparison will serve as a guide to other researchers working in this domain to choose the most optimal algorithms while designing expert systems for sentiment analysis.

The next section of the paper discusses the literature review followed by methodology. Further, results obtained from the various studies and their details are discussed. This is followed by conclusion and references in the end.

2. RESEARCH METHOD

Supervised, semi-supervised, and unsupervised machine learning are the three fundamental subcategories. This approach is perfect as it can handle vast amounts of data and is automated. For sentiment prediction and optimization, ML algorithms including Naive Bayes (NB), DT, Regression, and Support Vector Machine (SVM) have typically been utilised to address the issue of sentiment classification on Twitter [5]. While SVM and Multinomial NB have been demonstrated to perform better in terms of accuracy and optimization. Hierarchical ML approaches only perform somewhat well in classification tasks [6].

Few researchers [7]-[9] have successfully merged the aforementioned techniques in an ensemble model to predict sentiment obtaining an accuracy of 88% on movie review dataset. For Twitter sentiment analysis, majority voting is the most utilized ensemble classification approach. Also, there hasn't been much coverage of the usage of modern hardware architecture to speed up real-time sentiment analysis on Twitter. Authors in [10], [11] used SVM with Radial basis kernel function and NB respectively for sentiment analysis tasks and they could achieve an accuracy of only 82%. It is similar to the performance achieved by [12] who used SVM, NB and KNN for sentiment detection in Chinese documents. Authors in [13] achieved an accuracy of 86% using SVM, NB and maximum entropy algorithms. When compared to other ML approaches, SVM and NB have shown to perform better on benchmarks.

Table 2. Benchmark summary of CNN based Sentiment Analysis

Author(/s). year of publication	Purpose	Dataset	Results
Islam J and Zhang Y. 2016 [14]	Visual SA	1269 images from twitter	The performance advantage of GoogleNet over AlexNet was roughly 9%.
Severyn A and Moschitti A. 2015 [15]	Phrase level and message level task SA	Semeval-2015	Compared to the official system, rated first in the phrase level subtask and second in the message level subtask..
Yanmei L and Yuda C. 2015 [16]	Micro-Blog SA	1000 microblog comments	The suggested model can significantly increase the validity and accuracy of emotional orientation.
You Q, et al. 2015 [17]	Textual-visual SA	Getty Images, 101 keywords	Early single fusions are outperformed by the joint visual and textual model.
Ouyang X, et al. 2015 [18]	Sentiments of Sentences	rottentomatoes.co (movie reviews)	The suggested model performed better than the prior models, with an accuracy rate of 45.5%.

In recent years, sentiment analysis methods based on neural network architectures have become more common. The classification of brief text messages from Twitter using the CNN proposed by the authors in [19] has a higher accuracy of about 86%. Authors in [20] proposed a CNN model for improving the sentiment analysis and emotion recognition task and achieved an impressive accuracy of around 96%. More summary of similar benchmarks is listed in Table 2. With published sentiment prediction techniques that make use of deep CNN and recursive neural networks, it might be challenging to accurately capture the compositionality of words.

Researchers have recently embraced contemporary Transformer Neural Network models, which have excelled in numerous Natural Language Processing (NLP) applications. In addition to discussing issues with linguistic styles for sentiment analysis and NLP, [21] provides a thorough survey of "text representation models from the beginning to the present. There are 22 datasets from various domains, five classification techniques, and the well-known Bi-Directional Encoder Representation from Transformers (BERT) architecture. In [22], [23], authors suggested utilising DL techniques like BERT to evaluate Indian Covid-19 tweets during the lock down and post pandamic. BERT was used to investigate several emotions, and the outcomes were contrasted with those obtained from conventional Logistic Regression (LR), SVM, and Long Short-term Memory (LSTM) models. The BERT model, LR, SVM, and LSTM each have accuracy rates of 89%, 75%, 74.75%, and 65%, respectively. In order to improve the accuracy of text-based psychological analysis of online comments, authors in [24] created a hybrid model (BERT-BiLSTM-TextCNN). In this model, BERT generates word vectors while BiLSTM and TextCNN capture local correlation.

An attention-based bidirectional CNN-RNN deep model (ABCDM)" proposed in [25]. The present and the future taken into account in two layers of the BiLSTM and GRU. In order to simultaneously stress different words, the output of the ABCDM bidirectional layer is subjected to an attention model. In contrast, BiLSTM needs more time to train and may need extra hardware, such a GPU, to speed up the process. Authors in [26] examined time series data, projected stock price based on stock transaction history, and evaluated text sentiments using BERT and LSTM.

3. EXPERIMENTL STUDY

The experimental study is split into three parts based on the datasets used for evaluation, the techniques, and the hardware architecture. They are:

3.1. Study 1

We use the Semantic Relational Machine Learning (SRML) model developed by [27] for comparative performance evaluation of classifiers both as base classification techniques and implement the ensemble classifier. Four datasets as shown in Table 3 are used for evaluation, namely the Crowdflower dataset (Data world) sentiment analysis in text (D1.1) comprising of 40,000 tweets and 3-classes, the SemEval-2017 Task 4A (D1.2), 4B (D1.3) and 4C (D1.4).

Table 3. Study 1 with dataset D1 statistics

Dataset title and details	Class	Strongly negative	Negative	Neutral	Positive	Strongly positive	Total
D1.1: Crowdflower dataset (Data world) sentiment analysis in text [28]	3	-	15236	9465	15299	-	40000
D1.2: SemEval-2017, Task 4A [29]	3	-	3231	10342	7059	-	20632
D1.3: SemEval-2017, Task 4B [29]	2	-	2339	-	8212	-	10551
D1.4: SemEval-2017, Task 4C [29]	5	138	2339	10081	7830	382	20632

3.2. Study 2

Here, the dataset used for evaluation is the twitter (#newsfeed) dataset (D2) to evaluate the classifier performance. Accuracy is checked for 60K and 0.16M tweets on CPU-only to check scaling performance. ML algorithms used are LOGR, SVM, DT, RF, DT-GB and NB.

3.3. Study 3

The Dataset used for evaluation is the twitter (#politics) dataset (D3). We use Hashtags # 'cricket','dhoni','modi','BJP','Rahulgandhi','congress','Politics',’kohli’. The algorithms used for comparison are DT, SVM, KNN-1,3,5 and CNN. This study is further divided into 2 parts:

- First, we study the effect of increase in the number of tweets on the algorithms to evaluate their scaling performance on a CPU.
- We next study the performance of all classifiers on a GPU.

Python is used to implement Study 1. For feature representation, categorization, calculating similarity, and assessment, Scikit-learn is utilised. During pre-processing of data, stemming and stop word removal are performed using the Natural Language Toolkit. A dataset management tool is called Pandas. NumPy is a Python library for manipulating multi-dimensional arrays. To compare the performance of these algorithms, Study 2 and Study 3 are created in Google Collaboratory, a cloud-based notebook environment that enables Google Drive users to write, execute, and share code. For our GPU test, the platform provides a free instance of the NVidia Tesla K40, 12GB GPU.

A cross section of the three datasets is shown in Table 4 and the methodology adopted in each of the three studies is described in the sections below.

Table 4. Cross section of datasets used in the three studies

Study 1: Crowdflower dataset (D1.1, Data world) sentiment analysis in text [28]

tweet_id	sentiment	author	content
1956967341	empty	xoshayzers	@tiffanylue i know i was listenin to bad habit earlier and i started freakin at his part
1956967666	sadness	wannamama	Layin n bed with a headache ughhhh...waitin on your call...
1956967696	sadness	coolfunky	Funeral ceremony...gloomy friday...
1956967789	enthusiasm	czareaquino	wants to hang out with friends SOON!
1956968416	neutral	xkilljoyx	@dannycastillo We want to trade with someone who has Houston tickets, but no one will.
1956968477	worry	xxxPEACHESxxx	Re-pinging @ghostridah14: why didn't you go to prom? BC my bf didn't like my friends
1956968487	sadness	ShansBee	I should be sleep, but im not! thinking about an old friend who I want. but he's married now. damn, &amp; he wants me 2! scandalous!
1956968636	worry	mcsleazy	Hmmm. http://www.djhero.com/ is down
1956969035	sadness	nic0lepaula	@charviray Charlene my love. I miss you

D1.2: SemEval-2017, Task 4A [29], D1.3: SemEval-2017, Task 4B [29] and D1.4: SemEval-2017, Task 4C [29]

tweet_id	sentiment	content
628949369883000832	negative	dear @Microsoft the newOoffice for Mac is great and all, but no Lync update? C'mon.
628976607420645377	negative	@Microsoft how about you make a system that doesn't eat my friggin discs. This is the 2nd time this has happened and I am so sick of it!
629023169169518592	negative	I may be ignorant on this issue but... should we celebrate @Microsoft's parental leave changes? Doesn't the gender divide suggest... (1/2)
629226490152914944	positive	Microsoft, I may not prefer your gaming branch of business. But, you do make a damn fine operating system. #Windows10 @Microsoft
629650766580609026	positive	Just ordered my 1st ever tablet; @Microsoft Surface Pro 3, i7/8GB 512GB SSD. Hopefully it works out for dev to replace my laptop =)
630159517058142208	positive	Sunday morning, quiet day so time to welcome in #Windows10 @Microsoft @Windows http://t.co/7VtvAzhWmV
630807124872970240	neutral	@spyderharrison @Microsoft the reason I ask is because it may be the manufacturer's fault, and they could help you.
630909171437801472	neutral	OK this is my pure speculation. @Microsoft owns the cloud compute tech. @Cloudgine is utilizing the tech. 3rd party devs is open to use.
630982270409572352	neutral	We are still taking registrations for our Education Technology Update with @Acer and @Microsoft on August 28! Visit: http://t.co/lQvTHE6Chb

Study 2: The twitter (#newsfeed) dataset (D2)

tweet_id	sentiment	content
586266658731388929|	positive	RT @JohnGGalt: Amazing—after years of attacking Donald Trump the media managed to turn #InaugurationDay into all about themselves. #MakeAme…
586260160462589954|	positive	RT @vooda1: CNN Declines to Air White House Press Conference Live YES! THANK YOU @CNN FOR NOT LEGITIMI…
586238751334125569|	positive	RT @Muheeb_Shawwa: Donald J. Trump's speech sounded eerily familiar... POTUS plans new deal for UK as Theresa May to be first foreign leader to meet new president since inauguration
586159308745920512|	negative	RT @Slate: Donald Trump's administration: "Government by the worst men."
585917217696538625|	negative	RT @RVAwonk: Trump, Sean Spicer, etc. All lie for a reason. Their lies are not just lies. Their lies are authoritarian propaganda.
	Negative	RT @tony_broach: Chris Wallace on Fox news right now talking crap about Donald Trump news conference it seems he can't face the truth either…

Study 3: the twitter (#politics) dataset (D3)

tweet_id	content
1302204564267917313	RT @Niraj210171: When you vote out congress in hope that BJP gov would give us jobs and employments and when BJP does the same thing...
1302204564418953216	RT @fredhamilton: @jtiku @SortedEagle @MaheshJ95622388 modi is INCOMPETENT\n42 YR WORST ECONOMY (Pre -Covid)
1302204564490186753	RT @N_M5001: #RRBExamDates\n#speakup\n#SpeakUpforSSCRailwaysStudends\n\n5 millions tweets and still not a single BJP leader came to speak to us\u2026
1302204564888715264	RT @Nagesh_nsui6: The youth needs job not Modi ji\u2019s bhaashan. \nActions should speak louder than words.
1302204571276640256	RT @srivatsayb: Congress in our 2019 Manifesto promised to fill 24 lakh vacant govt jobs.\n\nBJP Govt has not only stopped hiring &amp; exams, it\u2026

4. METHODOLOGY

Methodologies used as part of the three studies are explained in this section.

4.1. Study 1: SRML model using D1 datasets

The architecture of the SRML model used in study 1 is shown in Figure 1. As shown, a sizable Twitter review dataset (D1) containing two class polarities and several emotions is downloaded from https://www.crowdflower.com/data-for-everyone. Following pre-processing, Word2Vec feature extraction is performed on the data. However, when standard Word2Vec is used, even words that have no bearing on sentiment classification will be embedded. As a result, Word2Vec is used in the suggested method together with a CBOW configuration to estimate the 1-norm and 2-norm features. SentiWordNet 3.0 is used to process the retrieved features for weighing [30]. Here, it is seen that many terms are irrelevant to the classification of sentiment and can be eliminated to save computational cost and errors. We offer a unique cascade feature selection (CFS) method that combines the Wilcoxon rank sum test, the ULR-based significant predictor test, and the cross-correlation test to address this issue. The whole model flow-chart is shown in Figure 2.

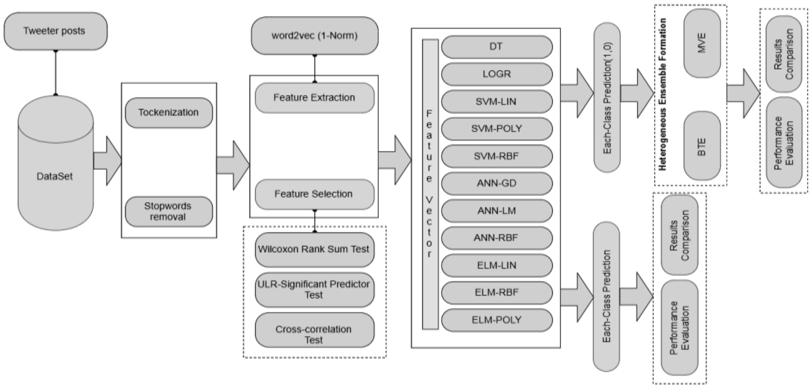


Figure 1. SRML model architecture for multi-class sentiment analysis under study 1

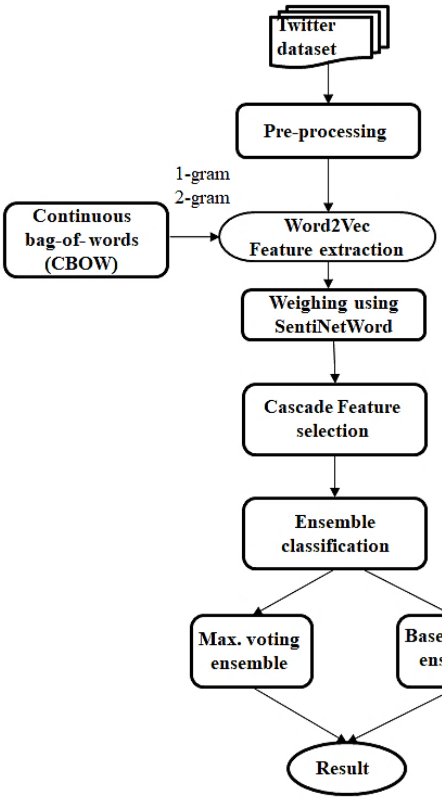


Figure 2. SRML flow-chart

4.2. Sentiment classification using base classifiers

The nine base classifies used in SRML benchmark study are: SVM, DT, LOG-R, ANN-GD, ANN-LM, CNN, KNN-1, 3, 5 and the proposed Ensemble technique which are explained below.

4.2.1. Decision Tree (DT)

The C5.0 model of the DT classifier is used to predict the task given and supplied user's input, and it performs recursive partitioning across extracted datasets. Each node of the tree, starting at the root, divides the feature vector into various branches according to an association rule between the split criteria.

4.2.2. Logistic Regression (LOGR)

The algorithm performs classification of the dependent variable by applying regression to the independent and dependent variables. In this instance, LOGR has been used to develop a prediction method and derived as in (1) and (2).

$$logit[\pi(x)] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_m X_m \tag{1}$$

LOGR returns $\pi(x)$ in (2) as:

$$\pi(x) = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots\ldots+\beta_m X_m}}{1+e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots\ldots+\beta_m X_m}} \tag{2}$$

The parameter values of the LOGR classifiers are tuned as: *C = .01, max_iter = 100.*

4.2.3. Support Vector Machine (SVM)

SVM is a binary linear non-probabilistic classifier that makes use of the data's pattern. The SVM uses the function to predict:

$$Y' = w * \phi(x) + b \tag{3}$$

In (3), regression risk is decreased to retrieve Y'.

$$R_{reg}(Y') = C * \sum_{i=0}^{l} \gamma(Y_i' - Y_i) + \frac{1}{2} * \|w\|^2 \tag{4}$$

where,

$$w = \sum_{j=1}^{l}\left(\alpha_j - \alpha_j^*\right)\phi(x_j) \tag{5}$$

In (5), α and α* state the relaxation parameter. The output obtained in (6) as,

$$Y' = \sum_{j=1}^{l}\left(\alpha_j - \alpha_j^*\right)\phi\left(x_j\right) * \phi(x) + b$$

$$= \sum_{j=1}^{l}\left(\alpha_j - \alpha_j^*\right) * K\left(x_j, x\right) + b \tag{6}$$

In (6), $K\left(x_j, x\right)$ states the kernel function. The SVM classifiers used in our research are set with the following parameter: *C = 0.1, kernel = linear.*

4.2.4. K-nearest neighbors (KNN)

KNN is a classification technique that uses the complete data set to categorize new data, therefore there is no training phase. The algorithm determines the separation between each new data point and each other point in the dataset when a new data point is supplied. Then it determines how many nearest neighbors there are in the data set based on the K value, which in our case is one, three, or five,

if K=1, then it uses the shortest distance between each point to assign it to the same category as the data point with the shortest distance.

if *K>1*, then it takes a list of *K* minimum distances of all data points.

4.2.5. Artificial Neural Network (ANN)

The three layers that make up the traditional ANN architecture are input I_h, hidden, and output O_h.

$$O_h = \frac{1}{1+e^{-I_h}} \tag{7}$$

Additionally, the results of ANN learning will be displayed in (8).

$$O_o = \frac{1}{1+e^{-O_i}}. \tag{8}$$

ANN iteratively lowers error value to provide correct classification. The error function is calculated mathematically using the formula in (9).

$$MSE = \frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} (\mathrm{y_i}' - \mathrm{y_i})^2 \tag{9}$$

In this work, ANN is applied with two different kernel functions which are:

4.2.6. Artificial Neural Network-Gradient Descent (ANN-GD)

As already mentioned, ANN aims to progressively decrease error function for all training sets.

$$w^{*} = argmin\ L(w) \tag{10}$$

$$\mathrm{L(w)} = \sum_{\mathrm{t=1}}^{\mathrm{N}} \mathrm{L}\big(\mathrm{y_t}, \mathrm{f_w(x_t)}\big) + \lambda \mathrm{R(w)} \tag{11}$$

According to our research and as shown in (10), ANN-GD aims to use the GD algorithm to get a local optimum for (12) where $f_w(x)$ is non-linear in the weight vector w. Here, GD updates w iteratively by replacing w_t by w_{t+1}using (11, 12).

$$w_{t+1} = w_t - \eta_t \nabla L \tag{12}$$

$$w_{j,t+1} = w_{j,t} - \eta_t \frac{\partial L}{\partial w_j} \tag{13}$$

In (12, 13), η_t denotes the learning rate, which typically falls off as t increases. Here, ∇L states the error value as per (14),

$$\frac{1}{\mathrm{n}} \sum_{\mathrm{i=1}}^{\mathrm{n}} (\mathrm{y_i}' - \mathrm{y_i})^2 \tag{14}$$

4.2.7. Artificial Neural Network–Levenberg-Marquardt (ANN-LM)

In contrast to earlier NN models, LM-ANN learn quickly and are both computationally and time efficient. The weight-update function is given in (15).

$$W_{j+1} = W_j - \left(J_j^T J_j + \mu I\right)^{-1} J_{j^{e}j} \tag{15}$$

4.2.8. Convolution Neural Network (CNN)

The ‘convolutional’ CNN filter, or ‘kernel,’ extracts the crucial details of the image as it goes through it. It has been frequently used on image datasets. As depicted in Figure 3, the network in this study was composed layers.

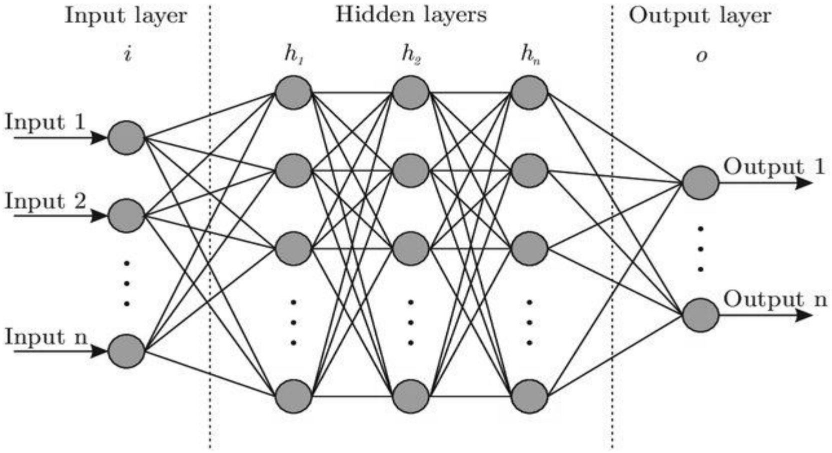


Figure 3. ANN architecture used in SRML

4.2.9. Ensemble Learning

ML technique called ensemble learning combines independent basic classifiers to create a powerful classifier for classification. We first determine the sentiment score for the tweet (SS). The pseudocode for this is represented using Algorithm 1 in Table 5.

Using Algorithm 1, a tweet’s sentiment score is computed. A collection of tweets known as the Test tweets were used to test the algorithm. Each of the basic classifiers in the ensemble classifier determines if a particular tweet in the Test tweet has a positive or negative sentiment (Positive/Negative). The classification report for each base classifier was also built using the testing data. Finding out whether a tweet is more likely to be favourable or negative is the next step. After distributing this probability, we use the ensemble technique to give each classifier a weight based on its accuracy. The system then determines the predictions of each classifier to determine the score for the tweet, both positive and negative.

Table 5. Algorithm 1 for calculation of sentiment score of a tweet

```
Input: "Testtweet (TT)
Output: Sentiscore (SS)
foreach $T_i$ in TT do                               // $T_i$ is $i^{th}$ tweet
  $PC_i = 0$                                         // $i^{th}$ positive count
  $NC_i = 0$                                         //$i^{th}$ negative count
  foreach classifier $C_i$ in ensemble do
     if $C_i$ predicts positive then
        $PC_i = +1$;
     End
     Else
        $NC_i = +1$;
     End
  Else
        $Prob(Positive_i) = \frac{PC_i}{PC_i + NC_i}$
        $Prob(Negative_i) = \frac{NC_i}{PC_i + NC_i}$
End
foreach classifier $C_i$ in ensemble do

$Weight_{C_i} = \frac{acc_{C_i}}{\sum_{j=1}^{n} acc_{C_j}}$          //$acc_{C_i}$ is accuracy of $i^{th}$ classifier; j denoted no. of learning classifiers in the
                                                     ensemble and $acc_{C_j}$ is accuracy of $j^{th}$ learning classifier
End
foreach $T_i$ in TT do
        $PS_i = 0$                                   // $i^{th}$ positive score (PS)
        $NS_i = 0$                                   // $i^{th}$ negative score (NS)
  foreach classifier $C_i$ in ensemble do
        if $C_i$ predicts positive then
$PS_i = Weight_{C_i} + Prob(Positive_i)$;
        End
        Else
$NS_i = Weight_{C_i} + Prob(Negative_i)$;
        End
    End
    return $PS_i$, $NS_i$"
End
```

The sentiment of the tweet is predicted using algorithm 2 in Table 6. The tweet's positive and negative score is one of the inputs used by this algorithm. When a tweet has more positive than negative feedback, it is said to have a positive emotion. The emotion of a tweet is called unfavourable if its negative score is higher than its positive score.

4.2.10. Distance calculation

'Cosine similarity' calculates how similar two tweets are to one another. Using the formula, cosine similarity can be calculated in (16):

$$Cos(T_1, T_2) = \frac{T_1 * T_2}{||T_1|| * ||T_2||} \tag{16}$$

where $T_1 and\ T_2$ represent vectors and output value 1 represents high similarity.

In this study, two distinct ensemble strategies were used: Majority voting ensemble (MVE) and Best trained ensemble (BTE).

4.3. Study 2: ML model comparison using D2 (#newsfeed) dataset on CPU-only

Here, the dataset used for evaluation is the twitter (#newsfeed) dataset (D2) to evaluate the classifier performance. The process for downloading data and data processing is explained below. This is applicable to both study 2 and 3.

4.3.1. Download related data from Twitter using Twitter API

With the support of the Twitter API, users can collect tweets from Twitter. Twitter offers REST API and Streaming API, two different types of APIs. For our analysis, we make use of Streaming API. We require a longer connection and an uncapped data rate for collecting a big number of tweets.

We need to have a twitter account before using the Twitter API. Following these, the user is given a username and password that are used to log in. We must log onto the dev.twitter.com website for this purpose using our Twitter credentials. By supplying the essential information on this page, we first construct an application that will be used for streaming tweets When a user wishes to access Twitter data, keys are used to verify their identity.

Considering that the goal of this study is to examine the sentiment of tweets posed on generic, political and media feed from new channel topics, we use *#generic*, *#politics* and *#newsfeed* and collect tweets related to these topics only [31]. As a result, we develop a Python script for this purpose that will be used to retrieve tweets from Twitter. Installing the tweepy Python open-source library is the initial step in writing this script. Tweepy makes it possible for Python to connect to Twitter and use its API to gather data for the study. All of the keys and secrets that we receive from the API are used in this script. To load the data from Twitter, we first develop a listener class.

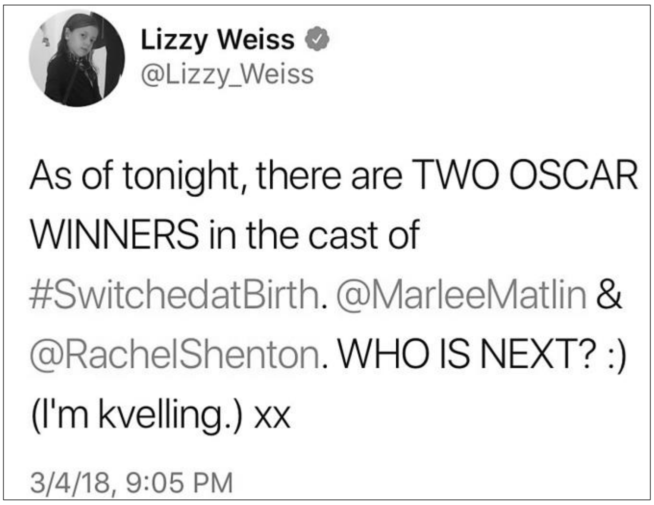
Table 6. Algorithm 2 for predicting sentiment score of a tweet in ensemble model

```
Input: “$T_i$ , $PS_i$, $NS_i$
Output: Sentiment ($S$)
if $PS_i > NS_i$ then
     $S$= ‘positive’;
  Else
    if $NS_i > PS_i$  then
    $S$= ‘negative’;
Else
Calculate cosine similarity of $T_i$ with all other tweets in test_data using the distance calculation
Find the most similar tweet of $T_i$ say $T_j$
      Calculate $PS_j$ and $NS_j$ of $T_j$ using algorithm 1
Use maximum voting
         if $PS_j > NS_j$ then
           $S$= ‘positive’;”
Else
      End
   End
End
```

'*OAuth*' protocol was first built up in order to collect data. *OAuth* is a widely used protocol for authorization. *OAuth* offers user authorization and security. Tweets from Twitter are imported after this script is executed, and we may use them for our study.

4.3.2. Data Pre-processing

Reducing the size of the feature set is made possible by the data preprocessing stage. This is necessary because, as illustrated in Figure 4, a tweet may contain multiple undesirable features.



Lizzy Weiss
@Lizzy_Weiss

As of tonight, there are TWO OSCAR WINNERS in the cast of #SwitchedatBirth. @MarleeMatlin & @RachelShenton. WHO IS NEXT? :) (I'm kvelling.) xx

3/4/18, 9:05 PM

Figure 4. Various features in a sample tweet

For each dataset preparation, to pre-process the raw data, we use following steps: feature extraction, formatting, tokenization, normalization, stemming, stop word removal, lemmatization and word-embedding. In brief, the data preprocessing involves:

- Retweets that begin with the letter *'RT'* are not considered
- External links and usernames preceded by *'@j'* are removed.
- Hashtag *#j* is deleted from the tweet (it's used to identify issues and phrases that are currently trending).
- Emoticons are replaced with their comparable meanings because they can be beneficial in detecting moods.
- The process of getting to a word's root is called stemming..
- Slangs are replaced with words that have the same meaning.
- Stop-words and unnecessary words are eliminated from tweets.

4.3.3. Feature extraction/representation

We extract features from preprocessed tweets at this stage. To convert training tweets into a vector or numeric representation, we used the BoW technique. From all the tweets, BoW learns a lexicon of recognized words. It describes the presence of recognized words within a tweet after learning vocabulary [32].

ML algorithms used in this benchmark study are LOGR, SVM, DT, RF, DT-GB and NB. All the algorithms are explained in the previous section, except DT-GB and Naïve Bayes.

4.3.4. Decision Tree- Gradient Boosting (DT-GB)

DT-GB is a machine learning technique for improving a model's prediction value through subsequent learning steps. Boosting is a strategy for expediting the improvement in projected accuracy to a sufficient ideal value. Gradient refers to the incremental modifications made at each stage of the procedure.

4.3.5. Naive Bayes (NB)

Because it is straightforward, simple to compute, faster for a large quantity of training data, and less sensitive to missing data, the NB Classifier is chosen. The algorithm is frequently employed to classify texts. Based on a conditional probability model, it assigns probabilitiesas per (17), $p(C_k\ j\ x_1,......, x_n)$ for each of K possible outcomes.

$$p(Ck\ j\ x)\ = \frac{(p(Ck)\ p(jx\ Ck)}{p(x)} \tag{17}$$

A scaling factor that is simply dependent on a constant (18).

$$P(E1/E2)\ = \frac{P(E1)\cdot P\left(\frac{E2}{E1}\right)}{P(E2)} \tag{18}$$

Here, *P(E1)* = the Probability of occurrence of event *E1*; *P(E2)* = the Probability of occurrence of event *E2* ; *P(E1/E2)* =The Probability of occurrence of event *E1* given event *E2* ; and *P(E2/E1)* =the probability of occurrence of event *E2* given event *E1*
Accuracy is checked for 60K and 0.16M tweets on CPU-only to gauge scaling performance.

4.4. Study 3: ML model comparison using D3 (#newsfeed) dataset on CPU and GPU

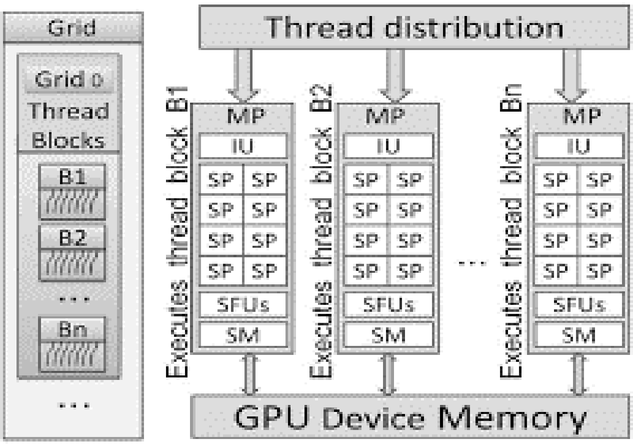


Figure 5. The Nvidia GPU architecture

The Dataset used for evaluation is the twitter *(#politics)* dataset (D3) downloaded as explained in the above section and pre-processed. The algorithms used for benchmarking are DT, SVM, KNN-1,3,5 and CNN which have all been explained in previous sections. The study is further divided into 2 parts:

- First, we study the effect of increase in the number of tweets on the algorithms to evaluate their scaling performance on CPU-only.
- We next study the performance of all classifiers on a Nvidia K40, 12GB GPU.

Many tasks in sentiment analysis which primarily uses NLP can exploit the massive parallelism offered by GPUs. GPU were initially used only for graphics/visualization tasks. Due to the massive number of lightweight cores as shown in Figure 5, researchers have leveraged it for parallelizing codes. Especially for NLP, once the text is hashed, GPUs can offer accelerated results as compared to only a CPU. Figure 6 shows the difference between a CPU and a GPU, with CPUs having very less cores which explains this huge difference.

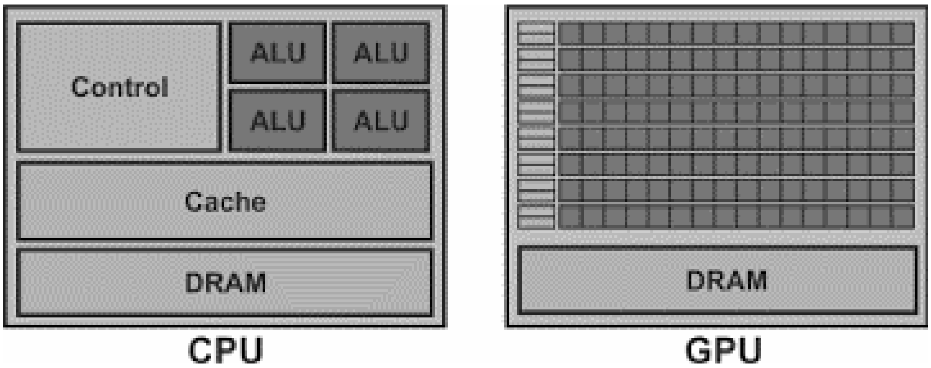


Figure 6. Difference between a CPU and GPU [33]

The goal of using a GPU is to evaluate the speed-up and computational power it can provide for processing tweets in a real-time scenario.

4.5. Evaluation Metrics

In an intra-model comparison on the D1.1, D2, and D3 datasets, the relative performance metrics listed in Tables 7 are evaluated for each base classifier and ensemble method to evaluate the outcomes of the three studies.

On datasets D1.1 and D1.2, we also calculate the average F1-score over positive and negative class as $F1^{PN}$, eliminating neutral class, for inter-model comparison with contemporary models (19).

Table 7. Evaluation metrics

Parameter	Mathematical Expression	Definition
Accuracy	$\frac{(TN\ +TP)}{(TN+FN+FP+TP)}$	Out of all modules, the percentage of projected job categories that are inspected is indicated.
Precision	$\frac{TP}{(TP+FP)}$	It refers to the consistency with which repeated measurements under the same conditions results in same findings.
Recall	$\frac{TP}{(\mathrm{TP+FN})}$	It indicates the number of relevant objects that must be identified.
F-measure	$2x\frac{Recall.Precision}{(\ Recall+Precision)}$	It's also known as the harmonic mean since it is a combination of precision and recall values into a single score.

$$\text{Average F1 } (F1^{PN})=\frac{1}{2}\ (F1^{Positive}+F1^{Negative})x\ 100 \tag{19}$$

Additionally, we construct average recall (AveRec) for dataset D1.2, which is determined by averaging the positive, negative, and neutral recall values in accordance with (20).

The Macro average mean absolute error (MAE^{M}) is calculated for dataset D1.4 as the classification metric.

$$(\mathrm{MAE^{M}})\ (h,T_e)=\frac{1}{|C|}\sum_{j=1}^{|C|}\frac{1}{\left|T_{e_j}\right|}\sum_{x_i\in T_{e_j}}|h(x_i)-y_i| \tag{20}$$

where y_i denotes the true label of x_i, and $h(x_i)$ is its predicted label, T_{e_j}represents the set of test documents whose true class is c_j. $|h(X_i) - y_i|$ represents the distance between classes $h(x_i)$ and y_i. We take the gap between strongly positive and negative , for instance, to be 3.

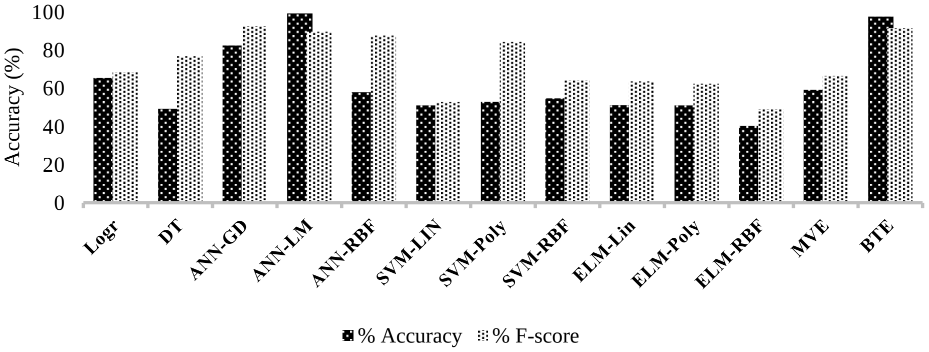


Figure 7a. Performance of various base classifiers in standalone and ensemble model using D1.1 dataset: Accuracy and F1-score for positive class

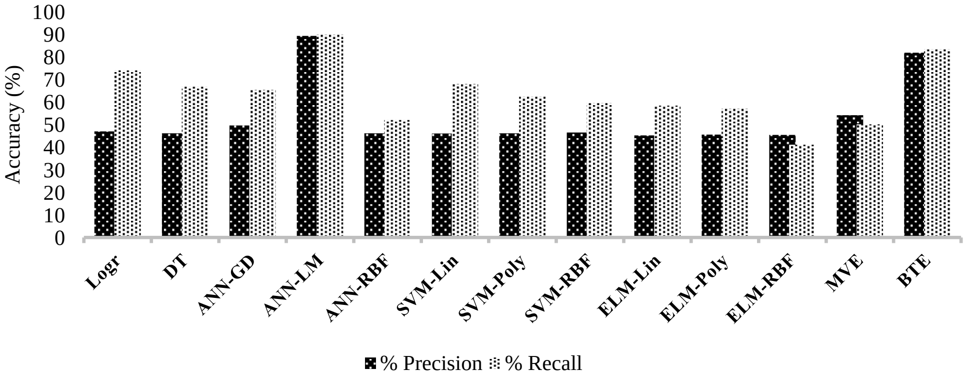


Figure 7b. Performance of various base classifiers in standalone and ensemble model using D1.1 dataset: Precision and Recall values for positive class

5. RESULTS AND DISCUSSION

The intra-model performance results of benchmark study-1 using SRML model on dataset D1.1 for positive class are shown in Figure 7a and 7b. Among all base classifiers in standalone mode, ANN-LM algorithm performed the best and ELM-RBF returned poor results. This is possibly due to refined, more efficient learning and weight estimation in ANN-LM and ANN-GD alogorithms. Logistic regression (LOGR) performed better than DT, ELM and SVM classifiers. Also, it is found that amongst the two ensemble strategies, BTE performed better than MVE. BTE ensemble exhibits an accuracy of 97.40%, precision 81.83%, recall 83.3% and F-Measure of 91.3%, higher than the MVE. Comparing ANN and Ensemble models, though ANN-LM has emerged as the single highest contributing model, the balanced results by combining predictions from more models, handling linear and non-linear type of data in the dataset using the CFS augmented BTE ensemble strategy is preferred than any single contributing model and is robust since it reduces the spread or dispersion of the predictions and model performance.

The results of benchmark study 2 on dataset D2 for 60K and 0.16M datapoints is shown in Figure 8. In this study, we did not consider any neural network based model in the benchmark comparison. All base classifiers performed poorly in terms of accuracy, with SVM outperforming LOGR and other algorithms. In scaling characteristics when increasing the tweets from 60K to 0.16M, datapoints, very little performance increase is observed as seen from the results. Though the plotted confusion matrix in Figure 9 for the best performing SVM is a good indication for classifying tweets into positive, negative and neutral class, but in terms of accuracy and overall performance, standalone ML classifiers are not suitable for design of a real-time, robust expert system for sentiment analysis.

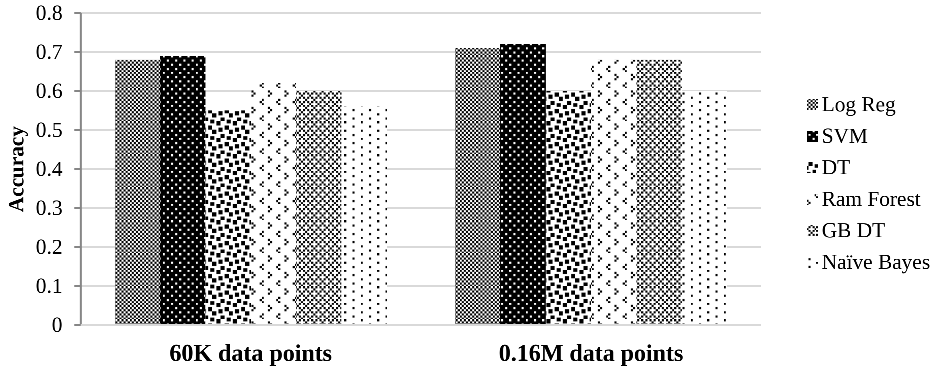


Figure 8. Benchmark evaluation of different classifiers using D2 dataset and effect of scaling no. of tweets

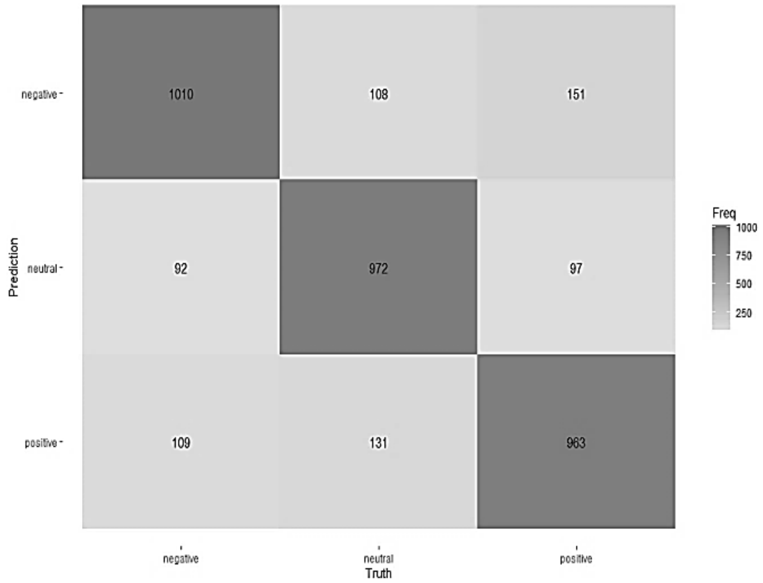


Figure 9. Confusion Matrix for the SVM model built for 10K datapoints

Table 8. Results from classifiers on CPU for twitter (#politics) dataset (D3) as part of study 3

Techniques	CPU Processor	memory/Disk	Epochs	CPU Time/Sec	Tweet Count	Accuracy (%)
Decision tree	Intel® Xeon® CPU @ 2.00GHz	13G, 30G	3	2031	100000	69.8
				3229	200000	70.21
				3943	300000	70.33
KNN-1			10	12	100000	69.2
				16	200000	69.81
				25	300000	70.021
KNN-5			10	13.33	100000	69.5
				17	200000	69.7
				28	300000	70
KNN-3			10	13.29	100000	70
				16.3	200000	69.9
				25	300000	70.1
Support vector machine			10	6	100000	82.1
				11	200000	82.26
				18	300000	82.23
Convolution neural network			10	8	100000	83.22
				10	200000	82.87
				15	300000	82.9

The results of study 3 on CPU alone using dataset D3 are shown in Table 8. The tweets are increased from 100K to 300K for all the models to study their scaling characteristics. It is observed that CNN returns the highest accuracy even as the number of tweets increase from 100K to 300K. The performance is sustained which is a very critical indicator in the design of a real-time expert system for sentiment analysis. Also, as second part of the study Table 9 shows the results of all techniques using the CPU+GPU. Though the GPUs are not able to influence the accuracy of the results, it is observed that GPUs aid in accelerating sentiment resolution much faster. We observe a minimum 10x speed-up in processing of tweets across all techniques. The comparative results of the CPU and CPU+GPU runs to study the time taken is shown in Figure 10. This is also a critical indicator for consideration in the design of a real-time robust expert system.

Table 9. Results from classifiers on CPU+GPU for twitter (#politics) dataset (D3) as part of study 3

Techniques	CPU Processor	Memory/ Disk	Epoch	CPU Time/ Sec	GPU Time/ Sec	Tweet Count	Accuracy (%)
Decision tree	Intel® Xeon® CPU @ 2.00GHz	13G, 30G	3	2031	265.814	100000	69.8
				3229	506.931	200000	70.21
				3943	626.531	300000	70.33
KNN-1			10	12	9	100000	69.2
				16	11	200000	69.81
				25	17.2	300000	70.021
KNN-5			10	13.33	10	100000	69.5
				17	10	200000	69.7
				28	16	300000	70
KNN-3			10	13.29	10	100000	70
				16.3	9.3	200000	69.9
				25	19	300000	70.1
Support vector machine			10	6	5.4	100000	82.1
				11	10	200000	82.26
				18	16	300000	82.23
Convolution neural network			10	8	6.8	100000	83.22
				10	8.8	200000	82.87
				15	12.9	300000	82.9

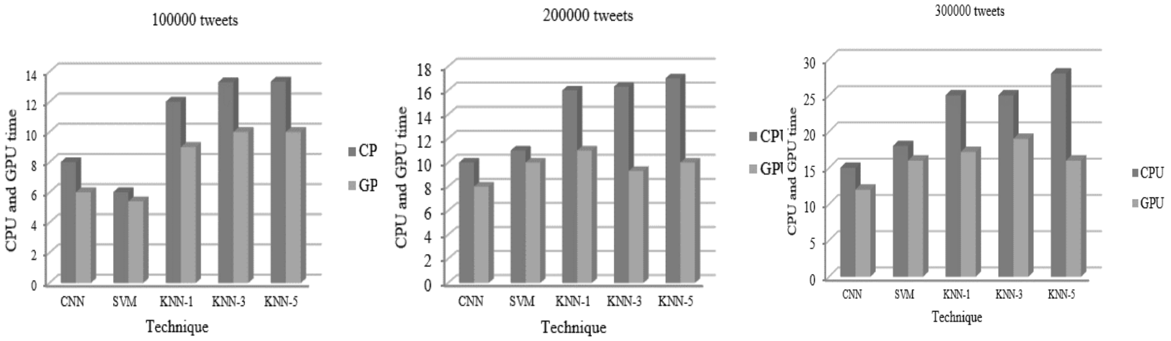


Figure 10. Performance on CPU vs. GPU on D3 dataset

Tables 10,11,12 and 13 present the inter-model comparison with current state-of-the-art models utilizing datasets D1.1, D1.2, D1.3, and D1.4. Previous studies have classified sentiment using a variety of complex ML and transformer models on dataset D1.1.

Table 10. Performance evaluation of SRML model with cutting-edge models on D1.1 dataset

			Positive class (%)			Negative class (%)			Ave (%)
Author(/s). year of publication	System	Acc	Pre	Rec	F1	Pre	Rec	F1	F1PN
Plaza-Del-Arco F, et al. 2022 [34]	Multi-task learning for HOF detection using BERT	Macro Avg Pre: 79.81 Macro Avg. Rec: 77.78							78.70 (Macro Avg)
Bostan L-A-M and Klinger R. 2018 [35]	Max Ent classifier with BOW as features via 10-fold cross validation in precision, recall, and F1	Pre, Rec and F1 values for various text-emotions using CrowdFlower dataset Joy- 42,35,38 Sadness-26,28,27 Fear-32,30,31 Anger-21,29,24 Disgust-6,23,9 Surprise-9,9,9							25 (Micro Avg)
Chandrasekaran G, et al. 2022 [36]	VGG-19	73	66	75	70	81	72	76	
	DenseNet121	89	86	88	87	92	90	91	
	Resnet50V2	75	74	62	68	76	84	80	
Proposed model	CFS augmented BTE	97.4	81.83	83.33	82.57	86.49	85.39	85.93	84.25

Table 11. Performance evaluation of SRML model with cutting-edge models on D1.2 dataset

Author(/s). year of publication	System	Acc	AveRec	F1PN
Cliche M. 2017 [37]	BB_twtr	68.1	68.5	65.8
Baziotis C, et al. 2017 [38]	DataStories	68.1	67.7	65.1
Rouvier M. 2017 [39]	LIA	67.6	67.4	66.1
Aziz RHH, et al. 2020 [40]	Classifier Ensemble	72.95	68.9	69.1
Proposed model	CFS augmented SRML BTE	74.87	71.28	72.71

Table 12. Performance evaluation of SRML model with cutting-edge models on D1.3 dataset

Author(/s). year of publication	System	Acc	F1-score
Cliche M. 2017 [37]	BB_twtr	89.7	89
Baziotis C, et al. 2017 [38]	DataStories	86.9	86.1
Rouvier M. 2017 [39]	Tweester	86.3	85.6
Aziz RHH, et al. 2020 [40]	Classifier Ensemble	90.8	94.4
Proposed model	CFS augmented SRML BTE	92.83	94.74

Table 13. Performance evaluation of SRML model with cutting-edge models on D1.4 dataset

Author(/s). year of publication	System	MAEM
Rozental A and Fleischer D. 2017 [41]	Amobee-C-137	0.599
Rouvier M. 2017 [39]	Tweester	0.623
Balikas G. 2017 [42]	TwiSe	0.640
Aziz RHH, et al. 2020 [40]	Classifier Ensemble	0.589
Proposed model	CFS augmented SRML BTE	0.521

Table 10 compares various existing studies with our strategy for sentiment analysis using D1.1 dataset. The authors in [35] though have used a slightly different datasets (text-based emotion detection and images) but from the same"CrowdFlower dataset. It is compared here as it provides insight into the classification approach used and their outcome. On every metric, it can be observed that the suggested CFS augmented SRML BTE model outperforms existing systems that have employed the latest BERT model, the Maximum Entropy classifier, and Deep Neural Networks like VGG-19, DenseNet121, and Resnet50V2.

On dataset D1.2 i.e., the SemEval-2017 Task 4A, Table 11 compares the proposed model to the existing studies. The three classes in this dataset—positive, negative, and neutral—are intended to help with the process of classifying messages based on their polarity. CNN, LSTM, neural networks, and weighted majority vote ensemble classifier were used by the top 4 systems for this task. The suggested CFS augmented SRML BTE system performs 3.6% better in terms of $F1^{PN}$ than the top-performing system [41].

On dataset D1.3 i.e., the SemEval-2017 Task 4B, Table 12 compares the proposed model to the existing studies. Two-point classification of the communication with specified themes is the intended use of this dataset. The Acc and F1-score of the top system [39] are improved by 2% and ~0.5%, respectively, by the suggested CFS augmented SRML BTE model. CNNs and LSTMs with an Attention model were utilized by the other top 3 systems for this task.

The proposed model is compared to the existing studies on dataset D1.4 i.e., the SemEval-2017 Task 4C dataset, in Table 13. This dataset was created for a topic-based, five-point classification job". In comparison to all other models, the suggested CFS augmented BTE model yields the lowest MAE^{M} (0.521), where lower is better. TwiSe system used LR classifier and the ensemble weighted majority vote classifier are the other two systems that used DL techniques to accomplish this task.

Therefore, from intra-model study 1 and inter-model comparisons, the CFS augmented Best trained Ensemble (BTE) emerges as the model of choice. Standalone/base ML clssifiers are not suitable and fall short in performance as seem from study 1, 2 and 3, except those from a neural network lineage. The ANN and CNN models emerged as the single highest contributing models from study 1 and 3. GPUs help in accelerating sentiment classification and have emerged as an alternative architecture of choice to CPU. Overall, SentiMLBench provides researchers with insights into the algorithmic performance and architecture choices most suitable for large and complex twitter sentiment analysis tasks.

6. CONCLUSION

One of the key approaches for twitter sentiment analysis has been to employ various ML-based classifiers and make do with the findings. There are no consolidated comparative performance benchmarks available for various classifiers from ML, DL and latest transformer-based approaches using datasets from various domains. Also, alternative hardware architectures need to be explored towards development of a robust real-time expert system for sentiment analysis. In this paper, we propose SentiMLBench which is a critical benchmark evaluation of various machine learning techniques compared in three intra-model studies using different state-of-the-art datasets and also compared with existing models in inter-model study. ML classifiers in standalone mode, in two ensemble techniques, scaling of tweets to measure algorithmic scaling characteristics and the impact of GPU are some of the areas explored. From results, the ANN and CNN models emerge as the single highest contributing models from study 1 and 3. Though the ANN-LM model outperformed all models including the ensemble technique in study 1, the CFS augmented BTE model is preferred owing to the linear and non-linear type of data in the dataset and its robustness since it reduces the spread or dispersion of the predictions and model performance. This model also outperforms some of the latest cutting-edge models from the inter-model comparison study. GPU emerged as an alternative hardware architecture of choice as it provides a minimum speed-up of 10x in processing of tweets. The recommended techniques can be leveraged by organizations that want to build a real-time consumer opinion monitoring system or by customers that want to choose the best product based on public opinion on the fly. SentiMLBench will provide researchers with insights to choose the optimal algorithm(s) and hardware architecture as a result of these findings. We plan to extend the outcomes of this study to develop a robust expert system for real-time sentiment analysis as part of future research. Although the proposed study focuses on data from Twitter, it can also be used in the analysis of data from other social media platforms.

ACKNOWLEDGMENTS

The researcher expresses profound gratitude and appreciation to Savitribai Phule Pune University, Internal Quality Assurance Cell (IQAC) for providing research funding under 'ASPIRE Research Mentorship Grant' funding scheme for this research project (Grant Number - 18TEC001501). We would like also to thank Pune Institute of Computer Technology, Pune for providing their research facilities.

REFERENCES

[1] J. Singh, G. Singh, and R. Singh, "Optimization of sentiment analysis using machine learning classifiers," Human-centric Computing and Information Sciences, vol. 7, no. 1, Dec. 2017, doi: 10.1186/s13673-017-0116-3.

[2] M. Haenlein and A. M. Kaplan, "An Empirical Analysis of Attitudinal and Behavioral Reactions Toward the Abandonment of Unprofitable Customer Relationships," *Journal of Relationship Marketing*, vol. 9, no. 4, pp. 200–228, Nov. 2010, doi: 10.1080/15332667.2010.522474.

[3] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-Based Methods for Sentiment Analysis," *Computational Linguistics*, vol. 37, no. 2, pp. 267–307, Jun. 2011, doi: 10.1162/coli_a_00049.

[4] E. Aydogan and M. A. Akcayol, "A comprehensive survey for sentiment analysis tasks using machine learning techniques," *2016 International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*, Aug. 2016, doi: 10.1109/inista.2016.7571856.

[5] A. Chinnalagu and A. K. Durairaj, "Context-based sentiment analysis on customer reviews using machine learning linear models," *PeerJ Computer Science*, vol. 7, p. e813, Dec. 2021, doi: 10.7717/peerj-cs.813.

[6] N. F. F. da Silva, E. R. Hruschka, and E. R. Hruschka, "Tweet sentiment analysis with classifier ensembles," *Decision Support Systems*, vol. 66, pp. 170–179, Oct. 2014, doi: 10.1016/j.dss.2014.07.003.

[7] J. V. Lochter, R. F. Zanetti, D. Reller, and T. A. Almeida, "Short text opinion detection using ensemble of classifiers and semantic indexing," *Expert Systems with Applications*, vol. 62, pp. 243–249, Nov. 2016, doi: 10.1016/j.eswa.2016.06.025.

[8] A. Yenkikar, M. Bali, and C. N. Babu, "EMP-SA: Ensemble Model based Market Prediction using Sentiment Analysis," *International Journal of Recent Technology and Engineering*, vol. 8, no. 2, pp. 6445–6452, Jul. 2019, doi: 10.35940/ijrte.b2197.078219.

[9] S. Poria, E. Cambria, A. Gelbukh, F. Bisio, and A. Hussain, "Sentiment Data Flow Analysis by Means of Dynamic Linguistic Patterns," *IEEE Computational Intelligence Magazine*, vol. 10, no. 4, pp. 26–36, Nov. 2015, doi: 10.1109/mci.2015.2471215.

[10] S. Kiritchenko and S. M. Mohammad, "Capturing Reliable Fine-Grained Sentiment Associations by Crowdsourcing and Best–Worst Scaling," *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, doi: 10.18653/v1/n16-1095.

[11] R. Socher, "Deep Learning for Sentiment Analysis - Invited Talk," *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 2016, doi: 10.18653/v1/w16-0408.

[12] S. TAN and J. ZHANG, "An empirical study of sentiment analysis for chinese documents," *Expert Systems with Applications*, vol. 34, no. 4, pp. 2622–2629, May 2008, doi: 10.1016/j.eswa.2007.05.028.

[13] K. Dashtipour, M. Gogate, A. Adeel, A. Hussain, A. Alqarafi, and T. Durrani, "A Comparative Study of Persian Sentiment Analysis Based on Different Feature Combinations," *Lecture Notes in Electrical Engineering*, pp. 2288–2294, Jun. 2018, doi: 10.1007/978-981-10-6571-2_279.

[14] J. Islam and Y. Zhang, "Visual Sentiment Analysis for Social Images Using Transfer Learning Approach," *IEEE Xplore*, Oct. 01, 2016. https://ieeexplore.ieee.org/document/7723683/

[15] A. Severyn and A. Moschitti, "Twitter Sentiment Analysis with Deep Convolutional Neural Networks," *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Aug. 2015, doi: 10.1145/2766462.2767830.

[16] L. Yanmei and C. Yuda, "Research on Chinese Micro-Blog Sentiment Analysis Based on Deep Learning," *2015 8th International Symposium on Computational Intelligence and Design (ISCID)*, Dec. 2015, doi: 10.1109/iscid.2015.217.

[17] Q. You, J. Luo, H. Jin, and J. Yang, "Joint Visual-Textual Sentiment Analysis with Deep Neural Networks," *Proceedings of the 23rd ACM international conference on Multimedia*, Oct. 2015, doi: 10.1145/2733373.2806284.

[18] X. Ouyang, P. Zhou, C. H. Li, and L. Liu, "Sentiment Analysis Using Convolutional Neural Network," *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, Oct. 2015, doi: 10.1109/cit/iucc/dasc/picom.2015.349.

[19] S. Poria, I. Chaturvedi, E. Cambria, and A. Hussain, "Convolutional MKL Based Multimodal Emotion Recognition and Sentiment Analysis," *2016 IEEE 16th International Conference on Data Mining (ICDM)*, Dec. 2016, doi: 10.1109/icdm.2016.0055.

[20] S. Barreto, R. Moura, J. Carvalho, A. Paes, and A. Plastino, "Sentiment analysis in tweets: an assessment study from classical to modern text representation models," *arXiv:2105.14373 [cs]*, May 2021, [Online]. Available: https://arxiv.org/abs/2105.14373

[21] N. Chintalapudi, G. Battineni, and F. Amenta, "Sentimental Analysis of COVID-19 Tweets Using Deep Learning Models," *Infectious Disease Reports*, vol. 13, no. 2, pp. 329–339, Apr. 2021, doi: 10.3390/idr13020032.

[22] G. Chandrasekaran and J. Hemanth, "Deep Learning and TextBlob Based Sentiment Analysis for Coronavirus (COVID-19) Using Twitter Data," *International Journal on Artificial Intelligence Tools*, vol. 31, no. 01, Feb. 2022, doi: 10.1142/s0218213022500117.

[23] A. Yenkikar and C. N. Babu, "AirBERT: A fine-tuned language representation model for airlines tweet sentiment analysis," *Intelligent Decision Technologies*, vol. Preprint, no. Preprint, pp. 1–17, Jan. 2022, doi: 10.3233/IDT-220173.

[24] X. Jiang, C. Song, Y. Xu, Y. Li, and Y. Peng, "Research on sentiment classification for netizens based on the BERT-BiLSTM-TextCNN model," *PeerJ Computer Science*, vol. 8, p. e1005, Jun. 2022, doi: 10.7717/peerj-cs.1005.

[25] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya, "ABCDM: An Attention-based Bidirectional CNN-RNN Deep Model for sentiment analysis," *Future Generation Computer Systems*, vol. 115, pp. 279–294, Feb. 2021, doi: 10.1016/j.future.2020.08.005.

[26] C.-R. Ko and H.-T. Chang, "LSTM-based sentiment analysis for stock price forecast," *PeerJ Computer Science*, vol. 7, p. e408, Mar. 2021, doi: 10.7717/peerj-cs.408.

[27] A. Yenkikar, C. N. Babu, and D. J. Hemanth, "Semantic relational machine learning model for sentiment analysis using cascade feature selection and heterogeneous classifier ensemble," *PeerJ Computer Science*, vol. 8, p. e1100, Sep. 2022, doi: 10.7717/peerj-cs.1100.

[28] "Sentiment Analysis in Text - dataset by crowdflower," *data.world*. https://data.world/crowdflower/sentiment-analysis-in-text

[29] S. Rosenthal, N. Farra, and P. Nakov, "SemEval-2017 Task 4: Sentiment Analysis in Twitter," *arXiv:1912.00741 [cs]*, Dec. 2019, [Online]. Available: https://arxiv.org/abs/1912.00741

[30] S. Baccianella, A. Esuli, and F. Sebastiani, "SENTIWORDNET 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining," 2010. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2010/pdf/769_Paper.pdf

[31] A. Yenkikar, C. N. Babu, and D. J. Hemanth, "SENTINET: A Deep Sentiment Analysis Network for Political Media Bias Detection," *DYNA*, vol. 97, no. 6, pp. 645–651, Nov. 2022, doi: 10.6036/10593.

[32] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv.org*, Sep. 07, 2013. https://arxiv.org/abs/1301.3781

[33] V. Thambawita, R. Ragel, and D. Elkaduwe, "To Use or Not to Use: Graphics Processing Units for Pattern Matching Algorithms," arXiv:1412.7789 [cs], Dec. 2014, Accessed: Jun. 01, 2021. [Online]. Available: https://arxiv.org/abs/1412.7789

[34] F. Plaza-Del-Arco, S. Halat, S. Padó, and R. Klinger, "Multi-Task Learning with Sentiment, Emotion, and Target Detection to Recognize Hate Speech and Offensive Language 4.0 International (CC BY 4.0)," 2022. [Online]. Available: https://arxiv.org/pdf/2109.10255.pdf

[35] L.-A.-M. Bostan and R. Klinger, "An Analysis of Annotated Corpora for Emotion Classification in Text Title and Abstract in German," 2018. [Online]. Available: https://aclanthology.org/C18-1179.pdf

[36] G. Chandrasekaran, N. Antoanela, G. Andrei, C. Monica, and J. Hemanth, "Visual Sentiment Analysis Using Deep Learning Models with Social Media Data," *Applied Sciences*, vol. 12, no. 3, p. 1030, Jan. 2022, doi: 10.3390/app12031030.

[37] M. Cliche, "BB_twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs," *arXiv.org*, 2017. https://arxiv.org/abs/1704.06125

[38] C. Baziotis, N. Pelekis, and C. Doulkeridis, "DataStories at SemEval-2017 Task 4: Deep LSTM with Attention for Message-level and Topic-based Sentiment Analysis," *ACLWeb*, Aug. 01, 2017. https://aclanthology.org/S17-2126/

[39] M. Rouvier, "LIA at SemEval-2017 Task 4: An Ensemble of Neural Networks for Sentiment Classification," *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 2017, doi: 10.18653/v1/s17-2128.

[40] R. H. H. Aziz and N. Dimililer, "Twitter Sentiment Analysis using an Ensemble Weighted Majority Vote Classifier," *2020 International Conference on Advanced Science and Engineering (ICOASE)*, Dec. 2020, doi: 10.1109/icoase51841.2020.9436590.

[41] A. Rozental and D. Fleischer, "Amobee at SemEval-2017 Task 4: Deep Learning System for Sentiment Detection on Twitter," *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 653–658, 2017, doi: 10.18653/v1/S17-2108.

[42] G. Balikas, "TwiSe at SemEval-2017 Task 4: Five-point Twitter Sentiment Classification and Quantification," *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 2017, doi: 10.18653/v1/s17-2127.

BIOGRAPHY OF AUTHORS

Anuradha Yenkikar received the B. E. degree in CSE from Dr. Babasaheb Ambedkar Marathwada University, in 2009, and Master's degree in IT from Savitribai Phule Pune University, Pune and pursuing her PhD from Ramaiah University of Applied Sciences, Bangalore. She has 10 years of teaching experience. She has published 8 papers in various International Journals and conferences. She has a filed a patent to her name Her current research interests include Machine Learning, Deep Learning, and Parallel Computing. She has completed a research project funded by Savitribai Phule Pune University, Internal Quality Assurance Cell (IQAC) under 'ASPIRE Research Mentorship Grant' funding scheme. She can be contacted at email: anu.jamkhande@gmail.com

C. Narendra Babu received the received B. Tech (CSE) from Kuvempu University in the year 2000, M. Tech (CSE) from M.S Ramaiah Institute of technology in the year 2004 and PhD from JNT University Anantapur in the year 2015. August 2014. He is presently working as Associate Professor under CSE in M.S. Ramaiah University of Applied Sciences, Bengaluru. He has 20 Years of Teaching Experience. His areas of interest include Data analytics, Data Modelling and analysis, Big Data, Social media analytics, Machine Leaning, Artificial Intelligence, Statistical Approaches, Innovations in education. He has filed 3 patents and published 22 international peer-reviewed journals and 17 proceedings of conference papers. He has written one book chapter in - Taylor & Francis. He is a member of IEEE Senior Member, IEEE – Education Society member and IAENG life member. He can be contacted at email: narendrababu.c@gmail.com