



Sentiment Classification for Film Reviews by Reducing Additional Introduced Sentiment Bias

Fery Ardiansyah Effendi¹, Yuliant Sibaroni²

^{1,2}School of Computing, Informatics Faculty, Telkom University

¹feryardiansyaheff@student.telkomuniversity.ac.id, ²yuliant@telkomuniversity.ac.id*

Abstract

Film business and its individual reviews cannot be separated and film review sites such as IMDb is a credible source of reviews posted in public forums. With IMDb site reviews being unstructured and bias-heavy, classification methods by reducing additional sentiment bias is needed to create a balanced classification with lower polarity bias. Elimination of additional sentiment bias will improve the model as polarity is defined by non-bias method, resulting in models correctly defined which sequences of words is either positive or negative. This research limits the dataset by 50.000 rows of randomly extracted reviews from the IMDb website using dataset preparation methods such as Preprocessing, POS-Tagging, and Word Embeddings. Then preprocessed data is used in classification methods such as ANN, SWN, and SO-Cal. This paper also used bias processing methods such as Hyperparameter Tuning and BPM, with outputs evaluated using Accuracy and PBR metrics. This research yields 77.39 % for ANN, 66.32% for BPM, 75.6% for SO-Cal, and 76.26% for Hybrid classification. Best PBR resulted in two lexicon-based methods on 0.0009 for BPM, and 0.00006 for SO-Cal. More advanced model configuration in ANN can improve the model, and much complex lexicon models will be a future in the research topic.

Keywords: Sentiment Classification, Machine Learning, ANN, Lexicon-based method, BAT, SO-Cal.

1. Introduction

In film credibility regarding the film industry, film reviews such as IMDb movie review sites cannot be separated because of their high credibility [1]. With the IMDb site frequently becoming a reference to rate films circulating in media with varied genres and more prominent communities to review the said film, IMDb is also becoming one factor in how the film itself has value [1]. IMDb also has a function for finding film references to watch and pulling audiences for recommended films from IMDb communities. From the survey done by [2], in film reviews, 82% adult-Americans are reading film reviews for their movies that are yet to be watched, and other surveys show that 77,5% of people of Chinese nationality read the reviews the first time before deciding what film to enjoy [2]. Therefore, positive reviews for a movie are defining factors to make the film sales increase and otherwise. Research done in [2] shows that added film valency with varied film reviews has more significant sales than little various results.

Research [1] uses IMDb dataset to determine the film's success with basic machine learning models such as

Linear Regression, Logistic Regression, and Support Vector Machine (SVM) with an accuracy of 51% for Linear Regression, 42.2% for Logistic Regression, and 39% for SVM respectively. The problem in research [1] is the lack of depth on methods being restricted to only basic approaches and disadvantages showing low accuracies through sentiment classification. The topic is then further explored in research [3] as Hybrid Feature Extraction (Machine Learning and Lexicon-based methods) is studied, resulting in a decent performance increase from previous research. The showcase of best classifier performances in research [3] is 78.333% and 83.933% using Maximum Entropy classifier with different feature selection methods. Research [3] has a fair system design and used it as a blueprint to model a Hybrid Machine Learning and Lexicon feature extraction. Research [3] explores semantic orientation with Positive and Negative Count (PC/NC), Positive Connotation Count (PCC), and Negative Connotation Count (NCC) as main features extracted from the certain Lexicon. The Lexicon used in [3] is limited to 6800-word vocabulary, including connotation words. Therefore, an opportunity to widen the vocabulary range

is used in this research. Publication [4] introduces additional bias reduction for sentiment analysis, suggesting methods of Bias Aware Thresholding, Sentiment-Orientation Calculator, and Proposed bias-processing strategy. Study [4] results in the best performance in the “Kitchen” dataset with 71.41 % and 0.76% PBR for the Proposed bias-processing method. Said bias-processing strategy is used in this research.

Based on the background, this paper aims to show performance comparisons of thresholds specified in the Bias Processing method in SentiWordNet Scoring calculation for determining the best threshold value in the sentiment classification IMDb reviews. This paper also aims to show individual performance comparisons of thresholds set in the Bias Processing Methods, Artificial Neural Network, and Semantic Orientation Calculator for determining the best threshold point in the sentiment classification of IMDb reviews. And also, this paper's goal aims to show the comparison between the Artificial Neural Network method and the Bias Aware Thresholding method from SentiWordNet, the Semantic Orientation Calculator (SO-Cal) Hybrid Classification of said methods with the best threshold in sentiment classification. For study limitation set for this study is that the dataset used is the “IMDB Dataset of 50K Movie Reviews,” which is already available on Kaggle and has only positive and negative review labels.

In publication [4], additional bias reduction for sentiment analysis with different datasets is conducted, suggesting the methods of Bias Aware Thresholding, Sentiment-Orientation Calculator (SO-Cal), and Proposed bias-processing strategy. Research [4] results in the best performance of the “Kitchen” dataset with the accuracy of 71.41 % and 0.76% PBR for the Proposed bias-processing strategy, which has the best performance compared to the two previous methods. The bias-processing approach proposed in research [4] has decent accuracy for Lexicon-based analysis. It is used in this dataset as one of the methods in Lexicon-based classification with SO-Cal, to balance the PBR. Evaluation metrics used in the study [4] are Accuracy, F-Measure, and Polarity Bias Rate to calculate the bias, also used in this research to measure best thresholds results. In the dataset In study [6] with different datasets, sentiment analysis done using the Lexicon-based method has a pretty good performance with the SVM Classifier with an accuracy of 83.27%. Multilevel Semantic Network was suggested by Research [7] for sentiment analysis method with similar data with a maximum accuracy of 74.2%.

State-of-the-art research in the field is Document Embeddings Neural Network with Cosine Similarity resulted in 97.4% accuracy in a neural network using grid search [23]. Study [23] is the best performance you can get with the IMDb dataset, even though methods used in hyperparameter optimization are lacking. Study

[23] Neural Network structure is the blueprint of this research, as it is similar, with a better lexicon in document embeddings using GloVe. This research also proposed a better version of GloVe, with more vocabulary and dimension. Another state of art research mentioned is XLNet, which is capable of denoising and autoencoder the dataset [24]. Research [24] yields 96.21% accuracy, almost reaching the Document Embeddings framework in different language understanding and modeling contexts. Study [24] proposed sampling of the IMDb dataset in 50.000 records with random sample involved, also used in this research.

Semantic Orientation Calculator was first proposed in a study [10] to classify the polarity of a document. Research [12] suggests a Supervised-learning approach with a Convolutional Neural Network model and Hyperparameter Tuning for Neural Network-based methods with configurable parameters, namely the number of Neurons, Dropout Rate, and Weight Initialization Mode. This became the blueprint of this research's Neural Network, as the Convolutional Neural Network model already works well with sentiment analysis [12]. Hyperparameter tuning is done to avoid bias in the model and personalize the model to the Input dataset [12]. Research [16] describes several Activation Functions that can be used as parameters in Hyperparameter Tuning, namely Sigmoid, Hard Sigmoid, Tanh, Softmax, Softsign, and ReLU functions. Other parameters such as Training Optimizer, Learning Rate, Batch Size, and Epoch were proposed in the study [17] and used to improve the model. Research [5] suggests several preprocessing techniques in sentiment analysis using Artificial Neural Networks such as removing Stopwords, Stemming, deleting Hyperlinks, deleting Hashtags, and deleting unused characters. Research [13] proposes to clean the data using the Preprocessing method with case folding, punctual removal, tokenization, stopword removal, and stemming. Study [13] elaborated further in Preprocessing Stopwords and not used characters such as numbers and special symbols (e.g., #, @, =, ...).

In the publication [4], the suggested Supervised Learning method for the IMDb dataset is Artificial Neural Network, with the proposed bias reduction methods, namely Semantic-Orientation Calculator (SO-Cal) and Bias Aware Thresholding. Bias processing strategy used instead of BAT in a significant decrease of PBR in study [4]. In the study [4], the evaluation analysis suggested was Confusion Matrix with accuracy, precision, recall, and the addition of the polarity bias rate (PBR). The research relationship related to this journal is that Artificial Neural Networks and Lexicon-based are the best methods proposed and used. In research [15], the results obtained in sentiment analysis refer to Hybrid and not to individual approaches from Lexicon-based and Neural Networks. And in a study conducted by [4],

reducing bias in the data classification process indicates an accuracy increase. Therefore, in this journal, individual research by classification method is carried out with bias processing methods to reduce bias.

2. Research Method

In research methods, there is a chart of the stages passed for the sentiment analysis classification. The description of the system design and its description is as follows:

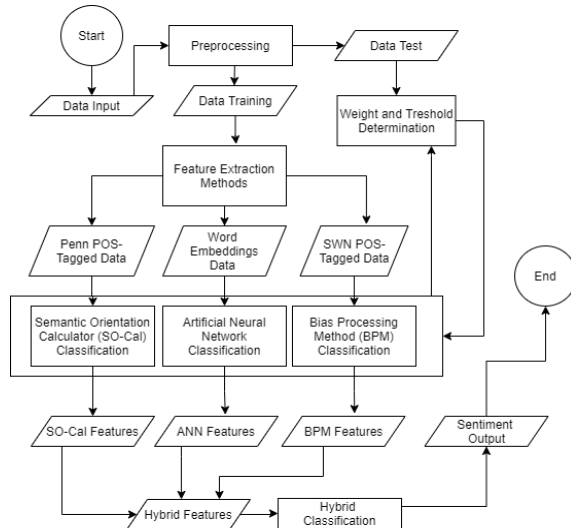


Figure 1. System Design for Sentiment Analysis Classification Process.

2.1. Dataset and Lexicons

The dataset used in this research is "IMDb Dataset of 50K Movie Reviews". This dataset contains film reviews in the IMDb site database, with 50,000 data lines taken as a sample as previous studies use random 50,000 lines in their test and train set combined. Sample used in this research is divided by two dataset such as test and train set with fraction at 50% or 25000 test data and 25000 train data respectively. The site reviews in this dataset are from films that have mixed reviews in polarity and also the polarity of its review labeled 'negative' or 'positive.' Neutral sentiments do not exist in the dataset as their polarity will not matter in sentiment classification, even though Lexicon based methods can yield neutral results.

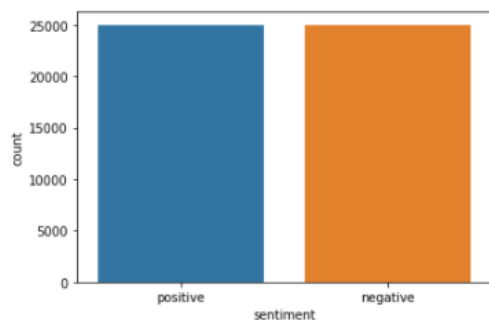


Figure 2. IMDb balanced dataset Distribution

Table 1. "IMDb Dataset of 50K Movie Reviews" Dataset Description

Attributes	Description
Reviews	Contains a sample of 50k reviews from the IMDb film review site with balanced sentiments. Example: "So I'm not a big fan of Boll's work, but then again, not many are. I enjoyed his movie, Postal...."
Sentiment	Related Review's Polarity, either Positive or Negative

Lexicons used in this research are Global Vectors for Word Representation (GloVe) which is used as a reference to map out vectors in the embedding layer of Neural Network. GloVe maps term in semantic vector space log-bilinear models from co-occurrence matrix. Examples of real probabilities in GloVe corpus are :

Table 2. Example of word occurrence probabilities in GloVe [22]

Probability and Ratio	K = solid	k = gas	k = water
P(k jice)	1.9 x 10 ⁻⁴	6.6 x 10 ⁻⁵	3.0 x 10 ⁻³
P(k steam)	2.2 x 10 ⁻⁵	7.5 x 10 ⁻⁴	2.2 x 10 ⁻³
P(k jice)/P(k steam)	8.9	8.5 x 10 ⁻²	1.36

Where P(k|j) is a co-occurrence probability of two words and k is a word shown in the example. Solid structured objects such as ice are more probable in co-occurrence with its type, 'solid,' and also 'steam with 'gas.' With co-occurrence probabilities of certain terms being close from one another or not, GloVe can map out words as shown in figure (3).

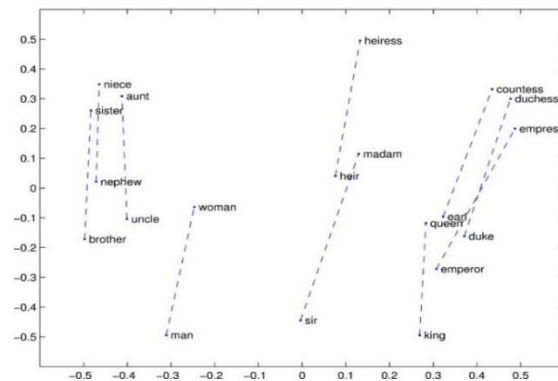


Figure 3. Map out of related terms in GloVe representation [22]

Detailed research can be found in study [22] on how vector is mapped in GloVe. However, this study focused on how GloVe performed well on sentiment datasets as it maps out the sentiment orientation of a term and so on. GloVe model used in this research is a pre-trained common crawl model with properties shown in table (3).

Table 3. GloVe properties used in this research.

Properties	Description
Tokens	840 Billion Tokens
Vocabulary Size	1.9 Million
Vector Size	300 Dimension
File Size	5,25 GB

These properties are used to max out performance as more vocabulary size with best as possible dimension will withhold the algorithm performance, but increase in sentiment analysis quality of input data. [22] Another Lexicon Used in this research is SentiWordNet in Bias Processing strategy and Semantic-Orientation Calculator method. SentiWordNet is a lexicon or dictionary that has a sentiment value towards a word and the sense or meaning contained in the word [4]. SentiWordNet version used in this research is 3.0 as it is Lexicon’s latest version. In Lexicon-based analysis, a SentiWordNet (SWN) dictionary is proposed with feature extraction using Part-Of-Speech Tag or POS-Tag [4]. POS-Tag is used to give Part-Of-Speech to text, such as nouns, verbs, and so on. A word may have a different sentiment value depending on its respective POS [4]. SentiWordNet includes a synonym ring or synset. Synset is a group of words that have a similar relationship of synonyms and senses. There are identical word sets with individual sense numbers, associated set POS-Tags, scores that determine the polarity of these word sets in positive and negative scores, and gloss or sense descriptions [4]. Example entries of SentiWordNet lexicon are shown in table (4).

Table 4. Example of SentiWordNet 3.0 word entries.

SWN POS Tag	Positive Score	Negative Score	Synset Terms	Gloss
a	0	0.75	Notional#1 imaginary# 1 fanciful#2	not based on fact; unreal;...
a	0.5	0.25	Fanciful#3	Having a curiously intricate quality

2.2. Preprocessing

Preprocessing is a step in preparing the data before the data is processed for sentiment analysis classification [13]. Case folding makes all letters in lowercase, “Unused Characters Removal” erasing junk characters (punctuation, numbers, URL, tags), tokenization separates sentences into several tokens, “Stopwords Removal” removes unimportant words, and Stemming, which is eliminating word affixes. Preprocessing steps are shown in Table (5).

Table 5. Examples of preprocessing results performed [10]

Description	Input	Output
Case Folding,	“One Of The Other Reviewers “	“one of the other reviewers“
Unused Character Removal,	“oz episode 100, you'll be hooked! http:bit.ly/film “	“oz episode, you'll be hooked.“
Tokenization	“one of the other reviewers “	[one, of, the, other, reviewers]
Stopwords Removal	[1, oz, episode, you'll, be, hooked]	[hooked]
Stemming	[other, reviewers, mentioned, watching, episode, you'll, be, hooked]	[other, reviewer, mention, watch, episode, you'll, be, hooked]

The research method starts from preprocessing data to clean the input data, and then there is feature extraction so that the cleaned data can be processed with related approaches. Classification is done to predict sentiment towards the test data, and finally, the evaluation process.

2.3. Feature Extraction Methods

After all of the necessary string-related preprocessing is done, Feature extraction is carried out to define the input of model listed in this research. Feature Extraction methods used in this research are Penn POS-Tagging for SO-Cal, SentiWordNet POS-Tagging for Bias Processing Method, and word Embeddings for Artificial Neural Network. Output of this process will be the input of related classifier listed. Part-Of-Speech Tagging in definition is a process to tag a word in review or a document by adjectives, nouns, verbs, and adverbs [25]. POS will determine how a word is represented in certain context and sentiment so that its sentiment polarity can be calculated by lexicon-based methods. POS-Tagging is also done to prepare the dataset for both Lexicon-based and Neural Network-based analysis. In this stage, preprocessing is carried out using POS-Tagger. POS-Tagger will generate Penn POS-Tag affixes. The POS used in this research is Penn Treebank English POS Tag that used commonly in English lexicon based sentiment analysis. List of POS-Tags used in the research is shown in table (6).

Table 6. List of POS-Tags Defined in Penn POS Tag [24]

POS-Tag	Description	Indication
NN/NNS/NNP/NNPS	Nouns	Aspect
JJ/JJR/JJS	Adjectives	Sentiment
VB/VBD/VBG/VBN/VBP/VBZ	Verbs	Sentiment
RB/RBR/RBS	Adverbs	Sentiment

Penn Treebank English POS-Tag introduces the indication of certain words being labeled as different POS-Tag Descriptions. Nouns considered as aspect, or word that associates with a sentiment around them, and Adjectives, Verbs, and Adverbs being a sentiment indication, or how the word should be portrayed. This tags later will be used as the score indexing for different lexicon-based method dictionaries, and used as input in SO-Cal Scoring. Second feature extraction used in this research is SentiWordNet POS-Tagging. SentiWordNet has 4 POS-Tags, namely adjective (a) or adjective, adverb (r) or adverb, verb (v) or verb, and noun (n) or noun [4]. The most frequently used POS-Tag is Penn-POS-Tag, and research [4] proposes to select SWN-POS-Tag for the SWN dictionary with the following comparison [10]. Conversion POS between SWN and Penn is shown in table (7).

In the comparison shown in table (6), SWN-POS-Tag generalizes POS-Tagger from Penn-POS-Tag to be more straightforward and processed by SentiWordNet. After declaring the rule of SWN POS-Tag, conversion preprocessing is done, shown in table (8).

Table 7. Comparison between SWN POS-Tag and Penn POS-Tag.

Penn POS	Description	SWN POS
JJ	Adjective	a
JJR.	Adjective, comparative	a
JJS	Adjective, superlative	a
RB	Adverb	r
RBR	Adverb, comparative	r
RBS	Adverb, Superlative	r
VB	Verb, base form	v
VBD	Verb, past tense	v
VBG	Verb, present participle	v
VBN	Verb, past participle	v
VBP	Verb, non 3 rd person singular present	v
VBZ	Verb, 3 rd person singular present	v
NN	Noun, singular	n
NNS	Noun, plural	n
NNP	Proper noun, singular	n
NNPS	Proper noun, plural	n

Table 8. Example of POS-Tagging with Input and Output

Input	Penn POS Tag	Output (SWN POS-Tag)
[other, reviewer, mention ,you, hooked	[JJ, NN, VB ,NN, VBD, ...]	[a , n , v, n , v, ...]

The research method starts from preprocessing data to clean the input data, and then there is feature extraction so that the cleaned data can be processed with related approaches. Classification is done to predict sentiment towards the test data, and finally, the evaluation process. This research also proposed the concept of Embedding layers or vector representations of a word that determines how close a word is to another word in polarity [12]. This is used in Artificial Neural Network Classification. Word Embeddings represent the number of words contained in one review in the form of word vectors and will later be used to train the data. Word Embeddings have a vector calculation reference according to the dictionary used. Examples are shown in table (9).

Table 9. Example of Word Embeddings implementation with GloVe.

Review	Word 1	Word 2	Word	Sentiment
R1	Good [0.52, 0.82 , 1.44 , 0.92,]	Interestin g [0.59, 0.24, 1.25, ...]	...	Positive
R2	Trash [-0.76, -0.81,]	Funny [-0.25, 0.25,]	...	Negative
....

For every word contained in a review or a document, vector representation of related words will be mapped according to its sentiment in the lexicon. For example in R1 most of the terms is positively based and mapped out as Good [0.52, 0.82 , 1.44 , 0.92,], Interesting [0.59, 0.24, 1.25, ...], Gripping [-0.42, 0.55, 1.00, ...], until the end of R1. This mapping indicates that R1 mostly has Positive terms, and this vector representation will help ANN calculations to determine sentiment polarity.

2.4. Artificial Neural Network Classification

After the preprocessing stage, an Artificial Neural Network-based analysis was carried out. In the Artificial Neural Network method, there is feature extraction with Word Embeddings, then followed by Hyperparameter Tuning and Artificial Neural Network classification. This analysis is separate from the Lexicon-based approach. Artificial Neural Network is a Machine Learning classification method inspired by the structure of the human brain by categorizing input values according to the neurons (cells in the brain) that communicate with each other using synapses [8]. The characteristic of Artificial Neural Network is that ANN can handle sequences of inputs better than methods in Lexicon-based method as input reviews mostly in the form of sequences. ANN has three layers of a structure named Input Layer, Hidden Layer, and Output Layer. The hidden layer serves to raw input data entering the network and then included in the neuron layer. Hidden Layer functions to process data determined by the input data, weight, and relation between them. Output Layer determines the output data based on the traversed synapses' weight and the relationship with the Hidden Layer [8].

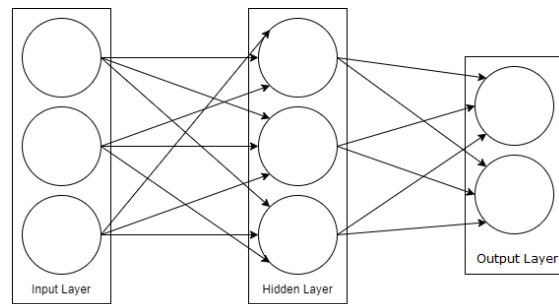


Figure 4. General Structure of Artificial Neural Network [8].

In general, every hidden and output node has the same architecture, except for the input layer, which stores data entered into the ANN algorithm. According to [19], ANN has the following architecture:

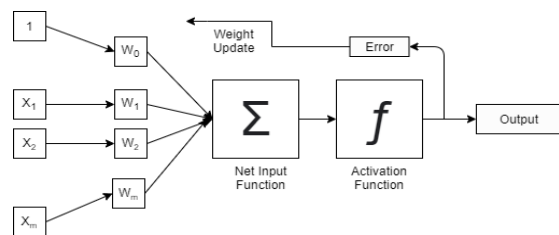


Figure 5. ANN Architecture with Neuron Functions [19]

In Figure (5), m describes the total number of input data entered. The representation of m in this study is the total number of reviews contained in the related dataset, 50.000 samples. x_m describes running input signal, which contains the input data to be entered and w_m describes the running weight contained in each synapse

with positive or negative values. Neuron has a process that is divided into several functions, namely Net Input Function and Activation Function. The Net Input Function serves to multiply the input signal by the synapse weight associated or connected to a neuron. The formula for the Net Input Function applies [19] :

$$n_k = \sum_{j=1}^m w_{kj}x_j \tag{1}$$

Where x_1, x_2, \dots, x_m are the input signals, $w_{k1}, w_{k2}, \dots, w_{km}$ are the weights of each synapse associated with neuron k, and n_k is referred to as the Net Input Function associated with several input signals in linear combinations. After the Net Input Function, an Activation Function functions to convert the output obtained into a non-linear function to limit the output range or distance of the Neural Network. The basis of the Activation Function formula that pays attention to bias is:

$$y_k = f(n_k + b_k) \tag{2}$$

Where y_k is the output signal from neurons, n_k is the Activation Function and b_k Is the bias of k neurons. The bias here has a constant form that can shift left or right to get the suitable model for the entered data. Activation function process Net Input Function output with the addition of bias depending on what activation function is used in the structure. The structure proposed in this research is Convolutional Neural Network Structure similar to Document Embeddings in study [23]. The design consists of the Input layer, Embedding Layer, two hidden layers with configurable activation functions and parameters, and the output layer. Design shown in figure (6)

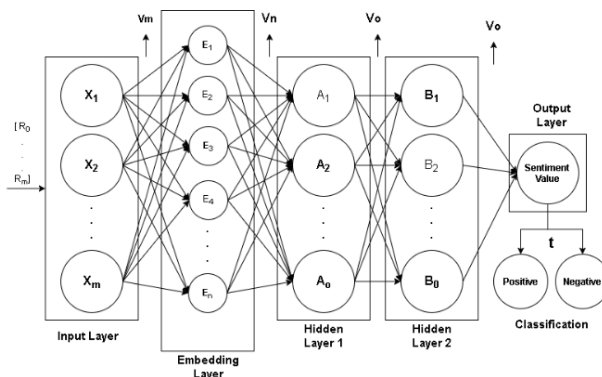


Figure 6. Proposed structure of ANN

R_m is a raw review dataset passing through the input layer with later encoded into input signals (x_1, x_2, \dots, x_m) and later processed into vector (E_1, E_2, \dots, E_n) by weight (V_m and V_n) and biases in the embedding layer. The weight used in this study's embedding layer is based on the GloVe (Global Words for Vector Representation). After the vector is flattened into embedding layer vectors, Forward Propagation and Backpropagation [19] are used to process forward the vectors by activation

functions. Forward Propagation is a calculation from the input layer's beginning to the output layer's end. In iterations for all data trains, the measure of the Net Input Function, or vector representation for the Activation Function by multiplying the input signal (x_1, x_2, \dots, x_m) with the specified weight ($w_{k1}, w_{k2}, \dots, w_{km}$) [19] model, and later hidden layer weights for each node. (A_1, A_2, \dots, A_o) indicates which nodes are activated based on how strong the value is, and $B_1 - B_o$ values are dependent on what nodes are activated in the first hidden layer. In the end, it will calculate the overall weight from the last hidden layer, outputting the sentiment value.

Furthermore, the Backpropagation Algorithm is used to improve the previous weight to get results close to the target value. This calculation is done by iterating backward from the last layer to reduce redundancy. In the backward iteration of Backpropagation, bias and weight updates are made to all nodes contained in the data train. Backpropagation iteration has the nature of Feed-forward Computation iteration, which calculates the Activation Function in equation (2) back to the Hidden Layer to update the associated bias and node weights so that the output data has minimal error [19]. The error function used in this study is binary cross-entropy because the labels contained in this study only have a binary representation, or only two representations, namely positive and negative. The error function in binary cross-entropy is as follows [22]:

$$CE = -\frac{1}{M} \sum_{m=1}^M [y_m \times \log(h_{\theta}(x_m)) + (1 - y_m) \times \log(1 - h_{\theta}(x_m))] \tag{3}$$

M is the amount of data contained in train data, y_m is the target label for the train data m , x_m is the input data and h_{θ} is the model used with weight θ [11]. BCE calculations on formula (3) are carried out to adjust the weight used and compare if the output of the related model produces a number that is close to the positive or negative binary label representation in the resulting margin of error [11]. Errors that accumulate in the model will later be represented by Binary Cross Entropy function verbosed as a loss function compared when hyperparameter tuning is performed. Then, after the model is built, hyperparameter tuning is introduced to configure what model is the best for the dataset. Hyperparameter Tuning is done to get maximum performance, and the output generated from Hyperparameter Tuning is an Artificial Neural Network model that can be used to do full classification. The Hyperparameters contained in the Artificial Neural Network model are activation function, dropout rate, and weight initialization mode [17]. Activation Function serves to limit the range of output distance of the Neural Network [16]. This parameter is used to activate certain

nodes that influence the output, weights, and biases. Dropout Rate determines which neurons are dropped (dropped-out) during training and is configured to reduce overfit [16]. Also, Weight Initialization Mode is configured to determine the weight initialization mode when training with the new model to avoid bias [16].

More training specified parameters also exist, such as training optimizer, learning rate, batch size, and epochs. Training Optimizer t determines the training optimization mode that minimizes the loss function, Learning Rate serves to pick the pace of the model learned in each iteration, Batch Size picks the number of data samples used in each iteration, and lastly epoch, or a Parameter that determines the number of returns of the entire dataset with the associated model stop [17]. Defined several parameters used in this research shown in table (10).

Table 10. Defined Hyperparameter Tuning Configuration Values.

Hyperparameter	Value
Batch Size	10,20,40,60,80,100,256
Epochs	10,50,100
Training Optimizer	SGD, RMSProp, Adagrad, Adadelta, Adam, Adamax, Nadam
Learning Rate	0.001, 0.01, 0.1, 0.2, 0.3
Weight Initialization Mode	Uniform, Lecun, Uniform, Normal, Zero, Glorot Normal, Glorot Uniform, He Normal, He Uniform
Activation function	ReLU, Softmax, Softplus, Softsign, Tanh, Sigmoid, Hard Sigmoid.
Weight Constraint	1,2, 3,4,5
Dropout Rate	0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8
Number of Neurons	1,5,10,15,20, and 25

Hyperparameters were set after testing the dataset with random search. Hyperparameter Tuning is done earlier in the experiment to prevent time loss in executing other code runs. Later, these parameters will be used in model building to get the best performance. Several Activation Functions proposed for Hyperparameter Tuning are Linear Function, Rectified Linear Unit (ReLU), Sigmoid, Hard Sigmoid, Hyperbolic Tangent Function (Tanh), Softmax, Softsign, and Softplus. [17]. Proposed a sigmoid activation function to handle Binary Classification problems such as sentiment classification. A sigmoid function performs with the application of non-linear derivatives as follows :

$$f(x) = \left(\frac{1}{1+exp^{-x}} \right) \tag{4}$$

In equation (4), x is the input value against the exponential formula as a non-negative derivative [17]. Any output close to the negative derivative will be scaled into a negative x gradient with a positive y gradient.[17]. This equation is used in the second hidden layer. The Sigmoid function is one of the best functions

in applying the Binary Classification model such as sentiment analysis and is also a logistic function. [17]. After determining the model with Hyperparameter Tuning, classification will be carried out based on ANN. After sharing the data, training will be carried out on the ANN model, generating positive and negative sentiment scores. The inputs in this classification are Word Embeddings vectors with Sentiment score outputs with positive or negative layer outputs. An example of testing data classification is described in table (11).

Table 11. Example of ANN classification in data tests.

Review	Word Embeddings	Sentiment Score	Sentiment Prediction
R1	Good [0.52, 0.82 , 1.44 , 0.92, ...], Interesting [0.59, 0.24, 1.25, ...] Gripping [-0.42, 0.55, 1.00, ...]	0.6	Positive
....

2.5. Bias Processing Method Classification

After the preprocessing stage, Lexicon-based analysis will also be carried out. In the Lexicon-based method, there is POS-Tagging feature extraction to change the reviews that have been included in the Preprocessing to add POS. This stage is followed by classification using Bias-Aware-Threshold and Semantic Orientation Calculator. The characteristic of Bias Processing Method Classification is that from previous researches such as study [4], this method yields the lowest Polarity Bias Rate compared with existing strategies such as SO-Cal and Bias Aware Thresholding or BAT. This analysis is separate from the Artificial Neural Network method. Lexicon Based Analysis analyses a list of words or phrases whose valence value is determined as a marker of polarity in a phrase or word [9]. The number of valence values is determined by [+v, -v], where the value of v is the non-negative value of the polarity marker. Lexicon-based analysis was carried out based on the dictionary that became the reference for the valence score. In a list of words or phrases (documents), the Lexicon-Based Analysis method produces two scores, namely a positive score $S_A^+(d)$ and a negative $S_A^-(d)$ which determines the following decision rules:

$$C_A(d) = \begin{cases} +, & \text{if } S_A^+(d) > S_A^-(d) \\ -, & \text{otherwise} \end{cases} \tag{5}$$

$C_A(d)$ indicates the polarity of the list of words or phrases [9]. Formula (5) applies the rules, which if the positive score in the lexicon-based algorithm is more than the negative score, then positive polarity applies and vice versa. The calculation used in this method is SentiWordNet, which SynsetScore is used to determining the sentiment score against a Synset. SynsetScore can be calculated by subtracting the positive and negative scores of a Synset contained in the SWN. Then applies:

$$SynsetScore = PosScore - NegScore \tag{6}$$

The scores for specific terms with POS-Tag can be calculated with TermScore, not apart from the words belonging to several synsets. The TermScore formula applies:

$$TermScore = \frac{\sum_{r=1}^n SynsetScore(r)/r}{\sum_{r=1}^n 1/r} \tag{7}$$

Where r represents the sense number as the dividend for both sums. In the sentiment analysis classification, word tokens will be generated that have been affixed by the SentiWordNet POS-Tag, and the overall sentiment score will be calculated using SentiScore. Before calculating the SentiScore, Positive Score and Negative Score sums are determined first. If the TermScore of a term is negative, then NegScore is added from the TermScore, and otherwise. After the sums of both scores, a SentiScore is calculated by the sum of both scores. Therefore, SentiScore calculation applies:

$$PosScore(r) = \sum_{i=1}^m TermScore(T_i) \tag{8}$$

$$NegScore(r) = \sum_{i=1}^m TermScore(T_j) \tag{9}$$

$$SentiScore(r) = PosScore(r) + NegScore(r) \tag{10}$$

Table 12. Example of SentiWordNet 3.0 scoring.

Input (Tokens)	SWN POS Tag	Output (Term Score)
[other, reviewer, mention, you, hooked]	[a, n, v, n, v]	[- 0.125, 0, 0.3, 0, 0.541]
[masterpiece, result, matters]	[n, n, n]	[0.5, 0.133, -0.215]
...

Table (12) shown the input and output of SentiWordNet Scoring. SentiWordnet Scoring takes input from preprocessed tokens on a review, with SentiWordNet POS-Tag as regular POS-Tag. From previous POS-Tag preprocess, a indicates adjective tokens, r indicates adverb, v indicates verb, and n indicates noun. Then the TermScore will be searched with related input. After SentiScore calculation, weight and threshold parameters are introduced to reduce additional sentiment bias. This method is called the Bias Processing Strategy by study [4], and it is as follows :

$$SentiScore'(r) = \alpha x PosScore(r) + (1 - \alpha) x NegScore(r) + t \tag{11}$$

Where α is the weight parameter ranges from 0 to 1, and t is the threshold parameter. This weight and threshold are learned from a small training dataset. If the sentiment score is above zero, then the sentiment is positive, and otherwise. To get the most of Accuracy and PBR presented as, threshold and weight parameter is trained according to higher numbers in Accuracy, and

lower PBR. The algorithm used in determining Bias Processing Method is shown in algorithm:

Bias Processing Method Algorithm

Input: Training Set $T = \{r_1, r_2, \dots, r_n\}$
Output: (α, t)

```

For  $\alpha_j$  in range (0,1) do
  Initialize  $S = \{S_k \mid S_k = 0, 1 \leq k \leq n\}$ 
  for  $r_i$  in  $T$ , do
     $S_i \leftarrow \alpha \times PosScore(r_i) + (1 - \alpha) \times NegScore(r_i)$ 
  end for

  Seek the  $t_j$  at which  $PBR = 0$  according to Algorithm 2
  Record  $\alpha_j, t_j, Accuracy(T, \alpha_j, t_j)$ 
end for

Seek the  $(\alpha_j, t_j)$  at which  $Accuracy(T, \alpha_j, t_j)$  is the highest
Return  $(\alpha, t)$  which  $Accuracy(T, \alpha, t)$  is the highest
    
```

The bias Processing Method algorithm requires a small training set and will be returning weight (α) and threshold. For every row in defined data train calculated the Bias Processing method then saved into an array. After all score from the data train is calculated with the sentiment, the algorithm seeks the best threshold by PBR, and record the weight, best threshold, the accuracy, to be later compared which accuracy is the highest. Moreover, the algorithm to determent the sentiment polarity of the BPM score is shown in algorithm:

Sentiment Polarity Determination Algorithm with specific threshold t

Input: Score set S (s_i is the sentiment score of corresponding review r_i from Training Set T, it is weighted with a specific weight α and assigned in Step 3-5 of Algorithm 1.); threshold t
Output: Label Pos or Neg for r in T

```

for  $S_i$  in S, do
  if  $S_i + t > 0$  then
    Polarity ( $r_i$ )  $\leftarrow$  Pos
  else
    Polarity ( $r_i$ )  $\leftarrow$  Neg
    
```

Sentiment polarity determination algorithm determines the sentiment from threshold and BPM Score. After weight and threshold are saved, classification is done from the best thresholds to validate which thresholds and weight combinations are the best performance. Classification examples are shown in the table (13).

Table 13. Example of BPM classification with various thresholds

Review	Input (Senti Score)	Sentiment	Threshold Prediction		
			T1	T2	...
R1	0.514	Positive	Neg	Pos	...
R2	-0.125	Negative	Pos	Neg	...
R3	0.9	Positive	Pos	Pos	...
...

2.6. Semantic Orientation Calculator (SO-Cal) Scoring

Semantic Orientation CALculator (SO-CAL) reduces bias by calculating the semantic orientation of a word or

phrase represented by a value. The semantic direction in SO-CAL is represented by the SO value [10]. The characteristic of SO-Cal is that this method uses more accurate lexicon structure with semantic orientation values rather than simple calculation of lexicon-based methods such as BAT or SentiWordNet Scoring.[10] The distance that SO-CAL can determine is -5 to +5 polarity numbers that have been found in the SO-CAL dictionary with the division based on the type of word noun, verb, adjective, and adverb, similar to SWN in the affixing of SWN-POS-Tag [4]. SO-CAL calculations depend on the kind of word classification that has been determined by the dictionary, namely SO-Carrying Word, Intensifier, or negator. An example of SO-Carrying Word can be seen in table (14) [10]:

Table 14. Example of SO-Carrying Word with SO Value.

Word	SO Value
Monstrosity	-5
hate(noun and verb)	-4
Disgust	-3
Masterpiece	+5
Relish	+4
Interesting	+3
...	...

The table above shows SO-Carrying Word, or words whose calculations are added directly to the observations of documents, sentences, or words. For example, "hate masterpiece" will result in the calculation $(-4 + 5) = 0$, and "relish disgust" will result in $(+4-3) = 1$. The other types of word classification are modifiers or intensifiers that strengthen a sentiment when used. The intensifier can be used to multiply the value of the SO-Carrying Word by its percentage. Examples of intensifiers can be seen in table (15) [10] :

Table 15. Example of Intensifier with modifier values or percentage.

Word	SO Value
Slightly	-50
Somewhat	-30
Very	+25
Really	+15
...	...

The table above shows the intensifier or words whose calculations are multiplied by observing documents, sentences, or related words (as modifiers). Negation is the last word classification type of SO-CAL, with the definition that if a sentiment word coexists with the word negation (“not,” “nobody,” “none”), the SO Value of that word will be multiplied by negative. The SO-Cal Scoring stage in the process chart contains a classification based on SO-CAL after the data has gone through the Preprocessing phase and affixed with penn POS-Tag. The input in this process is the word token from a review, then it is calculated based on the SO-CAL calculation. Examples of Classification using SO-Cal are shown in table (16):

Table 16. Example of process results using SO-Cal.

Input (Tokens)	POS-Tag	Output (SO-Cal Score)
----------------	---------	-----------------------

['somewhat', 'interesting', 'not', 'masterpiece']	[RB, JJ, JJ, NN]	$(3 \times (100 \% - 30\%)) + (-5) = 2.1 - 5 = -2.9$
...

The polarity determination of the SO-CAL results is by looking at the orientation of the SO-CAL calculation results. If the direction has a negative number, then the polarity of the document, sentence, or word is negative, and vice versa. [10]

2.7. Weight and Threshold Training

Weight and threshold training is done with the ROC-AUC Curve learning method. This method uses a different threshold to select the best Accuracy and threshold from score predictions. ROC-AUC works on Sensitivity and Specificity (True Positive Rate plotted against False Positive Rate) measurement, and find points above linear random search performance for the best prediction model.[24] The plotted area under the curve against Sensitivity and Specificity, the higher the curve, the better the performance is.

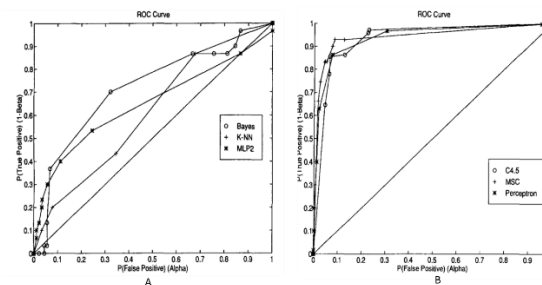


Figure 7. A: Worst performance ROC Curve in Research [24], B: Best Performance on study [24]

2.8. Hybrid Classification Method

After features is extracted from ANN, BPM, and SO-Cal, Hybrid Classification is carried out to improve the model as it derives the characteristic of selected algorithms. Majority vote system is done to validate sentiments across algorithms, which if certain sentiment has more sum of values than the other, then the prediction produced will be that majority, and otherwise.

Table 17. Example results of Hybrid Classification with Features

Review	ANN Features	BPM Features	SO-Cal Features	Sentiment Prediction
R1	Positive	Negative	Positive	Positive
R2	Negative	Negative	Positive	Negative
R3	Positive	Negative	Negative	Negative
R4	Negative	Positive	Positive	Positive
...

From table (17) shown examples of hybrid classifications with example features listed. Majority vote concludes that R1 and R4 has Positive majority with two features being Positive, therefore the sentiment it carries is Positive. Also from R2 and R3, the Sentiment carried were Negative, as majority vote from both

Reviews are two votes in Negative sentiment, therefore the sentiment it carries is Positive.

2.9. Data Evaluation

After knowing the parameters to be entered into the Artificial Neural Network model, training data iterations, data testing, and evaluation are carried out to determine the performance of the model that has been made. Details of the performance of the models that have been made are shown in table (13).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (12)$$

$$Polarity Bias Rate = \left| \frac{FP - FN}{TP + FN + TN + FP} \right| \times 100\% \quad (13)$$

TP denotes True Positives or the number of correct positive predictions on TP and TN. FP represents False Positives or the number of incorrect positive predictions. TN represents True Negatives, the number of accurate negative predictions, and FN denotes False Negatives or the number of inaccurate negative predictions. Accuracy is to map out correct predictions performance towards the whole dataset, and Polarity Bias Rate used to quantify how many rate of bias from a dataset with the best value being close to zero [9].

3. Results and Discussion

3.1. Artificial Neural Network Model Performance

After Hyperparameter tuning process is done, model building is carried by the set of tuned parameters. From the first iteration of Hyperparameter tuning yield results of best parameters shown in table (18).

Table 18. Hyperparameters yield in first iteration of Tuning

Hyperparameter	Value
Batch Size	80
Epochs	50
Training Optimizer	SGD
Learning Rate	0.2
Weight Initialization Mode	Glorot Uniform
Activation function	Linear
Weight Constraint	3
Dropout Rate	0.6
Number of Neurons	25

After knowing the parameters to be entered into the Artificial Neural Network model, training data iterations, data testing, and evaluation are carried out to determine the performance of the model that has been made. The details of the performance of the models that have been made are as shown in table (19)

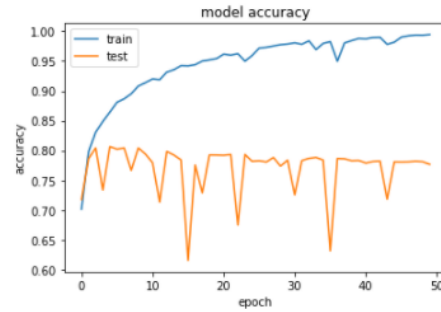


Figure 8. Model Accuracy on training and test data mapped by epoch in first iteration of hyperparameter

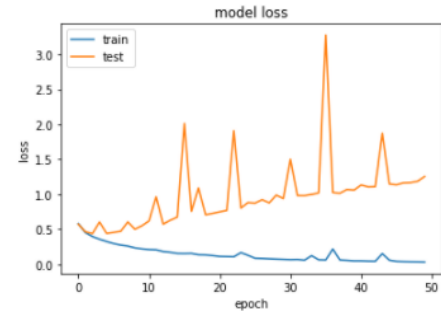


Figure 9. Model Loss on training and test data mapped by epoch on first iteration of hyperparameter

Table 19. ANN Performance on First Hyperparameter Tuning

Method	Best Threshold	Accuracy	PBR
ANN	0.01038	0.93396	0.00052

The evaluation of the model shown in Figures (8) and (9) indicates spiked patterns throughout the epoch in training. This sharp pattern is because the linear activation function tends to input without flattened the gradient descent. This model also yields worst results according to the model loss and accuracy mapping that train model is better in accuracy than test model, which indicates overfitting. Because of such reason, therefore second iteration of Hyperparameter tuning is introduced with simplified model, removing two parameters such as learning rate and weight initialization mode. This yields parameter shown in table (20).

Table 20. Hyperparameters yield in second iteration of Tuning

Hyperparameter	Value
Batch Size	256
Epochs	10
Training Optimizer	SGD
Activation function	Tanh
Weight Constraint	5
Dropout Rate	0.6
Number of Neurons	30

Parameters from the second iteration is more reasonable than the first one indicating an improvement in model building. And after defining second Hyperparameters, the parameters selected used in second iteration of ANN

model. The performance of related model is shown in table (21).

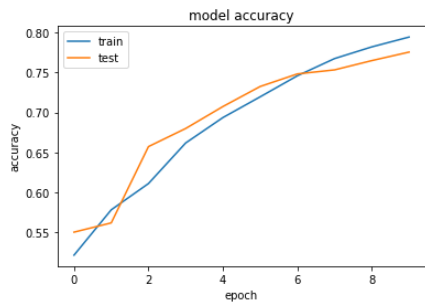


Figure 10. Model Accuracy on training and test data mapped by epoch on second iteration of hyperparameter

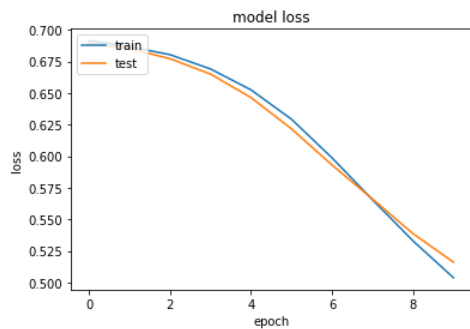


Figure 11. Model Loss on training and test data mapped by epoch on second iteration of hyperparameter

Table 21. ANN Performance on Second Hyperparameter Tuning

Method	Best Threshold	Accuracy	PBR
ANN	0.50963706	0.77396	0.01244

The evaluation of the model shown in Figures (10) and (11) indicates decent performance of ANN models without underfit and overfit. The performance of this model is 77.39% and it is far from state-of-the-art dataset accuracy, which is 97.4% in the same dataset. Improvement can be seen as changing the embeddings method to document and more complex Hyperparameter Tuning to avoid big gradient descent spiked data performance. This method yields model results of decently mapping the correct predictions, with normal PBR at 0.01244.

3.2. Bias Processing Method Performance with SentiWordNet Scoring

After calculating the Bias Processing Method score, the Threshold value will be determined from the SentiWordNet value from the threshold determining algorithm proposed. After obtaining the best thresholds from every weight, a classification will be carried out based on the SWN Classifier, or negative and positive sentiment determination based on the related entry. The results of the Bias Processing Method performance obtained are shown in table (22).

Table 22. BPM results based on a predetermined threshold.

Weight	Best Threshold	Accuracy	PBR
0.0	1.0	0.52246	0.45798
0.1	1.7125	0.58008	0.3056
0.2	3.05	0.60078	0.05746
0.3	1.57500	0.63266	0.00062
0.4	0.5	0.66326	0.0009
0.5	0.00649	0.65276	0.16588
0.6	3.55271	0.57416	0.38312
0.7	1.3322	0.51868	0.47304
0.8	4.4408	0.50422	0.4943
0.9	5.5511	0.50048	0.49904
1.0	71.319	0.5	0.5



Figure 12. Performance Plot of BPM against Weight

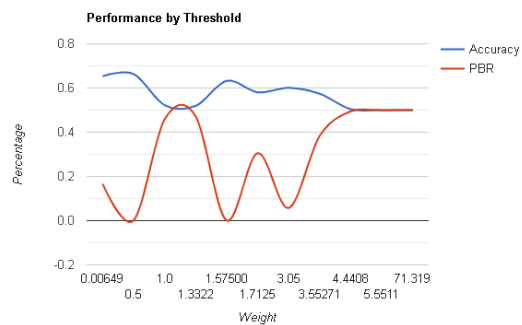


Figure 13. Performance Plot of BPM against Threshold

From figure (12) shows that performance is heavily influenced by a predetermined weight set in the classification algorithm. Figure (12) shows peaked performance from lesser weight until stagnant in midway point, then dropped as soon as it reaches the end of the weight measurement. The threshold plot in figure (13) shows that threshold did not necessarily affect performance, but more in the accuracy field. Even though maxed outperformance in Bias Processing Method is low, it is decent compared to earlier research with simple machine learning such as SVM and Linear Regression-only classification. Various thresholds also show that Bias Processing Method is at best reducing bias reaching numbers of 0.0009 and the accuracy of 66.326% with 0.4 weight and 0.5 on threshold. This performance indicates the model is average at best to map out correct predictions in both Positive and Negative sentiment, with excellent performance in rate of bias in the dataset.

3.3. SO-Cal Threshold Performance

After calculating the SO-Cal Scoring, the best threshold will be determined based on the ROC-AUC curve. A certain threshold is used to carry out the SO-Cal classification method, or determine negative and positive sentiment based on the related entry. The results of the Bias Aware Thresholding performance obtained in table (23)

Table 23. SO-Cal result based on a predetermined threshold.

Method	Best Threshold	Accuracy	PBR
SO-Cal Scoring	0.02516	0.75606	0.00006

Table (23) shows the classification results of SO-Cal scoring with 75.606 accuracy, 0.00006 PBR, and 0.02516 as the best threshold. This performance indicates the model is fairly good at map out correct predictions in both Positive and Negative sentiment, with low rate of bias in the dataset.

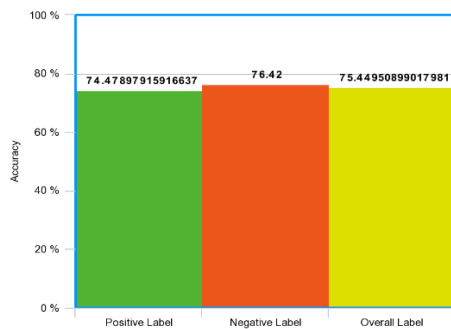


Figure 14. Performance of SO-Cal analysis in labeling dataset.

Figure (14) shows that SO-Cal analysis is fairly decent at classifying labels. Overall spread is fairly even in classifying negative and positive labels with good PBR in reducing bias. SO-Cal performance compared to Multilevel Semantic Network in the same dataset is close at 1.2% accuracy increase with SO-Cal.

3.4. Performance Comparison of Artificial Neural Network, Bias Processing method, SO-Cal, and Hybrid

After experimenting with several of the proposed sentiment analysis methods, the overall performance of each technique will be compared. For the Artificial Neural Network method, the model that has been formed from Hyperparameter Tuning will be reached. In contrast, the best-selected threshold for Bias Processing Method will be achieved with SentiwordNet and Semantic Orientation Calculator. The comparison among the related methods is shown in table (24)

Table 24. Performance Comparison of ANN, BPM, and SO-Cal.

	ANN	BPM Scoring	SO-Cal Scoring	Hybrid
Accuracy	0.7739	0.66326	0.75606	0.76264
PBR	0.0124	0.0009	0.00006	0.00512

From the performance table, SO-Cal and BPM are best for handling Bias Rate at 0.009 for BPM, and 0.00006 for SO-Cal. Also, ANN has the highest accuracy at 77.39%. Both models risked in on ANN being heavily configured, and Bias Processing Method is one of the low-performing models. Hybrid performance is good, as it is a good balance of PBR and accuracy.

4. Conclusion

Sentiment analysis is an analytical method that focuses on analyzing a person's opinions, sentiments, behavior, evaluations, surveys, and emotions. In this study, a dataset was examined in the form of an IMDb film review dataset with a total number of review respondents of 25.000, divided into positive and negative reviews. The method used in this research is Artificial Neural Network with the introduction of Hyperparameter Tuning to improve performance and reduce bias. Also, other methods in this research are Lexicon Based Method with Bias Processing Method in SentiWordNet and Semantic Orientation Calculator. Based on the research done, it can be concluded that Artificial Neural Network has the best performance on sentiment classification with bias processing method. Multiple layers of calculation through the classification process on Artificial Neural Network affected the whole classification performance as inputs undergo complex procedures and dataset personalization on Hyperparameter Tuning. Results generated from the related methods are pretty good, namely 77.39% for the Artificial Neural Network method, 66.32% for Bias Processing Method, 75.606% for the Semantic Orientation Calculator method, and 76.26% for Hybrid Classification. Accuracy performances on all methods tested have not much improved than listed approaches in previous publications as popular as the IMDb dataset is. This research focused on bias evaluation in every form suggested and the best performance on tested methods. Results showed that the best performance between the three methods listed was SO-Cal in 0.00006 PBR value. Overall this research can be used as future references to similar methods and dataset analysis. More advanced model configuration in Artificial Neural Network can improve the model, and much complex lexicon models will be a future in the research topic.

References

- [1] N. V R, "Predicting movie success based on imdb data," International Journal for Research in Applied Science and Engineering Technology, vol. V, no. X, pp. 504–507, 2017.
- [2] Y.-L. Chiu, K.-H. Chen, J.-N. Wang, and Y.-T. Hsu, "The impact of online movie word-of-mouth on consumer choice," International Marketing Review, vol. 36, no. 6, pp. 996–1025, 2019.
- [3] K. Kumar, B. S. Harish, and H. K. Darshan, "Sentiment analysis on imdb movie reviews using hybrid feature extraction method," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 5, no. 5, p. 109, 2019.

- [4] H. Han, Y. Zhang, J. Zhang, J. Yang, and X. Zou, "Improving the performance of lexicon-based review sentiment analysis method by reducing additional introduced sentiment bias," *PLOS ONE*, vol. 13, no. 8, 2018.
- [5] V. Despotovic and D. Tanikic, "Sentiment analysis of microblogs using multilayer feed-forward artificial neural networks," *Computing and Informatics*, vol. 36, no. 5, pp. 1127–1142, 2017.
- [6] K. Z. Aung and N. N. Myo, "Sentiment analysis of students' comment using lexicon based approach," 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS), 2017.
- [7] H. Ha, H. Han, S. Mun, S. Bae, J. Lee, and K. Lee, "An improved study of multilevel semantic network visualization for analyzing sentiment word of movie review data," *Applied Sciences*, vol. 9, no. 12, p. 2419, 2019.
- [8] N. Pradhan, G. Rani, V. S. Dhaka, and R. C. Poonia, "Diabetes prediction using artificial neural network," *Deep Learning Techniques for Biomedical and Health Informatics*, pp. 327–339, 2020.
- [9] M. Iqbal, A. Karim, and F. Kamiran, "Bias-aware lexicon-based sentiment analysis," *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, 2015.
- [10] A. Hasan, S. Moin, A. Karim, and S. Shamshirband, "Machine learning-based sentiment analysis for twitter accounts," *Mathematical and Computational Applications*, vol. 23, no. 1, p. 11, 2018.
- [11] Y. Ho and S. Wookey, "The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling," *IEEE Access*, vol. 8, pp. 4806–4813, 2020.
- [12] A. Dahou, M. A. Elaziz, J. Zhou, and S. Xiong, "Arabic sentiment classification using convolutional neural network and differential evolution algorithm," *Computational Intelligence and Neuroscience*, vol. 2019, pp. 1–16, 2019.
- [13] A. Tripathy, A. Anand, and S. K. Rath, "Document-level sentiment classification using hybrid machine learning approach," *Knowledge and Information Systems*, vol. 53, no. 3, pp. 805–831, 2017.
- [14] Z. Shaukat, A. Zulfiqar, C. Xiao, M. Azeem and T. Mahmood, "Sentiment analysis on IMDB using lexicon and neural networks", *SN Applied Sciences*, vol. 2, no. 2, 2020.
- [15] Ö. Ican and T. B. Çelik, "Stock market prediction performance of neural networks: A literature review," *International Journal of Economics and Finance*, vol. 9, no. 11, p. 100, 2017.
- [16] C. Nwankpa, S. Eze, W. Ijomah, A. Gachagan, and S. Marshall, "Achieving remanufacturing inspection using deep learning," *Journal of Remanufacturing*, vol. 11, no. 2, pp. 89–105, 2020.
- [17] I. Kandel and M. Castelli, "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset," *ICT Express*, vol. 6, no. 4, pp. 312–315, 2020.
- [18] M. Feurer and F. Hutter, "Hyperparameter optimization," *Automated Machine Learning*, pp. 3–33, 2019.
- [19] J. Heaton, "Ian Goodfellow, Yoshua Bengio, and aaron Courville: Deep learning," *Genetic Programming and Evolvable Machines*, vol. 19, no. 1-2, pp. 305–307, 2017.
- [20] L. Shi and B. Chen, "A vector representation of dna sequences using locality sensitive hashing," 2019.
- [21] X. Song, G. Wang, Y. Huang, Z. Wu, D. Su, and H. Meng, "Speech-xlnet: Unsupervised acoustic model pretraining for self-attention networks," *Interspeech 2020*, 2020.
- [22] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.
- [23] T. Thongtan and T. Phienthrakul, "Sentiment classification using document embeddings trained with cosine similarity," *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, 2019.
- [24] A. Singh, "Review of 'text classification algorithms: A survey,'" 2021.
- [25] A. S. Shafie, N. M. Sharef, M. A. Azmi Murad, and A. Azman, "Aspect extraction performance with pos tag pattern of dependency relation in aspect-based sentiment analysis," 2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP), 2018.