Deep Learning Algorithms with Adam Optimization for Detecting of Cyberbullying Comments

M. Dharani[1], Dr. S. Sathya[2]

[1]*Research Scholar, Department of Computer Science, Vels Institute of Science, Technolgy and Advanced Studies (VISTAS), Chennai. dharavinay@gmail.com*
[2]*Assistant Professor, Department of Information Technology, Vels Institute of Science, Technolgy and Advanced Studies (VISTAS), Chennai. ssathya.scs@velsuniv.ac.in*

The unavoidable utilization of web-based entertainment stages, like Facebook, Instagram, and X, has essentially intensified our electronic interconnectedness. Additionally, these stages are presently effectively available from any area at some random time. Nonetheless, the expanded prominence of virtual entertainment has additionally prompted cyberbullying. The peculiarity of cyberbullying has spread and has become perhaps of the most concerning issue confronting clients of virtual entertainment locales and produced critical unfriendly impacts on society and the casualty specifically. The utilization of oppressive and aggressive language has decisively extended in the virtual entertainment and systems administration time [16]. Youngsters are to a great extent liable for it. The greater parts of youngsters who utilize web-based entertainment for correspondence are survivors of cyberbullying. Affronts on person-to-person communication sites lead to unfavourable network connections. These remarks encourage a rude environment in web. Most of the instruments and calculations used to appreciate it and decrease it are idle. Tracking down fitting answers for recognize and diminish cyberbullying has become important to relieve its adverse consequences on society and the person in question. To characterize such remarks in a commonsense manner, the article expects to distinguish procedures to perceive tormenting in text by looking at and trying different things with different methodologies [27]. We recommended a compelling calculation to perceive unfriendly and badgering remarks, and we inspected these remarks to guarantee their legitimacy. In this paper, we were utilized three different profound learning calculations with Adam Enhancer. The existing methods like DNN, ANN, and RBFN are combined with Adam optimizer to detect the cyberbullying comments. The results show that by choosing the best features, the suggested RBFN with Adam Optimizer increases the accuracy of cyberbullying detection.

Keywords: Sentiment analysis, Cyberbullying, Deep Neural Network (DNN), Artificial Neural Network (ANN), Radial Basis Function Network (RBFN), Adam optimization [2].

1. Introduction

Cyberbullying has emerged as a pervasive issue in the digital age, characterized by the use of electronic communication to intimidate, threaten, or harm individuals. Unlike traditional bullying, which typically occurs face-to-face, cyberbullying leverages online platforms such as social media, messaging apps, and forums to perpetrate abusive behavior [1]. This phenomenon has garnered significant attention due to its profound impact on victims' mental health, social interactions, and overall well-being [3] [21]. As social media platforms continue to evolve and expand their reach, understanding and detecting cyberbullying behaviors have become crucial for safeguarding users and promoting a safer online environment [13]. The purpose of detecting cyberbullying in social media for research is multifaceted. Firstly, it aims to contribute to the growing body of knowledge on cyberbullying by identifying its prevalence, patterns, and underlying factors within digital communities [6] [32]. By analyzing social media interactions, researchers can uncover common tactics and behaviors associated with cyberbullying, shedding light on its dynamics and impact [26].

Secondly, detection efforts serve a practical purpose in mitigating the harmful effects of cyberbullying. Early identification of abusive content allows platform administrators, educators, and policymakers to implement targeted interventions and support mechanisms for victims [12]. This proactive approach can potentially reduce the duration and severity of cyberbullying incidents, thereby fostering a safer online environment for users of all ages. Furthermore, research into cyberbullying detection methodologies contributes to the development of effective algorithms and tools [20]. These technological solutions enable automated or semi-automated detection of abusive content, complementing human moderation efforts on social media platforms. By refining detection algorithms through empirical research, stakeholders can improve the accuracy and efficiency of identifying cyberbullying instances, ultimately enhancing platform safety and user experience [8].

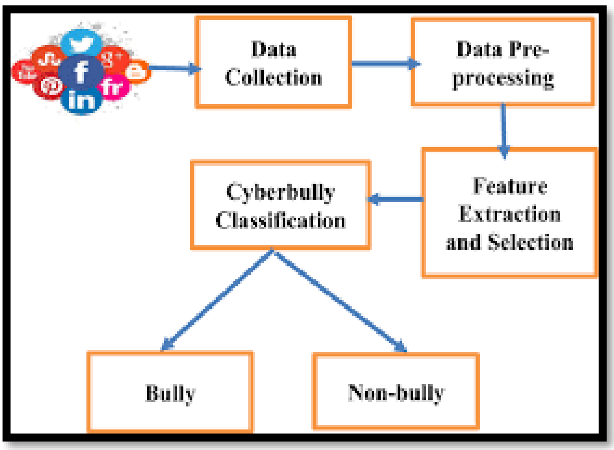

Figure 1. Developed model for classification of cyber bullying

2. Literature Survey

The consequences of the proposed model exhibit critical improvement in the presentation of grouping on all the datasets in contrast with late existing models. Much work has been finished on the opinion examination with Facebook [9].

Table 1. Comparison of literature

S. No	Author	Title	Methods	Result
1	Abulkarim Faraj Alqahtani and Mohammad Ilyas1	A Machine Learning Ensemble Model For the detection of Cyberbullying	Stacking model which includes five classifiers (DT, RF, LSVC, KNN, and LR) with Four different methods for feature extraction, namely BoW, TF-IDF, Word2Vec, and GloVe	From accuracy stacking model with GloVe features gives better result.
2	Ashwini Kumar1 and Santosh Kumar2	Optimized Deep Neural Networks Using Sparrow Search Algorithms for Hate Speech Detection	The LSTM model's boundaries have been improved with the assistance of the sparrow search strategy, which gives a precise record of the organization Design and boundary settings utilized by the model.	This exploration recommended a technique for hyperparameter improvement using the Long Short-Term Memory with Sparrow Search Algorithm (LSTM-SSA) model
3	Olawale Lukman Olaitan a, Adeniji Oluwashola David a* and Odejayi Adeniyi Michael	Deep Learning Approach for Classification of Tweets in Detecting Cyber Truculent	nigram model, bigram model, trigram model and the ngram model	The unigram model performed genuinely well with 96.3% precision.
4	Reem Albayari, Sherief Abdallah and Khaled Shaalan	Cyberbullying Detection model for Arabic text using Deep learning	various DL algorithms (LSTM, GRU, LSTM-ATT, CNN-BLSTM, CNN-LSTM and LSTM-TCN) were used	As a result of the models' evaluation, a hybrid DL model is proposed that combines the best characteristics of the baseline models CNN, BLSTM and GRU for identifying cyberbullying
5	Chandradeep Bhatt 1, Parul Goyal 2, Ghanshyam Prasad Dubey 3, Shalini Singh 4 and Vishal Kumar 5	Detection of cyber-bullying in social-media using classification algorithms of machine learning	Decision Trees, Naïve Bayes, Random Forest, Logistic Regression, K Nearest Neighbour, and Support Vector Classifiers with feature extraction. Count vectorizer and TF-IDF is used as feature extraction.	With temporal frequency - inverse document frequency for feature extraction, support vector machines give the accuracy of 91.98%.
6	Abdulkarim Faraj Alqahtani, Mohammad Ilyas	An Ensemble-based multi-classification machine learning classifiers approach to detect multiple classes of cyberbullying	Ensemble Machine Learning Ensemble classifiers outperform individual models (up to 89% accuracy)	Ensemble Machine Learning Ensemble classifiers outperform individual models (up to 89% accuracy)
7	Guo Xingyi, Hamedi Mohd Adnan	Potential cyberbullying detection in social media platforms based on a multi-	Deep Learning (BERT & attention mechanism) Deep learning outperforms SVM with high precision (0.841) and recall (0.881)	Deep Learning (BERT & attention mechanism) Deep learning outperforms SVM with high

		task learning framework		precision (0.841) and recall (0.881)
8	Bandeh Ali Talpur, Declan O'Sullivan	Cyberbullying severity detection: A machine learning approach	Machine Learning with feature engineering	Random Forest achieved high accuracy (AUC: 0.971)
9	Madhurima S; K Abhijith Ajith; Swathy VS; Boppuru Rudra Prathap	Abusive words detection on Reddit comments using Machine learning algorithms	Various machine learning techniques, like Random Forest, Extreme Gradient Boosting Classifier (XGB), Gradient Boosting Machine (GBM), Support Vector Machine (SVM), and Convolutional Neural Network were used	Random forest method gives the better performance
10	*Aya Mousa a, Ismail Shahin a, Ali Bou Nassif b, Ashraf Elnagar c*	Detection of Arabic offensive language in social media using machine learning models	To classify the abusive language, a cascaded model consisting of Bidirectional Encoder Representation of Transformers (BERT) models (AraBERT, ArabicBERT, XLMRoBERTa, GigaBERT, MBERT, and QARiB), deep learning models (1D-CNN, BiLSTM), and Radial Basis Function (RBF) is presented in this work.	The most elevated results are gotten from executing the cascaded model began by ArabicBERT followed by BiLSTM and RBF

3. Methodology

Adam Optimization

Adaptive Moment Estimation, or Adam, is a popular and effective optimization approach that is used extensively in deep neural network training [22]. It is a popular choice for many machine learning applications because of its capacity to handle problems including noisy training data, non-stationary targets, and sparse gradients [28]. Fundamentally, Adam combines components from momentum and RMS prop, two well-known optimization strategies [21]. Adam provides momentum-driven updates and adjustable learning rates, combining the best features of both approaches to produce better performance and faster convergence. Adam's capacity to dynamically modify his comprehension rate for any parameter throughout training is one of its unique characteristics. Conventional optimisation methods frequently employ a set learning rate, which can result in less-than-ideal results, particularly when dealing with non-stationary objectives or scenarios with changing gradients [18]. In order to solve this problem, Adam uses estimations of the initial and subsequent phases of the gradients to calculate each person's adaptive learning rate.

Adam keeps track of two exponential declining moving averages of previous gradients: the mean (first moment) and the uncentered variance (second moment) [5]. These moving averages provide information about the direction and magnitude of parameter updates by acting as estimations regarding the gradient's mean and variance, respectively. Using hyper parameters beta1 and beta2, the algorithm uses exponential decay to update these averages that move at each iteration regulating the rates of deterioration. Predictions of the first and second moments might be biased towards zero in the early phases of training, especially if the momentum terms are modest. Adam uses bias correction to overcome this problem by modifying the estimations to take the initial bias into consideration [19]. In order to maintain stability in the optimization process and guarantee precise updates to the model parameters, this correction is essential. Using variables obtained from the decaying rates beta1 and beta2, the estimations are scaled. Every iteration, Adam computes the changes for the model's parameters using momentum-driven updates and adaptive learning rates. It multiplies the preliminary moment estimate by the learning rate and divides it by the square root of the subsequent momentary estimate (with bias correction).

Through this approach, optimisation becomes more stable and effective by ensuring that parameters with big gradient receive smaller updates and variables with small gradient receive greater updates. Adam uses momentum, a method that dampens oscillations and helps expedite the descent of gradients in the appropriate direction. The hyper parameter beta1 governs the momentum term, which upholds an exponentially decreasing moving average of previous gradients. Adam's addition of momentum improves the optimisation process's robustness and rate of convergence, especially when there are noisy or sparse gradient [25]. Adam necessitates specifying a number of hyper parameters, such as the training rate, beta1, beta2, and epsilon, like the majority of optimisation algorithms. While beta1 and beta2 regulate the moving averages' decay rates, the learning rate sets the step size of variable updates. A little constant called epsilon is introduced to the denominator in order to guard against divide by zero and guarantee numerical stability. It is essential to adjust these hyper parameters in order to maximise Adam's performance and attain ideal convergence [33].

Adam has proven to be significantly more advantageous in reality than conventional optimisation methods, especially when it comes to developing deep neural networks [7]. With its momentum-driven updates and flexible learning rates, it can handle complex optimisation landscapes and large-scale datasets with better generalisation performance and faster convergence. However, in order to guarantee peak performance and avoid problems like divergence or overfitting, accurate hyper parameter tuning is crucial [24]. Furthermore, adaptations of Adam, including Adam and AMSGrad, have been suggested to tackle particular issues or improve performance even more in particular situations, underscoring the continuous research and development activities in machine learning optimisation methods [14].

Pseudo Code for Adam Optimizer

- **Step 1**: Set up the parameters.
- **Step 2**: Calculate gradients.
- **Step 3**: Refresh the skewed initial instant approximation.
- **Step 4**: Update the biased second moment estimate.
- **Step 5**: Calculate the bias-corrected first moment estimation.
- **Step 6**: Determine the bias-corrected estimate of the second moment.
- **Step 7**: Adjust the settings.

DNN Combined with Adam Optimizer

The Adam optimizer is a crucial component of deep learning training for Deep Neural Networks (DNNs), providing effective and flexible optimization for model parameter learning. Adam works well with DNN architectures to address issues like non-stationary targets and sparse gradients, as well as to speed up convergence and enhance generalization performance. An artificial neural network type known as Deep Neural Networks (DNNs) has multiple undetectable layers between the input layer and output layers of the network. Their exceptional efficacy lies in their ability to discern intricate patterns and correlations within data, rendering them appropriate for an extensive array of applications like as image identification, processing natural languages, and learning through reinforcement [7]. Adam uses estimations of the first through second moment of the gradients to dynamically modify the development rates for each model parameter [29].

Adam's ability to adapt enables him to effortlessly increase or decrease the learning rates according on the gradients' size, which improves generalization performance and speeds up convergence. Sparse gradients are a common occurrence for DNNs, particularly during the early phases of training or while working with high-dimensional data. By independently modifying the pace of learning for each parameter according to the gradients' scale, Adam's adaptive rate of learning technique helps lessen the effects of sparse gradients. The optimization environment can be non-stationary, or changing over time, in many real-world applications. Adam is resistant to non-stationary goals and facilitates effective DNN training in dynamic contexts thanks to its adaptive learning rates, which enable it to quickly adjust to modifications in the optimization landscape [31]. Adam uses momentum, which dampens oscillations and speeds up the descent of gradients in the appropriate direction. With the help of this momentum term, Adam is better able to negotiate challenging optimization landscapes by maintaining an exponentially decreasing moving average of previous gradients.

In order to correct for bias produced during the early training iterations, Adam adds bias correction on the projections of the first along with second moments. This bias correction guarantees more consistent training dynamics and helps to increase the moment estimations' accuracy. To obtain the best results when training DNNs using Adam, it is crucial to adjust the hyper parameters, including the rate of learning, beta1, beta2, and epsilon. Although Adam generally provides strong performance for a variety of applications and architectures, optimal performance requires experimentation and precise hyper parameter optimization[10]. For learning Deep Neural Networks (DNNs), the Adam optimizer is an incredibly powerful optimization technique that provides robustness against non-stationary objectives, adaptive learning rates, good handling of sparse gradients, and momentum integration[32]. Deep learning practitioners use Adam because it enables faster convergence, greater generalisation efficiency, and more consistent training dynamics when paired with DNN architectures.

Pseudo Code for DNN with Adam Optimizer

- **Step 1:** Set the Neural Network's Initial Parameters.
- **Step 2:** Define Forward Propagation.
- **Step 3:** Define Backward Propagation.
- **Step 4:** Setup Adam Optimizer.
- **Step 5:** Training Loop Testing.

- **Step 6:** Assessment, Fine-Tuning, and Optimization.

ANN Combined with Adam Optimizer

Artificial Neural Networks (ANNs) have brought about a revolution in deep learning by showcasing their exceptional ability in a range of tasks; including reinforcement learning, image recognition, and natural language processing. In order to minimize a loss function, the parameters of ANNs must be optimized. Stochastic gradient descent (SGD) and other optimization methods are frequently used for this purpose. Conventional optimization methods, such as SGD, might suffer from sluggish convergence or become trapped in local minima. They also frequently necessitate precise tweaking of hyper parameters, such as momentum and learning rates. In order to overcome these difficulties, scientists have developed adaptive optimization algorithms that combine the advantages of momentum with adaptive learning rates, for instance Adam (Adaptive Moment Estimation). Adam provides a number of benefits when combined with ANNs to train deep neural networks.

By dynamically modifying the learning rates for every parameter according to adaptive estimations of the first and second phases of the gradients, Adam optimization improves training. Because of his adaptability, Adam is able to manage deep learning challenges that frequently involve non-stationary targets and sparse gradients. Two exponential decreasing moving averages of previous gradients are maintained by the algorithm: the mean (first moment) and the uncentered variance (second moment). The parameters known as beta1 and beta2 regulate the exponential decay that updates these moving averages at each iteration. The gradients' mean is captured by the first moment estimate, while the gradients' scale is shown by the second moment estimate.

Adam can produce bigger changes for variables with small gradient and fewer modifications for variables with large gradient thanks to its adaptive learning rate method, which promotes faster convergence and better generalization performance. Adam has a number of benefits when used to train deep neural networks. First off, it eliminates the need for labor-intensive and difficult manual hyper parameter tuning, such as rate of learning and momentum. More stable and effective optimization results from Adam's adaptive learning rate method, which automatically modifies the rate at which one learns rates depending on the gradients' magnitude. Second, compared to conventional optimization techniques, Adam speeds up convergence, enabling deep neural networks to achieve their ideal performance sooner.

Large-scale neural networks trained on complicated datasets—where quick convergence is essential for real-world applications—benefit greatly from this. Adam's momentum term also aids in navigating across saddle points and escaping shallow local minima, which improves resilience and generalizatio performance. With its ability to provide adaptable learning rates, growth, and bias correction to improve convergence and generalization performance, Adam optimization is a potent technique for training deep neural networks. In the deep learning field, Adam is a well-liked option since it offers substantial advantages when combined with ANNs to tackle challenging machine learning problems. However, in order to fully utilize Adam's potential and attain ideal outcomes in practice, cautious hyper parameter tweaking and testing are necessary [17].

Pseudo Code for ANN with Adam Optimizer

- **Step 1:** Set the biases and weights at random initially.
- **Step 2:** Set up the Adam parameters, including epsilon, beta1, and beta2.
- **Step 3:** Indicate the quantity of training epochs.
- **Step 4:** Repeatedly go through mini-batches and shuffle the training set throughout each epoch.
- **Step 5:** Use back propagation to calculate gradients for every mini-batch.
- **Step 6:** Use the Adam optimizer to modify the setting parameters (weights and biases), which entails calculating biased and bias-corrected instant estimates.
- **Step 7:** The learning rate is modified as indicated throughout time.
- **Step 8:** The estimations of the initial and subsequent moments of gradients are used by the Adam optimizer to modify the rate of learning for each parameter.

RBFN Combined with Adam Optimizer

Combining the Adam optimizer with the Radial Basis Function Network (RBFN) algorithm creates a powerful framework for handling challenging pattern recognition, regression, however, and function approximation applications. Radial basis functions (RBFs) are at the heart of RBFNs' distinctive architecture, which consists of input, hidden, and output layers [32]. The input layer of an RBFN processes data, whereas the hidden layer calculates the radial basis function activations. The similarity between the input data and standard vectors, also known as centroids, which are frequently initialized randomly or obtained from the training data, is measured by these functions. The output layer creates the final output by combining the answers from the hidden layer. RBFNs can effectively capture nonlinear connections within data thanks to their architecture.

Adaptive optimization algorithms such as the Adam optimizer improve RBFN training by dynamically varying the rate of learning for each parameter. It adaptively updates parameters by maintaining the two averages of gradient (first and second moments), which leads to faster convergence and better resilience [23]. The Adam optimizer modifies the weights of interconnections across the hidden and output layers iteratively during training. The goal of this adjustment procedure is to reduce the error between the target values and the network's predictions. Through the utilization of adaptive learning stages, the Adam optimizer expedites convergence and adeptly navigates the intricate optimization terrain linked to RBFNs. There are various benefits when using the Adam optimizer in conjunction with RBFN. First of all, RBFNs are excellent at capturing intricate nonlinear correlations in data, which makes them useful for a variety of applications. Second, the adaptive learning rates of the Adam optimizer improve the training process's resilience and efficiency, allowing RBFNs to converge more quickly and perform better in generalization [24].

RBFNs are well-suited for situations where model visibility is crucial due to their simplicity and interpretability. There are still issues with using the Adam optimizer to optimize RBFNs. The ideal number of radial base functions and their centroids is still up for debate, and poor initialization can result in less-than-ideal results. Furthermore, careful hyper parameter optimization is needed for RBFNs using the Adam optimizer in order to guarantee convergence and avoid over fitting. In conclusion, the combination of the Adam optimizer with the Radial Basis Function Network algorithm provides a strong foundation for addressing a variety of machine learning challenges. In order to tackle real-world problems in a variety of fields, including engineering, finance, healthcare, and beyond, RBFNs with the Adam optimizer show promise due to their combined abilities in capturing complicated patterns and modifying learning rates.

Pseudo Code for RBFN with Adam Optimizer

- **Step 1:** Set the radial basis functions' centroids to their initial values.
- **Step 2:** Initializes the RBFN's weights and biases.
- **Step 3:** Identifies the activation function typically Gaussian radial basis functions that is applied to the hidden layer neurons.
- **Step 4:** Determines the Adam optimizer's hyper parameters, including the total amount of epochs, learning rate, and various other variables.
- **Step 5:** Using the Adam optimizer, run over the initial training data for several epochs, computing gradients and updating the network parameters through forward and backward passes.
- **Step 6:** Suggests that the RBFN is able to be used to make prediction on new data after training.

4. Experimental Results

The objective of this research work is to use text datasets (categorical data) to forecast heart disease. Numerous techniques are employed to identify problems in the healthcare industry, such as medical data sets, and to forecast patient illness risks and individual costs. Preprocessing the data and making predictions are the two primary processes in this research project [4]. During preprocessing, the duplicate record, data that is missing, and noise in the reliable information will be eliminated from the database [30]. In order to predict the attitudes and viewpoints in the kind of textual data, the dataset was used in the prediction process using Deep Neural Network, Artificial Neural Network, and Radial Basis Function Networkand Adam optimization [11.]

Data Set

The text-based dataset used in this study was downloaded as a.csv file from the Kaggle repository.

Preprocessing

A variety of actions are taken as part of preprocessing procedures for sentiment analysis in order to clean and prepare the data needed for sentiment classification jobs. These methods aim to portray the sentiment from based on text data in a unique way by using the best cleaning processes. During the preprocessing step, methods including tokenization, punctuation and stop word removal, part-of-speech labelling, stemming, and lemmatization are often used [15].

Results and Discussions

This section provides a full report on the outcomes of the cyberbullying algorithms, which are built in the Python computer language. The algorithms include DNN, ANN, RBFN, and DNN with Adam Optimizer, ANN with Adam Optimizer, and RBFN with Adam Optimizer.

All six of the algorithm effectiveness is evaluated using f-measure, accuracy, precision, and recall. The figure 2 shows the proportion of precision, recall and f-measure for all the six methods.

Figure 2 shows the results of the performance study of all six algorithms. The DNN algorithm achieves the highest precision (74.57%), recall (70.03%), and F-measure (76.28%) values. The precision value of 79.05%, recall value of 75.33%, and F-measure value of 78.67% are achieved using DNN with Adam optimizer. The ANN Algorithm is achieved with precision of 80.98%, recall of 78.21%, and F-measure of 81.82%.

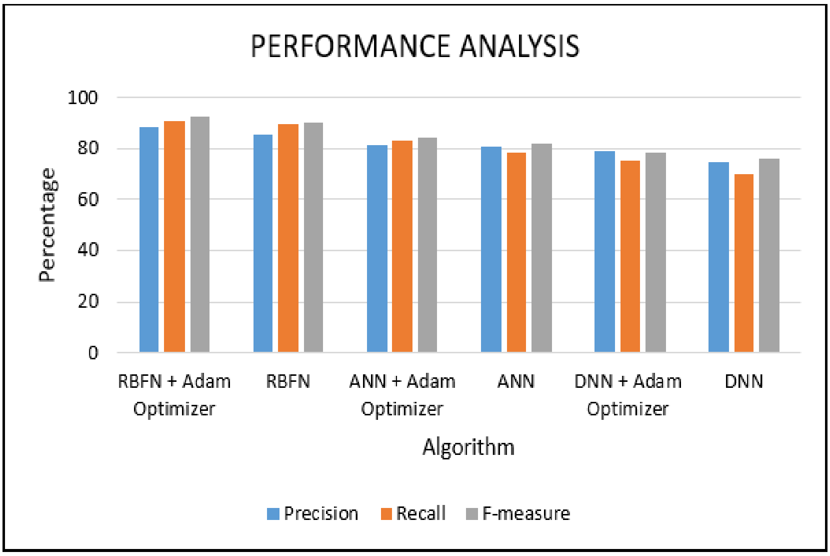

Figure 2. Performance analysis of algorithm

The accuracy, recall, and f-measure values of the ANN with Adam optimizer are 81.26%, 83.39%, and 84.16%, respectively. The precision, recall, and f-measure values that the RBFN algorithm obtains are 85.48%, 89.76%, and 90.25%. With Adam optimizer, the RBFN is achieved with a precision value of 88.63%, a recall value of 90.79%, and an F-measure value of 92.74%.

Table 2. Performance analysis of algorithm

Algorithms	**Precision**	**Recall**	**F-measure**
DNN	74.57	70.03	76.28
DNN + Adam Optimizer	79.05	75.33	78.67
ANN	80.98	78.21	81.82
ANN + Adam Optimizer	81.26	83.39	84.16
RBFN	85.48	89.76	90.25
RBFN + Adam Optimizer	88.63	90.79	92.74

The results demonstrate that, in comparison to the other five available approaches, the RBFN with Adam optimizer performs better.

The performance analysis is displayed in Table 2, and the accuracy of each algorithm is displayed in Table 3.

Table 3. Accuracy analysis of algorithm

Algorithms	Accuracy
DNN	65.39
DNN + Adam Optimizer	68.75
ANN	79.16
ANN + Adam Optimizer	87.50
RBFN	90.83
RBFN + Adam Optimizer	93.75

As demonstrated in Figure 3, the accuracy of the comparison measure is assessed using methods that are currently available and advised. The algorithm is shown on the x-axis, while the accuracy numbers are shown on the y-axis.

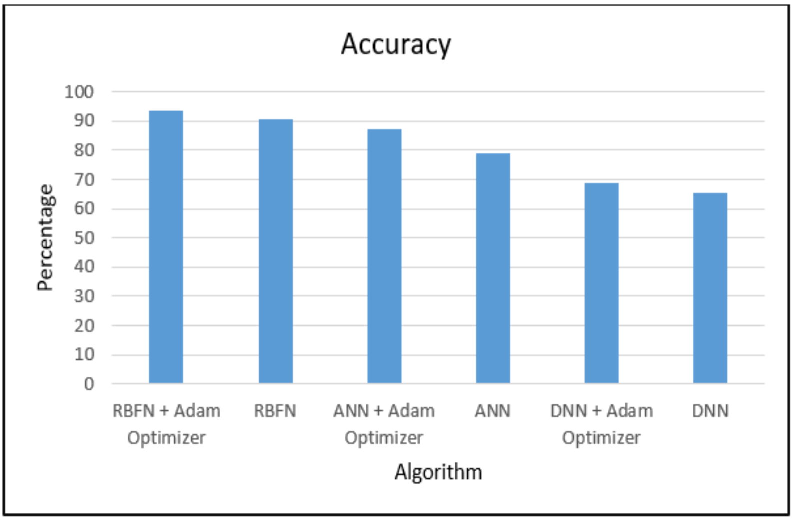

Figure 3. Accuracy of algorithms

For the given database, the suggested RBFN with Adam Optimizer approach is more precise than existing methods like DNN, ANN, RBFN, and DNN with Adam Optimizer and ANN with Adam Optimizer. The results show that by choosing the best features, the suggested RBFN with Adam Optimizer increases the accuracy of cyberbullying detection.

5. Conclusion

It was found that the most explored topic in the environment of social media was the identification of cyberbullying. Creating a system to identify cyberbullying is essential to protecting people from harassment and ensuring safety. In this work, the RBFN with Adam Optimizer approach is used to achieve this. In this research effort, the punctuation, urls, html tags, and emotions from the input tweet messages are eliminated first. Preprocessing is followed by sentiment feature extraction to increase classification accuracy. The study's test results demonstrate the effectiveness of the recommended strategy. The RBFN with Adam Optimizer outperforms existing algorithms in terms of classification performance when it comes to decreased better precision, recall, fmeasure, and accuracy.

References

1. Thomas, H. J., Scott, J. G., Coates, J. M., & Connor, J. P. (2019). Development and validation of the Bullying and Cyberbullying Scale for Adolescents: A multi-dimensional measurement model. *British Journal of Educational Psychology*, *89*(1), 75-94.
2. Srinivasareddy, S., Narayana, Y. V., & Krishna, D. (2021). Sector beam synthesis in linear antenna arrays using social group optimization algorithm. *National Journal of Antennas and Propagation (NJAP)*, *3*(2), 6-9.
3. Patchin, J. W., & Hinduja, S. (2015). Measuring cyberbullying: Implications for research. *Aggression and violent behavior*, *23*, 69-74.
4. Arora, G. (2024). Desing of VLSI Architecture for a flexible testbed of Artificial Neural Network for training and testing on FPGA. *Journal of VLSI Circuits and Systems, 6*(1), 30-35.
5. Prasad Babu, P., & Vasumathi, A. (2023). Role of Artificial Intelligence in Project Efficiency Mediating with Perceived Organizational Support in the Indian IT Sector. *Indian Journal of Information Sources and Services, 13*(2), 39–45.
6. Chen, Y., Zhou, Y., Zhu, S., & Xu, H. (2012). Detecting offensive language in social media to protect adolescent online safety. *In IEEE international conference on privacy, security, risk and trust and 2012 international confernece on social computing*, 71-80.
7. Hori, Y., Claycomb, W., & Yim, K. (2012). Guest editorial: frontiers in insider threats and data leakage prevention. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 3*(1/2), 1-3.
8. Romsaiyud, W., na Nakornphanom, K., Prasertsilp, P., Nurarak, P., & Konglerd, P. (2017). Automated cyberbullying detection using clustering appearance patterns. *In IEEE 9th International Conference on Knowledge and smart Technology (KST)*, 242-247.
9. Kiyomoto, S., & Martin, K.M. (2011). Model for a Common Notion of Privacy Leakage on Public Database. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, *2*(1), 50-62.
10. Tohma, K., & Kutlu, Y. (2020). Challenges Encountered in Turkish Natural Language Processing Studies. *Natural and Engineering Sciences, 5*(3), 204-211.
11. Mumtaj Begum, H. (2022). Scientometric Analysis of the Research Paper Output on Artificial Intelligence: A Study. *Indian Journal of Information Sources and Services, 12*(1), 52–58.
12. Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine learning*, *29*, 131-163.
13. Kao, Y. C., Liu, J. C., Wang, Y. H., Chu, Y. H., Tsai, S. C., & Lin, Y. B. (2019). Automatic Blocking Mechanism for Information Security with SDN. *Journal of Internet Services and Information Security, 9*(1), 60-73.
14. Bordoloi, M., & Biswas, S. K. (2018). Sentiment analysis of product using machine learning technique: a comparison among NB, SVM and MaxEnt. *International Journal of Pure and Applied Mathematics*, *118*(19), 71-83.
15. Brajendra, S. R., & Nilay, K. (2015). A Survey of Stemming Algorithms for Information Retrieval. *IOSR Journal of Computer Engineering (IOSR-JCE), 17*(3), 76-78.
16. Hui, H., An, X., Wang, H., Ju, W., Yang, H., Gao, H., & Lin, F. (2019). Survey on Blockchain for Internet of Things. *Journal of Internet Services and Information Security, 9*(2), 1-30.
17. Gümüş, A. E., Uyulan, Ç., & Guleken, Z. (2022). Detection of EEG Patterns for Induced Fear Emotion State via EMOTIV EEG Testbench. *Natural and Engineering Sciences, 7*(2), 148-168.
18. Khalifa, K., & Omar, N. (2014). A hybrid method using lexicon-based approach and Naive Bayes classifier for Arabic opinion question answering. *Journal of Computer Science*, *10*(10), 1961-1968.
19. Fattah, M. A. (2017). A novel statistical feature selection approach for text categorization. *Journal of information processing systems, 13*(5), 1397-1409.

20. Ali, W. N. H. W., Mohd, M., & Fauzi, F. (2018). Cyberbullying detection: an overview. *In IEEE Cyber Resilience Conference (CRC)*, 1-3.
21. Goljanin, A., Demirović, A., & Žiko, M. (2024). Models of Contepmporary Geodynamic Processes on the Rim of the Sarajevo Depression. *Archives for Technical Sciences, 1*(30), 15-24
22. Lozano-Blasco, R., Quilez-Robres, A., Rodriguez-Araya, R., & Casanovas-López, R. (2022). Addiction to new technologies and cyberbullying in the costa rican context. *Education Sciences*, *12*(12), 876. https://doi.org/10.3390/educsci12120876
23. B. Sri Nandhinia, and J.I. Sheeba, "Online Social Network Bullying Detection Using Intelligence Techniques", International Conference on Advanced Computing Technologies and Applications, Procedia Computer Science, 45, pp. 485 – 492, 2015.
24. Jelena, T., & Srđan, K. (2023). Smart Mining: Joint Model for Parametrization of Coal Excavation Process Based on Artificial Neural Networks. *Archives for Technical Sciences, 2*(29), 11-22.
25. Nahar, V., Li, X., & Pang, C. (2013). An effective approach for cyberbullying detection. *Communications in information science and management engineering*, *3*(5), 238.
26. Tian, L., Lai, C., & Moore, J. D. (2018). Polarity and intensity: the two aspects of sentiment analysis. *First Grand Challenge and Workshop on Human Multimodal Language (Challenge-HML), Melbourne, Australia*, 40–47.
27. Chen, L., Wang, W., Nagaraja, M., Wang, S., & Sheth, A. (2011). Beyond Positive/Negative Classification: Automatic Extraction of Sentiment Clues from Microblogs. *Kno. e. sis Center, Technical Report*.
28. Jiandani, D., Karkera, R., Manglani, M., Ahuja, M., & Tewari, A. (2018). Comparative Analysis of Different Machine Learning Algorithms to Detect Cyber-bullying on Facebook. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, *6*(IV), 2322-2328.
29. Puri, A., & Lakhwani, K. (2013). Enhanced Approach for Handwritten Text Recognition Using Neural Network. *International Journal of Communication and Computer Technologies (IJCCTS), 1*(2), 79-82.
30. Escortell, R., Aparisi, D., Martínez-Monteagudo, M. C., & Delgado, B. (2020). Personality traits and aggression as explanatory variables of cyberbullying in Spanish preadolescents. *International journal of environmental research and public health*, *17*(16), 5705. https://doi.org/10.3390/ijerph17165705
31. Cristina, B., Viviana, P., & Andrea, B. (2018). Developing Corpora for Sentiment Analysis: The Case of Irony and Senti–TUT. *Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, 4158-4162.
32. Holmes, G., Donkin, A., & Witten, I. H. (1994). Weka: A machine learning workbench. *In IEEE Proceedings of ANZIIS'94-Australian New Zealnd Intelligent Information Systems Conference*, 357-361.
33. Novitasari, N. F. (2021). Cyberbullying in Movie Cyberbully: an Analysis From the Psychological Perspective. *CELTIC, 8*(1), 44-64.