Automatic Detection of Cyberbullying using Multi-feature based Artificial Intelligence with Deep Decision Tree Classification

Natarajan Yuvaraj[1], Victor Chang[2], Balasubramanian Gobinathan[3], Arulprakash Pinagapani[4], Srihari Kannan[5], Gaurav Dhiman[6], Arsath Raja Rajan[7]

[1]Research and Development, ICT Academy, Chennai, India Email: yraj1989@gmail.com

[2]School of Computing, Engineering and Digital Technologies, Teesside University, Middlesbrough, UK

Email: ic.victor.chang@gmail.com

[2]Department of Computer Science and Engineering, Jaya Sakthi Engineering College, India

Email: jeevashanmugi@gmail.com

[4]Department of Computer Science and Engineering, Rathinam Technical Campus, India

Email: arulprakash247@gmail.com

[5]Department of Computer Science and Engineering, SNS College of Engineering, India

Email: harionto@gmail.com

[6]Department of Computer Science, Government Bikram College of Commerce, Patiala, India

Email: gdhiman0001@gmail.com

[7]Research and Development, ICT Academy, Chennai, India. Email: arshathraja.ru@gmail.com

Abstract:

Recent studies have shown that cyberbullying is a rising youth epidemic. In this paper, we develop a novel automated classification model that identifies the cyberbullying texts without fitting them into large dimensional space. On the other hand, a classifier cannot provide a limited convergent solution due to its overfitting problem. Considering such limitations, we developed a text classification engine that initially pre-processes the tweets, eliminates noise and other background information, extracts the selected features and classifies without data overfitting. The study develops a novel Deep Decision Tree classifier that utilizes the hidden layers of Deep Neural Network (DNN) as its tree node to process the input elements. The validation confirms the accuracy of classification using the novel Deep classifier with its improved text classification accuracy.

Keywords: Smart City; Cyberbullying detection; Deep Neural Network; Decision Trees; Artificial intelligence.

1 Introduction

With rapid urbanization and globalization, modern cities face challenges in maintaining development and qualified living for its citizens. The smart cities emerged such challenges with the integration of immersive technologies [1]. The assessment of online content from the social media platform in smart cities can be regarded as a vital asset that remains a critical challenge [2].

Cyberbullying (CB) is an electronic form of bullying[3] [4] that creates an intentional and aggression committed by a group or an individual against another group or an individual on social media platforms [7]. CB has hatred messages that are transmitted via social networking, emails, etc. through personal or public computers or through personal mobile phones. This has aroused as a serious threat among nations [1].

The suggested variances between the CB and traditional bullying reveal the inadequacy of CB findings from conventional bullying. The CB features are partially related and partially distinct from conventional bullying [5] [6]. CB has impacted the victims psychologically and physically with its increasing prevalence, where most vulnerability is reported among youths [7].

Hence, it is vital to detect the CB context and its applications to reduce the vulnerability in smart cities. However, from the cyber world view, the application involving CB involves difficulties associated with ignorance of aggressors and their identity, lack of direct communication, and relating consequences over others.

The failure to direct communication causes partial interpretation of the significance or the nature of the message, leading to confusion over the individual intentionality with exchange or interaction messages. Despite the problems, while identifying an individual's behavioral intent, the major factor that creates a transition from aggression to CB is the intention of harming oneself [8]. Nowadays, researchers are trying to develop these techniques for solving real-life problems.

In the current scenario, an automated behavior of the social network alerts the moderators to review the contents. However, existing frameworks lack an intelligent automated system to alter and detect the CB contents faster and accuracy than the traditional reporting system. Therefore, the moderator responded to the alert and required action is taken against the user content [9].

The existing research is carried out conventionally on available datasets or the surveyed data, where the perpetrators or the victims report the impressions [14]. The other issue is detecting contents only from the available literature for achieving the aims of automated detection to identify the CB accurately. The complexity in identifying the events is hence increased. The requirement of well-developed tools in integrating the features with an automated decision model [10] mounts.

Various research outputs on intelligent CB detection [11]utilized machine learning algorithms and also adopted common and psychological features. These intelligent systems are principally limited, with the comment of an individual leaving the context. An existing study has reported the utilization of the user context [9] in action that involves users' characteristics and history of

their comments to improve CB detection/classification performance. Nowadays, the researcher developed new approaches for automatically detecting complex real-life problems [22]. In this paper, a novel automated classification model is developed to identify the CB texts from the twitter engine inside a smart city.

The study contributes to the following in the field of CB detection:

(a) The author(s) developed an automated classification model that fits with large redundant datasets. The study utilized CB tweets as the input datasets with abundant datasets that have higher complexity in finding the CB texts.

(b) Considering the above limitations, we developed a text classification engine that initially pre-processes the tweets, eliminates noise and other background information, extracts the selected features and classifies without data overfitting.

(c) The author(s) developed a classifier that does not fall under convergence due to its overfitting problem. A novel Deep Decision Tree classifier is thus developed that utilizes hidden layers of Deep Neural Network (DNN) as its tree node to process the input elements. Such hybridization enables the classifier to produce improved classification accuracy than conventional classifiers.

The outline of the study is given below: Section 2 discusses the related works. Section 3 provides the proposed classification method. Section 4 evaluates the entire work and Section 5 concludes the work with possible directions of future scope.

2 Related Works

Nandhini, B. S., & Sheeba, J. I. [11] used the Fuzzy rule with a Genetic algorithm in classifying CB with parametric optimization. Potha et al. [12] employed a Support Vector Machine (SVM) classifier and identified various features like local, sentimental, contextual and gender-specific language features. The classifier used in the study is a non-linear one and it is combined with tf-idf measure. Kumar and Sachdeva [21] found that the direct/indirect features offer maximal higher impacts on a classifier. Al-garadi et al. [13] used various machine learning classifiers like naïve Bayes (NB) [18, 21], SVM [12, 21], random forest (RF) [18] and k-nearest neighbor (KNN) [25] classifiers. The study extracts different features, which are extracted from Twitter data. Balakrishnan et al. [18] developed an automated detection model with Big Five and Dark Triad models with user personality behavior as the only feature. The automated detection model uses NB, RF and J48 classifier to classify various classes of CB like a bully, spammer, aggressor and normal. Murnion et al. [14] used Artificial Intelligence to detect CB with sentiment text analytics from an automated data collection system from the chat data. Ho et al. [20] use a logistic regression model for classification with 90 extracted features. Balakrishnan et al. [16, 17] used RF classifiers with multiple decision trees (DT), where classification is finally determined based on the majority of votes. Sánchez-Medina et al. [19] used ensemble classification trees with

Dark Triad for identifying the personality trait. Lee et al. [15] used a three-layered neural network model as an unsupervised learning model for detection.

These methods use limited extracted features to train the classifier and the results are limited to the particular behavior of CB in online social media platforms. Further, the consideration of CB as a seed word is limited in the classification of CB texts. Since it is a distinctive vocabulary, the CB detection fails in classifying with limited features.

3 Proposed Method

In the present research, the Deep DT classifies the CB tweets and it is evaluated using the weight score calculation of optimal words chosen by the feature selection method of the collected tweets. This reduces well the cost of training data construction and further with the dependency between the phrases. The architecture of the proposed classification model is given in Fig. 1.

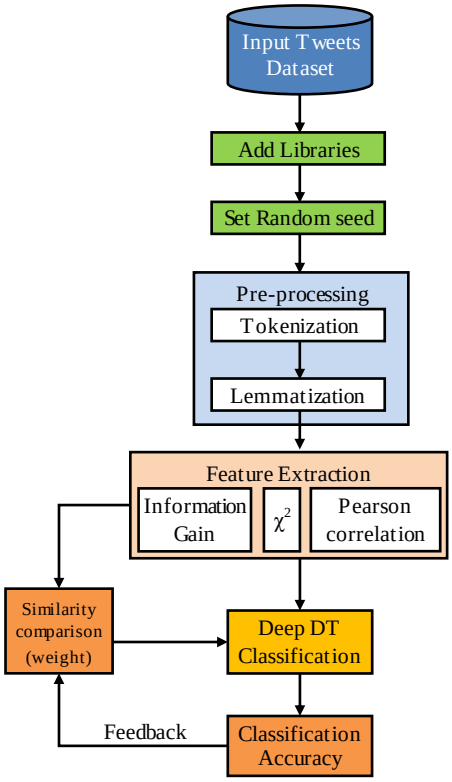

Fig 1. Overview of the proposed system

Consider an annotated dataset D={(x_i,~c_i)}, where $x_i$is the twitter CB datasets and without label~c_i. The datasets are divided into smaller subset $L \subset D$. The aim is to detect the CB instances from the twitter data that may vary from long to short paragraphs. Initially, the libraries are added to collect the dataset from the online portals. The random seeds are then set to produce a similar pattern of results and finally, the dataset is added with a corpus.

3.1 Pre-processing

The pre-processing converts the raw data into a machine-understandable format, since most of the collect tweets is inconsistent and it has more missing variables and missing root words. These tweets lack in analyzing certain trends or behavior and prone to typographical errors, which affects the classification process. In order to resolve these limitations, the study uses tokenization and lemmatization.

3.2 Feature Selection

The text features are selected from the large tweets datasets using three different methods, namely Information Gain [23], Chi-Square $\chi 2$ [24] and Pearson Correlation [25] and the details of the feature extractors are given below.

Information Gain: Decision tree algorithm is utilized to implement the feature extraction using information gain. The information gain is defined as the measure of entropy that is used widely in the machine learning domain. It acts as a statistical method that assigns the weights of features based on the correlation between the categories and the features. Consider a dataset $S(s_1,s_2,\ldots,s_n)$, which is regarded as the collection of varying instances say n s.t. $A(A_1, A_2,\ldots, A_p)$ is the attributes set for p, where $C(c_1, c_2,\ldots, c_m)$ is regarded as the collection of different label categories m. $p(c_i)$ represents the i^{th} class label proportion with c_i (i = 1, 2,…,m) in S. The dataset entropy is thus represented as:

$$H(C) = -\sum_{i=1}^{m} p(c_i)\log_2\left(p(c_i)\right)$$

The information gain on each feature is defined used for classification of input data, where $A_q(a_{q1}, a_{q2},\ldots,a_{qk})$ represents the q^{th}attribute (q=1,2,…,p). The conditional entropy for an attribute$A_q(a_{q1}, a_{q2},\ldots,a_{qk})$ is thus represented as:

$$H\left(C\middle|A_q\right) = -\sum_{j=1}^{k} p\left(a_{qj}\right)\sum_{i=1}^{m} p\left(c_i\middle|a_{qj}\right)\log_2\left(p\left(c_i\middle|a_{qj}\right)\right)$$

where

a_{qj}-A_q attribute value with k value,

$p(a_{qj})$- probability of categorical variable C.

$p(c_i|a_{qj})$- the conditional probability of C after the value of A_q is fixed.

Then, information gain is estimated as the difference between the value $H(C)$ and$H(C|A_q)$ and this offers the attribute value $A_q$as stated below:

$$IG\left(A_q\right) = H(C) - H\left(C\middle|A_q\right)$$

Usually, higher the information gain is, the feature is then considered vital for classification.

If the value of information gain is high, the feature is considered to be vital for the purpose of classification.

Chi-Square χ^2: The Chi-square statistics are used in feature extraction as an information theory function that helps in the extraction of elements, say t_k over a class c_i. These elements are considered to be distributed widely and differently in sets of negative and positive ci examples.

$$\chi^2\left(t(k),c(i)\right)=\frac{N\left(AD-CB\right)^2}{\left(A+C\right)\left(B+D\right)\left(A+B\right)\left(C+D\right)}$$

where:

N - total documents,

A - total documents in c_i containing t_k;

B – total documents containing the other than c_i;

C – total documents in c_i without t_k;

D – total documents without t_k other than c_i.

The next step is the assignment of scores for each c_i, as discussed in the above equation and the collective scores are summed into a single final score. The final score helps in the classification of attributes and the top score is selected.

Pearson Correlation: The Pearson correlation coefficient in the present study is used to estimate optimal features by calculating the degree of linear correlation between the extracted class and the original class.

$$sim_i=\frac{\sum_{j=1}^{N}\left(X_j-\bar{X}\right)\left(Y_{ij}-\bar{Y}\right)}{\left[\sum_{j=1}^{N}\left(X_j-\bar{X}\right)^2\right]\left[\sum_{j=1}^{N}\left(Y_{ij}-\bar{Y}\right)^2\right]}$$

where,

sim_i- similarity between i^{th} class and an original class of a dataset,

X_j and Y_{ij}–selected attribute data to be tested on i^{th} class,

Xand Y - the average value of selected attribute data and with the original class of a dataset and finally, the entire attribute data is normalized.

3.3 Non-Linear Deep DT Classification

Most of the non-linear classification methods use similarity or distance measurement to train the instances of CB data, including kNN and kernel SVM [12]. Hence, in this study, we use the DT classifier to predict the instances of high-dimensional CB

dataset based on feature selection rather than using similarity or distance measurement. However, the problem of overfitting rules the DT classifier in reducing its future prediction accuracy. In order to avoid overfitting, the approximation of class boundaries is required with optimal linear hyperplanes. It would enable the classifier to operate in an accurate way than with data overfit.

The common policy used in DT to avoid overfitting is to restrict the total number of instances to the minimal level prior to splitting and to limit the maximum tree depth in DT. The first restriction on the CB dataset eventually fails as the CB dataset is large and noisy and the second limitation on the CB dataset increases the computational complexity and limits the data classification. Therefore, it is essential to reduce the overfitting of data classes after classification using DT and on the other hand, the problems associated with the limitation of maximum tree depth and restriction of total instances compromise the classification of large and noisy text datasets.

In order to resolve the overfitting, especially in text classification, the study uses L1 regularization, which fits the noisy tweets with the overall class distribution. On the other hand, several noisy features exist in CB datasets that mislead the classification after extracting features in the sample space and increasing the errors. It further misleads the classifier with reduced accuracy. Hence, the partitioning of classes using DT into homogeneous class assignments in a sample space fails to conform to the dominant class, i.e., bullying. Hence, it is very necessary to lose focus on noisy tweets while choosing the boundary for DTs, which eases the non-linear classification. As in the previous section (feature extraction), we deploy three different methods to extract the features to reduce the problem of noisy data. The performance of each feature extraction model is evaluated on the Deep DT classifier, as discussed in section 4.

Further, limit the maximum tree depth in DT reduces the problem of overfitting, but it fails to process a large number of tweets at a time. During this instance, for the purpose of text classification, a DT classifier requires several key nodes. Finally, it is difficult to find the key tokens for classifying the CB texts. To resolve the problem of limiting the tree depth and finding the accurate key tokens, we in this study use DNNs supports the DT classifier to increase its maximum tree depth (Fig 2) such that it does not create data overfit. The increasing number of nodes in DT eventually creates a newer token to reduce the classification problem. The tree depth is increased by the substitution of decision tree nodes with hidden layers in DNN.

3.4 DNN

The multilayer perceptron is the most frequent architecture of a feedforward neural network. The input layer, an output layer and hidden layer all consist of at least three layers (Fig. 2). DNN is a multi-layered MLP. More precisely, using fewer neurons,

additional layers and therefore, connections enable the modeling of rare dependencies in the training data. Nevertheless, the DNN learning process can result in overfitting and declining performance.

In the theory of DNN, the universal approximation theorem says that a single hidden layer of MLP is enough to estimate with a certain accuracy all compactly supported continuous real functions. However, in many cases, DNN predictions are more exact, as research shows, compared to those obtained by ANN networks.

DNN (Fig 2) is trained with weights of input features and it is fitted with dropouts that eliminate the neurons during training to reduce data overfit. Hence, the error during the training of DNN gets reduced using the selected cross-entropy error function as below.

$$E = \sum_{i=1}^{n} y \log o_N + (1-y)\log(1-o_N)$$

DNN changes weights depending on the degree of an error function during the training process to minimize the error. There are several different algorithms for training purposes. Depending on a particular problem, the algorithms may vary in performance.

The size of the input twitter dataset D for a DNN classification model $P(y|x)$ is influenced by the selection of CB from D.The challenge of model building is to summarize the underlying distribution from the specific instance D of the samples. The problem with the memory of the data set is known as overfitting rather than identifying the dataset distribution.

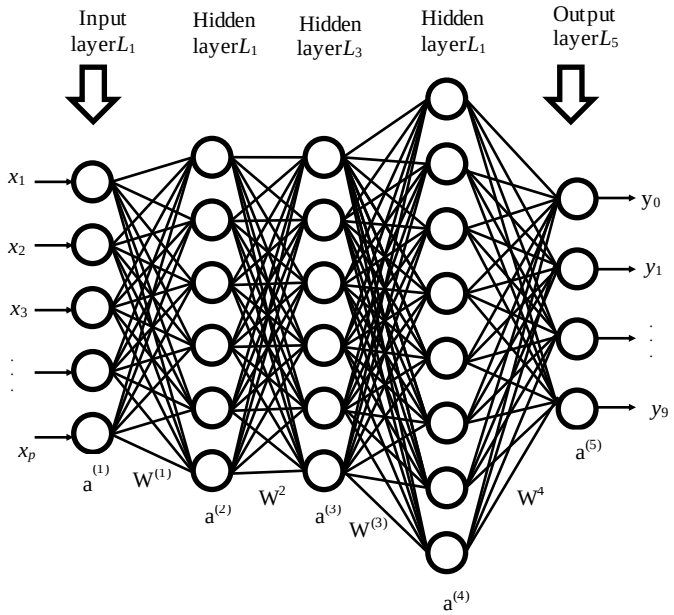

Fig 2.DNN architecture

An activation feature is considered as a real function that determines the value of the neuron returned. The present study uses Rectified Linear Units as the activation function.

4 Results

The experiments are conducted in this section on a large twitter dataset [26] to evaluate the performance of the Deep DL model. The evaluation is conducted further to test if the classifier falls in convergence and to find whether it is subjected to data overfitting. We used 30,384 tweets containing both CB and non-CB tweets without manual labeling or tagging. This unsupervised learning uses the help of the feature extraction models to label the datasets with repeated training. The feature extraction models labeling the tweets as CB are based on the attributes shown in Table 1, where most datasets are of text type and few are numeric. Table 1 shows both the common and uncommon traits of an individual while communicating with a large audience on the twitter platform. These uncommon traits help the feature extraction models to select the CB features and ensure that the classifier picks CB tweets from input datasets. Hence, the input tweet data are classified as CB and non-CB, where the former indicates the vulnerable behavior and the latter indicate genuine behavior. Out of 30,384, more than 1252 tweets are classified as CB datasets; however, the labeled data is not used to train the classifier. A feedback mechanism is provided with the present automated classification engine that either rewards or penalizes based on the similarity comparison of retrieved tweet datasets with the features available in Table 1.

The datasets consist of imbalanced classes that penalize the unsupervised classifier and produce inaccurate classification while identifying the CB instances. The Deep DT with imbalanced classes ignores the minor classes and performs well with major classes. The weight adjustment approach avoids oversampling of the minority class, i.e., abnormal class and undersampling the majority class, i.e., normal class.

The entire set of experiments is conducted with the topmost algorithms performed well in existing methods that include: ANN (Artificial Neural Networks), SVM (Support Vector Machines), RF (Random Forest) and LR (Linear Regression). These existing methods are compared with DEEP DT to find the classification accuracy. The present study utilized three feature selection methods, namely information gain, χ^2 and Pearson correlation techniques. 10-fold cross-validation is conducted and the proposed classified is tested individually with all these three feature selection methods.

Table 1. Selected Attributes to classify the Tweets

Attributes	Class	Format
Noun	CB/non-CB	Text
Pronoun	CB/non-CB	Text
Adjective	CB/non-CB	Text
local Features	The basic features extracted from a tweet	Text
Contextual Features	Professional, religious, family, legal and financial factors specific to CB	Text
Sentiment Features	Positive or Negative (foul words specific to CB) or direct or indirect CB	Text
Emotion features	Polite words, modal words, unknown words, number of insults and hateful blacklisted words, harming with a detailed description, power differential, any form of aggression, targeting a person, targeting two or more persons, intent, repetition, one-time CB, harm, perception, reasonable person/witness, and racist sentiments	Text
Gender-specific language	Male/Female	Text
User feature	Network information, user information, his/her activity information, tweet content, account creation time, verified account time	Text/ Numeric
Twitter basic features	number of followers, number of mentions, and number of following, favorite count, popularity, number of hashtags and status count	Numeric
Linguistic fea-	Other languages words, punctuation marks and ab-	Text

tures	breviated words rather than abusive sentence judgments.	

4.1 Experiment

The performance is estimated against various metrics that include: accuracy, F-measure, geometric mean (G-mean), percentage error, precision, sensitivity and specificity. The details of the metrics are given below:

Accuracy is defined as the total number of predictions required to ensure that the system works correctly. It is estimated as the ratio of the total number of correct predictions and the total predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

where:

TP is the true positive cases, where the model classifies the CB classes correctly.

TN is the true negative cases, where the model classifies the non-CB classes correctly.

FP is the false positive cases, where the model wrongly classifies the CB classes correctly.

FN is the false-negative cases, where the model wrongly classifies the non-CB classes correctly.

F-measure is the weighted harmonic mean of the recall and precision values, which ranges between zero and one. A higher value of F-measure refers to higher classification performance.

$$F - measure = \frac{2TP}{2TP + FP + FN} \quad (5)$$

G-mean is defined as the aggregation of sensitivity and specificity measure, which intends to maintain the trade-off between them, especially when the dataset is found to be imbalanced. This is measured as below:

$$G - mean = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}} \quad (5)$$

Mean Absolute Percentage Error (MAPE) is defined as the prediction accuracy measure that measures the total loss while predicting the actual classes. It is measured as the ratio of the difference between the actual (A_t) and predicted class (F_t), and the actual class. The entire value is multiplied by 100% and divided by the fitted points (n). The formula for the percentage error is defined as below:

$$MAPE = \frac{100}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right| \quad (5)$$

Sensitivity is defined as the ability of the deep learning model to identify the true positive rate correctly.

$$Sensitivity = \frac{TP}{TP + FN} \quad (3)$$

Specificity is defined as the ability of the deep learning model to identify the true negative rate correctly.

$$Specificity = \frac{TN}{TN + FP} \quad (4)$$

4.2 Analysis

The performance of the proposed classifier in comparison with existing machine learning classifiers, namely ANN, SVM, RF, LR and NB, is presented in this section. The results of predicting the CB are validated against 60%, 75% and 90% training data with various feature extraction methods: Information Gain, χ^2 and Pearson correlation techniques.

Figs. 3 - 5 provide the classification accuracy of Deep DT classifier while training it with feature selection methods on different percentages of input datasets: 60%, 75% and 90%. The results of Fig. 3 show that the Pearson correlation performs with optimal feature selection due to the highest classification accuracy of Deep DT classifier than information gain and χ^2. With increasing residuals of hidden layers, the number of tree tokens increases optimally with increased classification accuracy. The information gain feature selection tool reported the least performance on all training datasets (Figs. 3 - 5) than χ^2 and Pearson correlation. Therefore, the class of CB is determined accurately with Pearson correlation and Deep DT as the classifier. At the initial stage of residuals, we found that there is a linear increase in classification accuracy during training. This linearly suddenly drops and the value slowly reaches constant accuracy at the latter stages of residuals (i.e., as it is increasing from 400). We also found that the classification accuracy with 60% is deviating more than 75% and 90%, where the accuracy with 90% training data has smoother curves compared with other training data.

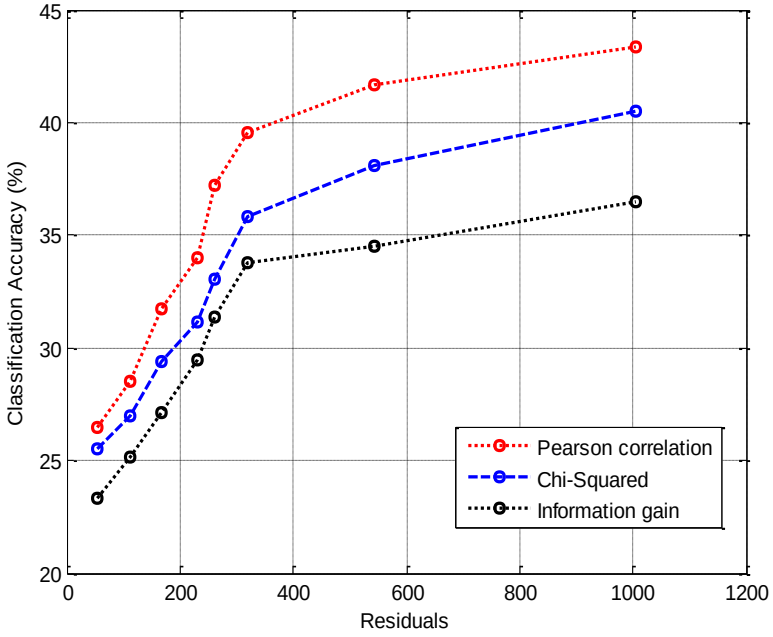

Fig. 3. Comparison of feature selection methods with 60% training data

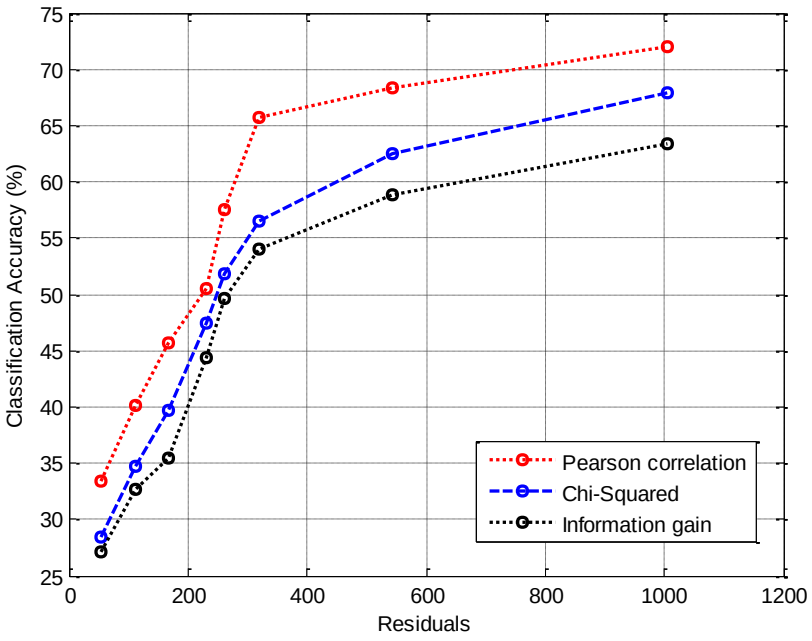

Fig. 4. Comparison of feature selection methods with 75% training data

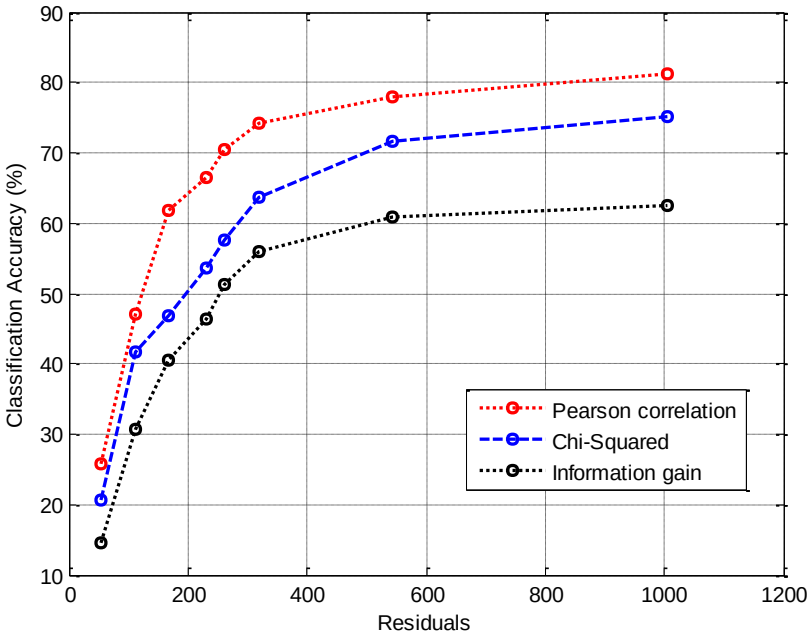

Fig. 5. Comparison of feature selection methods with 90% training data

Table 2. Results of accuracy with selected feature extraction methods on various classifiers

Feature selection methods	**Training data**	**NB**	**LR**	**RF**	**SVM**	**ANN**	**Deep DT**
Information Gain	60%	56.89	57.19	59.28	59.54	60.90	81.97
χ^2		57.48	60.09	62.45	63.82	67.06	83.91
Pearson correlation		60.29	67.07	70.09	75.31	79.15	86.19
Information Gain	75%	95.18	95.33	95.41	95.46	95.64	95.98
χ^2		97.12	97.14	97.15	97.24	97.26	97.65
Pearson correlation		97.67	98.39	98.42	98.50	98.51	98.98
Information Gain	90%	98.58	98.58	98.66	98.66	98.68	99.07
χ^2		98.65	98.65	98.73	98.73	98.75	99.14
Pearson correlation		98.94	98.96	98.98	98.98	98.99	99.33

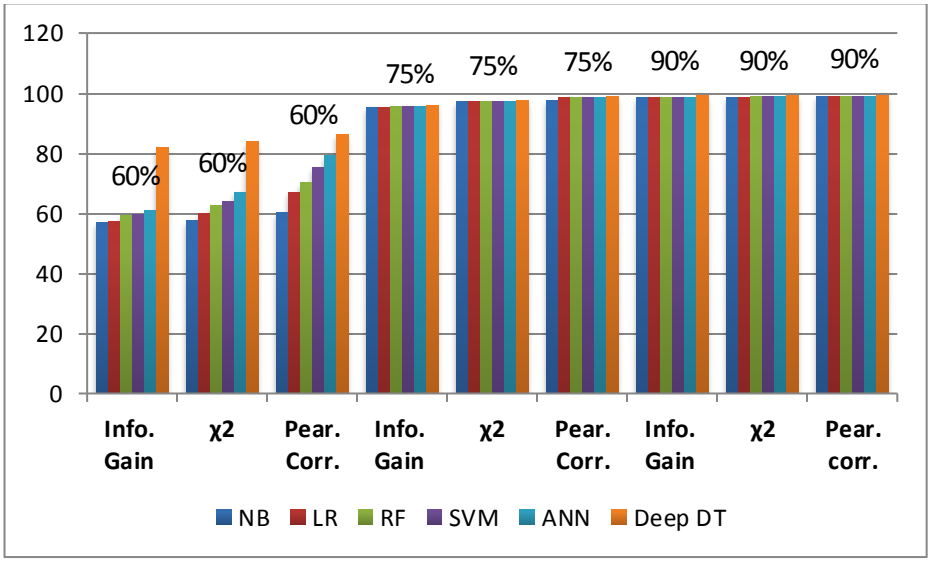

Fig. 6. Accuracy analysis using different classifiers

Table 3. Results of sensitivity with selected feature extraction methods on various classifiers

Feature selection methods	Training data	NB	LR	RF	SVM	ANN	Deep DT
Information Gain	60%	62.96	66.03	66.55	68.16	68.66	69.22
χ^2		65.67	66.47	68.53	68.89	70.71	71.79
Pearson correlation		66.91	67.64	73.09	74.86	75.06	82.19
Information Gain	75%	69.91	71.31	74.08	76.76	86.21	87.85
χ^2		77.45	72.37	74.38	85.06	86.40	90.12
Pearson correlation		78.74	80.13	80.22	86.76	87.42	93.81
Information Gain	90%	90.16	91.74	92.55	92.68	93.42	93.88
χ^2		91.74	91.81	92.62	92.75	93.49	97.80
Pearson correlation		91.81	96.79	97.89	98.47	98.75	99.10

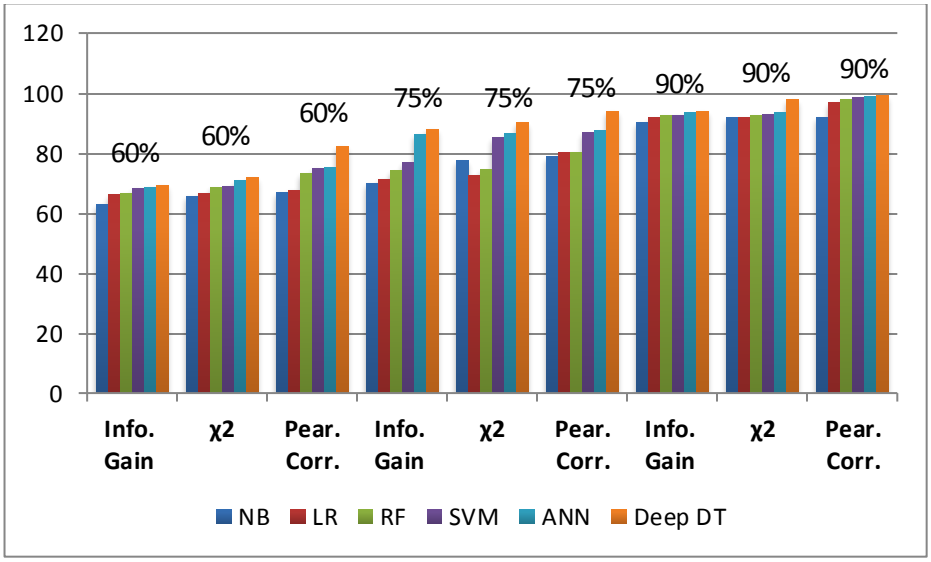

Fig. 7. Sensitivity analysis using different classifiers

Table 4. Results of specificity with selected feature extraction methods on various classifiers

Feature selection methods	Training data	NB	LR	RF	SVM	ANN	Deep DT
Information Gain	60%	71.62	73.50	76.57	79.12	80.49	81.60
χ^2		74.61	75.59	78.43	81.59	83.11	83.80
Pearson correlation		75.40	77.49	79.10	81.80	83.59	85.97
Information Gain	75%	95.93	95.99	96.03	96.68	96.72	97.10
χ^2		96.46	96.62	96.63	97.26	97.67	98.36
Pearson correlation		97.74	97.99	97.99	97.99	97.99	98.97
Information Gain	90%	97.97	98.53	98.73	98.77	98.83	99.20
χ^2		98.67	98.68	98.77	98.81	98.86	99.23
Pearson correlation		98.74	98.75	98.84	98.84	98.93	99.27

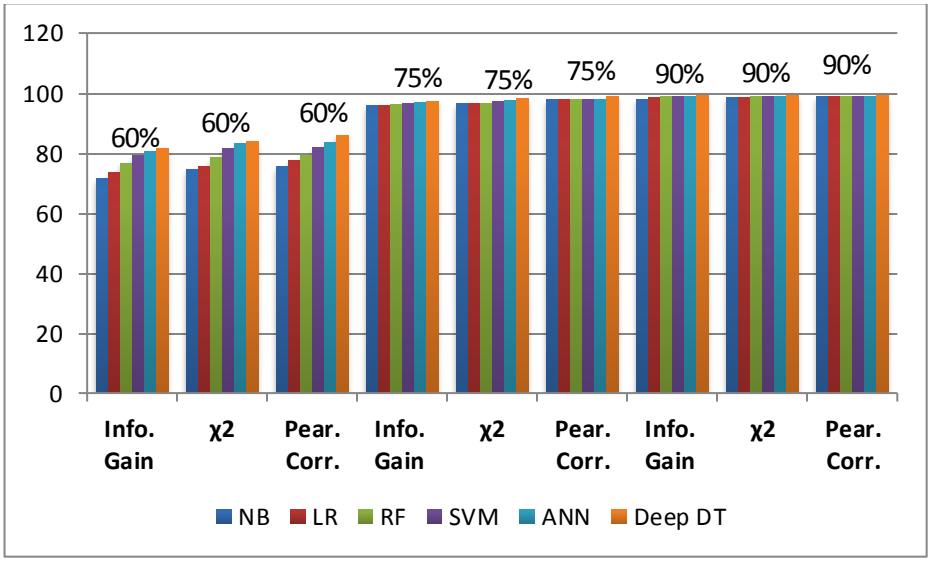

Fig. 8. Specificity analysis using different classifiers

Table 5. Results of F-measure with selected feature extraction methods on various classifiers

Feature selection methods	Training data	NB	LR	RF	SVM	ANN	Deep DT
Information Gain	60%	39.61	41.71	52.95	53.11	55.48	63.95
χ^2		53.59	61.26	61.60	62.06	63.59	80.85
Pearson correlation		59.47	68.01	68.90	70.02	75.06	80.89
Information Gain	75%	67.96	70.94	71.27	71.51	76.07	81.57
χ^2		70.90	71.16	71.33	74.15	77.38	81.89
Pearson correlation		78.60	78.73	79.25	80.33	81.01	85.16
Information Gain	90%	87.10	87.22	89.16	89.19	90.55	90.87
χ^2		87.16	87.28	89.22	89.25	90.61	90.93
Pearson correlation		90.41	91.84	91.99	92.50	92.71	93.67

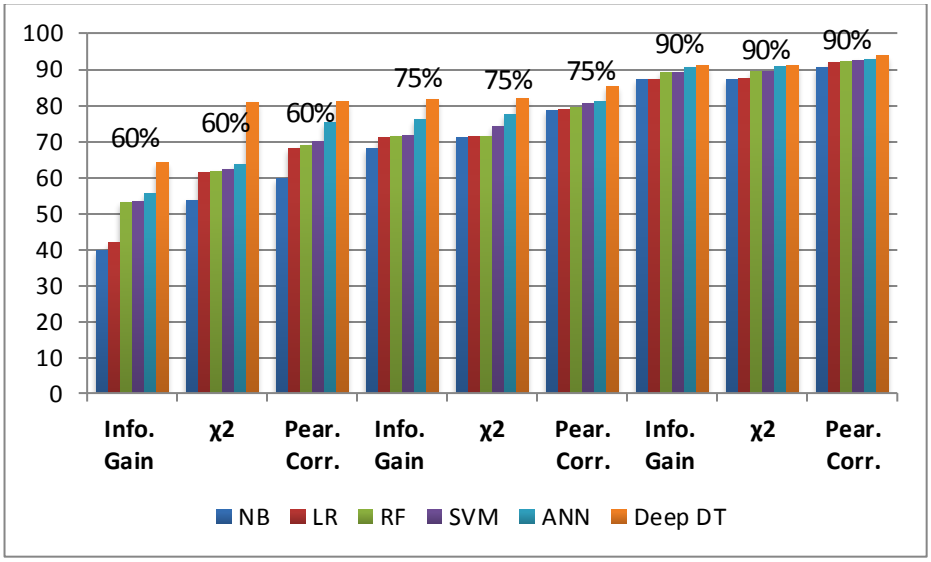

Fig. 9. F-measure analysis using different classifiers

Table 6. Results of G-mean with selected feature extraction methods on various classifiers

Feature selection methods	Training data	NB	LR	RF	SVM	ANN	Deep DT
Information Gain	60%	44.73	57.65	60.68	45.93	75.94	80.70
χ^2		71.21	71.44	73.16	75.22	77.31	82.95
Pearson correlation		73.76	73.99	75.49	75.53	77.68	83.48
Information Gain	75%	80.36	80.87	81.16	81.63	82.48	87.09
χ^2		80.65	80.89	81.40	82.16	82.74	87.50
Pearson correlation		83.10	83.95	85.57	87.13	92.17	94.01
Information Gain	90%	94.48	95.24	95.61	95.68	96.01	96.38
χ^2		95.24	95.31	95.68	95.75	96.08	96.45
Pearson correlation		95.31	97.88	98.43	98.73	98.87	99.21

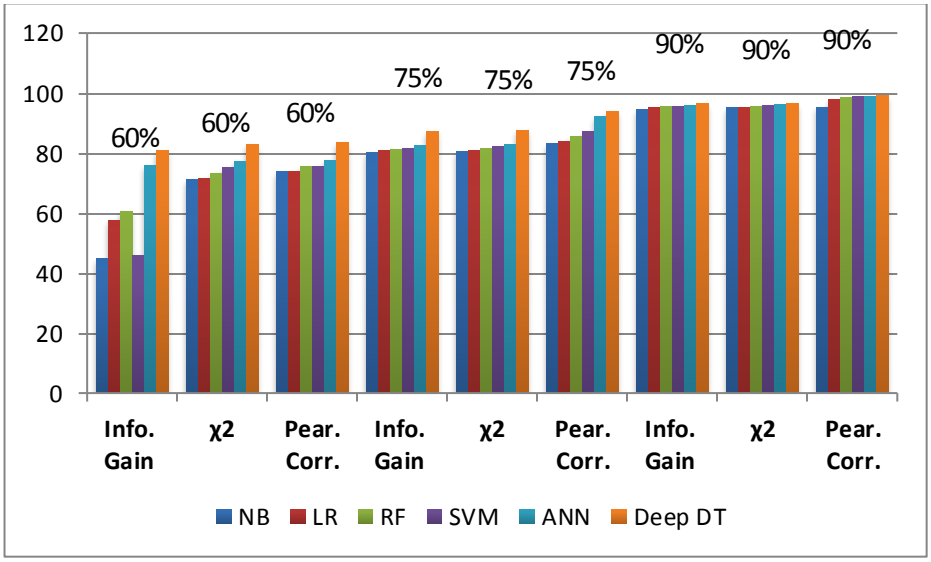

Fig. 10. G-mean analysis using different classifiers

Table 7. Results of MAPE with selected feature extraction methods on various classifiers

Feature selection methods	Training data	NB	LR	RF	SVM	ANN	Deep DT
Information Gain	60%	87.86	71.89	63.99	62.66	55.85	91.64
χ^2		72.07	71.84	63.94	62.61	55.30	54.80
Pearson correlation		72.02	65.69	58.97	40.85	55.26	54.76
Information Gain	75%	69.33	32.01	31.49	29.88	37.99	36.26
χ^2		32.37	30.40	29.51	29.16	29.38	29.16
Pearson correlation		31.13	28.24	25.21	22.63	27.33	26.60
Information Gain	90%	29.55	26.74	23.97	21.30	22.05	22.30
χ^2		28.13	26.61	21.48	12.98	11.83	17.40
Pearson correlation		20.60	17.93	17.84	10.50	11.65	10.54

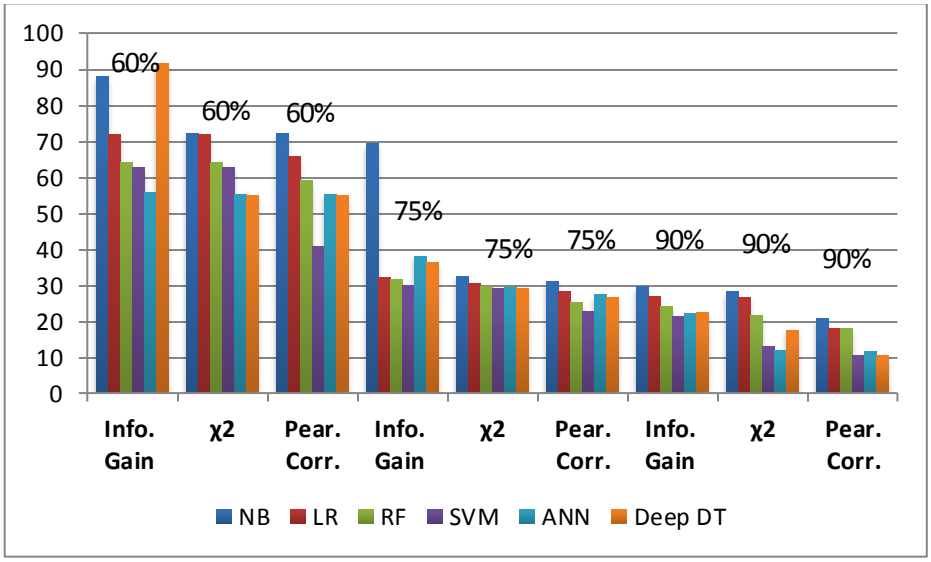

Fig. 11. MAPE analysis using different classifiers

Tables 2 - 7 and Figs. 6 - 11 show the results of classification accuracy, sensitivity, specificity and F-measure, G-mean and MAPE in predicting the CB over 60%, 75% and 90% of training data with the selected feature selection tools.

The results of the simulation show that the proposed method has higher classification accuracy than the existing classifiers. It is further inferred that the Pearson Correlation has an optimal selection of features that have boosted the classification accuracy with 90% training data than 75% or 60% datasets. The other metrics show optimal performance for Pearson Correlation than the other feature selection tools. Further, the MAPE of the Deep DT is lesser than the other methods.

5 Conclusions and Future Scope

In this paper, an automated classification engine was developed that picks the cyberbullying texts from the twitter datasets. The novel Deep Decision Tree classifier was considered so that it failed to get converged into a solution space, where the data had overfitted. On the other hand, the limited node formation associated with overfitting was reduced using a Deep Decision Tree classifier and this had improved the generation of node tokens. Such considerations could have enabled the classifier to perform an optimal way to select the cyberbullying texts from large datasets without degrading the classifier accuracy. The validation could confirm the increased classification accuracy of the Deep Decision Tree classifier (93.58%) than existing machine learning classifiers. For future work, this approach is further hybridized with new optimization and deep-learning approaches. Moreover, testing this approach to real-life high-dimensional datasets can also be seen as a future contribution.

Declaration of competing interest

None

Acknowledgment

This work is partly supported by VC Research (VCR 0000061) for Prof Chang.

References

[1] Singh, P., & Dhiman, G. (2018). A hybrid fuzzy time series forecasting model based on granular computing and bio-inspired optimization approaches. *Journal of computational science*, *27*, 370-385.

[2] Kumari, K., Singh, J. P., Dwivedi, Y. K., & Rana, N. P. (2019). Towards Cyberbullying-free social media in smart cities: a unified multi-modal approach. *Soft Computing*, 1-12.

[3] Olweus, D., & Limber, S. P. (2018). Some problems with cyberbullying research. *Current opinion in psychology*, *19*, 139-143.

[4] Vaillancourt, T., Faris, R., & Mishna, F. (2017). Cyberbullying in children and youth: implications for health and clinical practice. *The Canadian Journal of Psychiatry*, *62*(6), 368-373.

[5] Savage, M. W., & Tokunaga, R. S. (2017). Moving toward a theory: Testing an integrated model of cyberbullying perpetration, aggression, social skills, and Internet self-efficacy. *Computers in human behavior*, *71*, 353-361.

[6] Ansary, N. S. (2020). Cyberbullying: Concepts, theories, and correlates informing evidence-based best practices for prevention. *Aggression and violent behavior*, *50*, 101343.

[7] Dhiman, G., & Kaur, A. (2017). Spotted hyena optimizer for solving engineering design problems. In *2017 international conference on machine learning and data science (MLDS)* (pp. 114-119). IEEE.

[8] Cuadrado-Gordillo, I., & Fernández-Antelo, I. (2016). Adolescents' perception of the characterizing dimensions of cyberbullying: Differentiation between bullies' and victims' perceptions. *Computers in Human Behavior*, *55*, 653-663.

[9] Dadvar, M., Trieschnigg, D., Ordelman, R., & de Jong, F. (2013, March). Improving cyberbullying detection with user context. In *European Conference on Information Retrieval* (pp. 693-696). Springer, Berlin, Heidelberg.

[10] Dhiman, G., & Kaur, A. (2018). Optimizing the design of airfoil and optical buffer problems using spotted hyena optimizer. *Designs*, *2*(3), 28.

[11] Nandhini, B. S., &Sheeba, J. I. (2015). Online social network bullying detection using intelligence techniques. Procedia Computer Science, 45, 485-492.

[12] Potha, N., Maragoudakis, M., &Lyras, D. (2016). A biology-inspired, data mining framework for extracting patterns in sexual cyberbullying data. Knowledge-Based Systems, 96, 134-155.

[13] Al-garadi, M. A., Varathan, K. D., &Ravana, S. D. (2016). Cybercrime detection in online communications: The experimental case of cyberbullying detection in the Twitter network. Computers in Human Behavior, 63, 433-443.

[14] Murnion, S., Buchanan, W. J., Smales, A., & Russell, G. (2018). Machine learning and semantic analysis of in-game chat for cyberbullying. Computers & Security, 76, 197-213.

[15] Lee, H. S., Lee, H. R., Park, J. U., & Han, Y. S. (2018). An abusive text detection system based on enhanced abusive and non-abusive word lists. Decision Support Systems, 113, 22-31.

[16] Peter, I. K., &Petermann, F. (2018). Cyberbullying: A concept analysis of defining attributes and additional influencing factors. Computers in human behavior, 86, 350-366.

[17] Balakrishnan, V., Khan, S., Fernandez, T., &Arabnia, H. R. (2019). Cyberbullying detection on twitter using Big Five and Dark Triad features. Personality and individual differences, 141, 252-257.

[18] Balakrishnan, V., Khan, S., &Arabnia, H. R. (2020). Improving Cyberbullying Detection using Twitter Users' Psychological Features and Machine Learning. Computers & Security, 101710.

[19]Sánchez-Medina, A. J., Galván-Sánchez, I., &Fernández-Monroy, M. (2020). Applying artificial intelligence to explore sexual cyberbullying behaviour. Heliyon, 6(1), 1-9.

[20] Ho, S. M., Kao, D., Chiu-Huang, M. J., Li, W., & Lai, C. J. (2020). Detecting Cyberbullying "Hotspots" on Twitter: A Predictive Analytics Approach. Forensic Science International: Digital Investigation, 32, 300906.

[21] Kumar, A., &Sachdeva, N. (2019). Cyberbullying detection on social multimedia using soft computing techniques: a meta-analysis. Multimedia Tools and Applications, 78(17), 23973-24010.

[22] Dhiman, G., & Kaur, A. (2019). STOA: a bio-inspired based optimization algorithm for industrial engineering problems. *Engineering Applications of Artificial Intelligence*, *82*, 148-174.

[23] Zhang, G., Hou, J., Wang, J., Yan, C., & Luo, J. (2020). Feature Selection for Microarray Data Classification Using Hybrid Information Gain and a Modified Binary Krill Herd Algorithm. *Interdisciplinary Sciences, Computational Life Sciences*. https://doi.org/10.1007/s12539-020-00372-w

[24] Bahassine, S., Madani, A., Al-Sarem, M., &Kissi, M. (2020). Feature selection using an improved Chi-square for Arabic text classification. *Journal of King Saud University-Computer and Information Sciences*, *32*(2), 225-231.

[25] Feng, W., Zhu, Q., Zhuang, J., & Yu, S. (2019). An expert recommendation algorithm based on Pearson correlation coefficient and FP-growth. *Cluster Computing*, *22*(3), 7401-7412.

[26] Baral, C., Fuentes, O., &Kreinovich, V. (2018). Why deep neural networks: a possible theoretical explanation. In *Constraint Programming and Decision Making: Theory and Applications* (pp. 1-5). Springer, Cham.

N. YUVARAJ, received the BE degree in Computer Science Engineering from Coimbatore Institute of Engineering and Technology, and the ME degree in Computer Science engineering from Sri Shakthi Institute of Engineering and Technology, Anna University, Tamil Nadu, India, in 2010 and 2012 respectively. He is currently working as a Deputy Manager, Research & Development, ICT Academy, Tamil Nadu, India.

Victor Chang received PhD in Computer Science from University of Southampton, UK. He is currently a Professor in Data Science and Information Systems at Teesside University, Middlesbrough, UK. He is the Conference Chair of 4 international conferences, Associate Editor of IEEE TII and Editor of Information Fusion. He won numerous awards and funding, and is an active and influential researcher.

B. Gopinathan received PhD in Information and communication Engineering from Anna University. He is working as a Principal in Jaya Sakthi Engineering College ,Thiruninravur from January 2020 to till date. His research interests includes Medical Image Processing, Artificial Intelligence Techniques and Fuzzy Logic. He has authored more than 40 papers in international journals and conferences in his research areas.

P. Arulprakash completed his Ph.D in Information and Communication Engineering from Anna University, Chennai during 2018. He is having 11 years of teaching experience in various Engineering colleges and currently working as Associate Professor and Head of the Department in the Department of Computer Science and Engineering in Rathinam Technical Campus, Eachanari, Coimbatore.

K. Srihari received the ME and Ph.D. degree from Anna University, Chennai. He is currently working as an Associate Professor in the Department of Computer Science and Engineering, SNS College of Engineering, affiliated to Anna University-Chennai, Tamilnadu, India. Dr. K. Srihari published over 60 papers in international journals.

Gaurav Dhiman received his Ph.D. in Computer Engineering from Thapar Institute of Engineering & Technology, Patiala. He is currently an Assistant Professor in Government Bikram College of Commerce, Patiala. He has published more than 50 peer-reviewed *SCI-SCIE* journals articles and 5 international books. He serves as the lead guest editor of more than ten special issues in various peer-reviewed journals.

R. ARSHATH RAJA, received his BEng degree in Electronics and Communication Engineering from Lord Venkateshwara Engineering College, Anna University in 2012. He completed his Masters of Engineering in Communication Systems from Sree Sastha Institute of Engineering and Technology in 2014. He is currently pursuing his Ph.D in wireless communication in BS. Abdur Rahman Crescent Institute of Science and Technology.