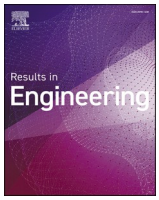
Results in Engineering 25 (2025) 103993

Contents lists available at ScienceDirect

Results in Engineering

journal homepage: www.sciencedirect.com/journal/results-in-engineering



Quantum behaved binary gravitational search algorithm with random forest for twitter spammer detection

Kanta Prasad Sharma [a], Gendal Lal [b], Madhu Shukla [c], Anupam Yadav [d,k], Jayaprakash B [e], Bhanu Juneja [f], Jayant Jagtap [g], Amrita Singh [h], A. Bhowmik [i], A. Johnson Santhosh [j,*]

[a] Department of CSE, Amity School of Engineering & Technology, Amity University, Greater Noida Campus, Greater Noida, Uttar Pradesh, India
[b] Department of Computer Science and Engineering, Vivekanand Global University, Jaipur, India
[c] Department of Computer Engineering, Faculty of Engineering & Technology, Marwadi University Research Center, Marwadi University, Rajkot, Gujarat 360003, India
[d] Department of Computer Science and Engineering, Graphic Era Hill University, Dehradun, India
[e] Department of Computer Science & IT, School of Sciences, JAIN (Deemed to be University), Bangalore, Karnataka, India
[f] Centre for Research Impact & Outcome, Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura, Punjab 140401, India
[g] NIMS School of Computing Science and Artificial Intelligence, NIMS University Rajasthan, Jaipur, India
[h] Department of Computer Science and Engineering, Chandigarh Engineering College, Chandigarh Group of Colleges-Jhanjeri, Mohali, Punjab 140307, India
[i] Chitkara Centre for Research and Development, Chitkara University, Baddi, Himachal Pradesh, 174103, India
[j] Faculty of Mechanical Engineering, Jimma Institute of Technology, Jimma University, Jimma, Ethiopia
[k] Graphic Era Deemed to be University, Dehradun, Uttarakhand-248002, India

ARTICLE INFO

Keywords:
Twitter
Twitter spammer detection
Gravitational search algorithm
Quantum computing
Binary gravitational search algorithm
Random forest
Machine learning
Metaheuristic

ABSTRACT

The emergence of social media platforms like Twitter has significantly changed the landscape of communication by increasing accessibility for widely disseminating official announcements, professional interactions, and important news in real-time. Despite these advantages, the prevalence of spammers and their spamming activities is increasing regularly. To mitigate the growing number of spammers, it is essential to develop an efficient and robust method for Twitter spammer detection. This research presents a novel QBGSRF method by combining the quantum-behaved binary gravitational search algorithm (QBGSA) with random forest (RF) for timely detection of Twitter spammers. The QBGSA algorithm adds the characteristics of quantum computing (QC) and binary gravitational search algorithm (BGSA), which enables the quantum agents to quickly determine solutions using the superposition attributes of QC and the position update via bit-flipping based on velocity probabilities of the BGSA algorithm. In the proposed QBGSRF method, the quantum agents utilize the aforementioned attributes and the principles of the RF algorithm to construct the decision trees for effectively detecting Twitter spammers. The proposed method is assessed for the datasets of 1KS-10KN and Social Honeypot. In order to access the efficacy of the proposed method, the results are also evaluated using the BGSRF method (a combination of BGSA and RF algorithm) and RF algorithm. The experimental evaluations indicate that the proposed method outperforms the aforementioned and state-of-the-art methods.

1. Introduction

In the social world, communication is an important factor to share the concepts and fostering personal development [1]. The ease of internet facilities has increased communication access for humans by means of social communication platforms [2,3]. The global count of social media consumers is steadily rising, from 2.73 billion users in the year 2017 to 5.17 billion users in the year 2024 [4]. Twitter is one of the popular social media platforms to deliver information in the form of text, known as tweets [5,6]. Moreover, Statista's recent reports indicate that the Twitter network has 611 million monthly active users [7]. Among other social networking platforms like Facebook, Telegram, WhatsApp, etc., Twitter can be considered a secure and authentic source to deliver any official information to a wide audience [8]. The distribution of Twitter users among the age groups and countries is different, as depicted in Fig. 1(a, b). The distribution data among the users of different age groups indicate that there is major usage of Twitter by the users of 25–34 years of age [9], as shown in Fig. 1(a). Regarding

* Corresponding author.
E-mail address: johnson.antony@ju.edu.et (A.J. Santhosh).

https://doi.org/10.1016/j.rineng.2025.103993
Received 6 September 2024; Received in revised form 9 December 2024; Accepted 8 January 2025
Available online 9 January 2025

countries, there is the major usage of Twitter that can be noticed in the United States, followed by Japan, India, Indonesia, and United Kingdom [10]. The top five countries with the highest count of Twitter users are illustrated in Fig. 1(b). Additionally, the male users constitute a larger proportion of Twitter users compared to female users, with 60.90 % being male and 39.10 % female [11].

Twitter is primarily utilized by journalists, government entities, and people to disseminate up-to-the-minute news and official announcements [12]. It also engages people in debate, shares public opinion, and makes them aware of current issues in society. Business organizations use Twitter to engage their clientele and advertise their services and products. The widespread adoption of Twitter has also attracted spammers who post unsolicited tweets with trending hashtags to lure legitimate users into unauthorized activities. These spammers try to direct users to malicious websites for their personal gain.

Twitter users can report suspicious accounts that send the same content to multiple users and post duplicate content as spammers. There are also terms and conditions on Twitter, and neglecting those terms results in blacklisting the accounts [13]. Despite the mechanisms of Twitter for identifying and blacklisting spammers, the advanced knowledge and dynamic strategies adapted by spammers enables them to bypass the measures of Twitter. The current spam detection mainly focused on the static machine learning algorithms that fails to adapt the evolving techniques of spammers and complexities of high-dimensional data. Moreover, the increasing volume of tweets also demand the methods that can accurately detect spammers are computationally efficient for handling large datasets.

Therefore, it is essential to develop novel methods to effectively and timely detect Twitter spammers. The present work addresses the issue by presenting a novel QBGSRF method to effectively detect Twitter spammers. This approach leverages the superposition attributes of QC to explore diverse candidate solutions, the global optimization abilities of the BGSA, and the robust classification characteristic of RF algorithm. The combination of these strengths in the proposed method effectively handles the dynamic and evolving behaviour of spammers while ensuring the scalability to large datasets.

In the proposed QBGSRF method, the GSA algorithm is a heuristic optimization method that adheres to the rules of mass interactions and gravity laws [14]. While GSA is effective for optimizing solutions in continuous problems, adapting GSA to discrete problems and integrating quantum computing concepts necessitates its transformation into binary GSA (BGSA) [15]. BGSA is an improved variant of GSA and utilizes binary strings to optimize solutions. It updates agent positions on the basis of bit change probabilities, which are determined by the velocity of mass agents. The integration of QC with BGSA and RF algorithms is used to effectively handle the dynamic nature of spammers.

The paper structures the subsequent section as follows: Section 2 provides related work on Twitter spammer detection using different machine learning and data mining techniques. Section 3 describes the datasets used for experiments, along with data preprocessing and feature selection steps for Twitter spammer detection. Section 4 discusses the proposed QBGSRF method along with an overview of the preliminary methods: QC, BGSA, and RF algorithms. Section 5 provides the results and discussion for evaluating the conducted experiments. Section 6 concludes the work by summarizing the key findings and proposing directions for future research work.

2. Related work

The section describes work related to Twitter spam and spammer detection. Spam has become widespread across social media networks due to the ease of access to internet services and the increased adaptability of humans to online networking platforms. Since Twitter is generally used for posting official and ethical information to a broad audience, spammers focus more on Twitter to easily disrupt legitimate users because Twitter users often believe the information in tweets to be authentic. The concept of Twitter spammer detection has been extensively addressed in recent years by different researchers, which is discussed here.

Tajalizadeh and Boostani [16] utilized the density-based clustering

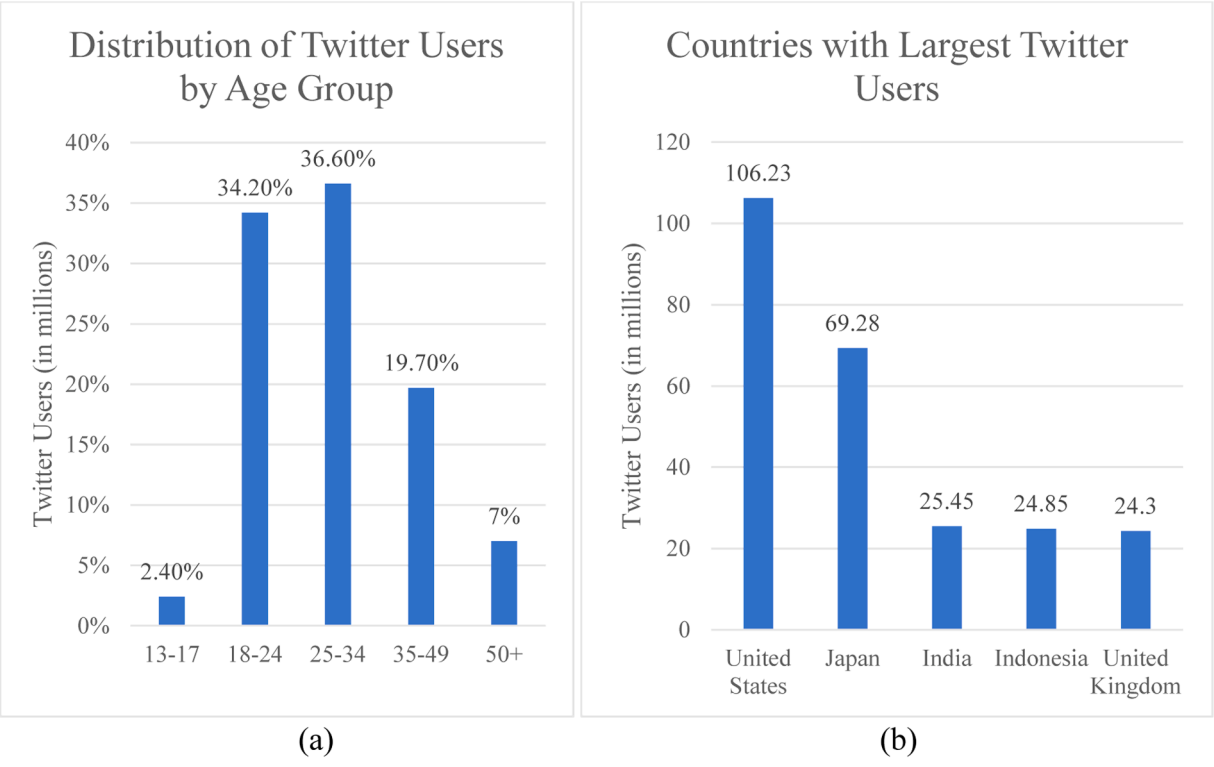


Fig. 1. (a) Distribution of twitter users worldwide by age group [9], (b) leading countries based on twitter users [10].

platform named Den-Stream along with an Incremental Naive Bayes (INB) method to detect spam on the Twitter platform. The INB-DenStream method was proposed for the replacement of Euclidean distance and improvement in accuracy. The method was effective compared to DenStream, but could be improved by incorporating real-time distance learning during the online phase. Wang et al. [17] utilized multiscale drift detection test (MDDT) on K-L convergence to detect spam on Twitter. The MDDT method was used for detecting drift in distribution. The experimental evaluations indicated the occurrence of consistent change patterns among features during drift, and the MDDT improved the evaluation accuracy. The authors suggested to incorporate the neural networks and imbalanced classification methods for the improvement of concept drift detection in the future. Lingam et al. [18] presented a deep Q-learning method that utilizes the particle swarm optimization (PSO) to detect spam influential users and spam bots. The authors integrated the PSO algorithm with the Q-value function for improving the convergence rate and also determined the influential participants and communities to mitigate the impact of spam content. The experiments were performed on offline datasets. Conducting experiments on the online datasets with ground truth evaluation was described as a future challenge. Güngör et al. [19] utilized different machine learning methods, including logistic regression, J48, and Naive Bayes for detecting Twitter spam. The experiments were performed on a manually extracted dataset of 758 tweets, with effective performance of the J48 classifier compared to other algorithms. Although the performance results were effective, the experiments were performed only for a limited dataset.

Further, Abkenar et al. [20] introduced a new method on the basis of RF and DE (Differential Evolution) algorithms for Twitter spam detection. This method utilized the Synthetic Minority Over-sampling TEchnique (SMOTE) to tackle the issue of imbalanced data, and DE for balancing the hyperparameters of the RF algorithm. The improved method increased the performance accuracy to detect Twitter spam compared to using the RF algorithm alone. The authors suggested extending the work to determine spammers by drawing a social graph based on relationships among individuals and available Twitter data. Ahmad et al. [21] discussed the incorporation of a Support Vector Machine (SVM) along with relevant features for detecting spam on the Twitter platform. The SVM classifier utilized polynomial and Gaussian functions for the learning process. An analysis of the SVM classifier compared to other machine learning algorithms indicated the effective performance of the SVM classifier for Twitter spam detection. The authors suggested to expand the research to include ontological and semantic features. Santoshi et al. [22] considered the Naive Bayes algorithm to classify Twitter spam and non-spam data. The classifier effectively classified Twitter spam compared to other machine and deep learning techniques. Although the authors indicated the effective performance of the classifier, they did not describe the details of the dataset or potential challenges of the approach used. Vives et al. [23] hybridized the heuristic algorithm GSA with the machine learning algorithm DT to detect Twitter spammers. The GSA constructed DTs using gravitational forces as information transferring agents. The authors conducted comparisons with limited machine learning algorithms and did not compare with other hybrid methods. Concone et al. [24] proposed the multi-stage spam account detection algorithm (SpADe) for detecting spam on Twitter. The authors initially extracted spammers using computationally less expensive features and then progressed towards the extraction of complex information to classify spam accounts, which appeared more intricate. The performance effectiveness can be determined for the comparison with single stage approaches. The unavailability of the reject option in the SpADe algorithm may lead to misclassification errors, for which authors mentioned to integrate human expertise during classification as a future direction.

Recently, Diqi [25] incorporated generative adversarial networks for detecting spam on the Twitter platform. The method utilizes deep learning techniques to handle the large volume of Twitter data, achieving a higher G-Loss value, the effective ability of the method to handle the extensive data. The model can be further enhanced by incorporating deep belief networks and optimizing hyperparameters. Thomas and Meshram [26] described the Chimp Sailfish Optimization-based Deep Neuro Fuzzy Network to filter Twitter spam. The authors conducted experiments for Twitter spam detection and compared the results with machine learning and ensemble learning methods. They noted the necessity of using a data augmentation process in order to enhance the efficiency of Twitter spam detection. Choi et al. [27] combined the DT algorithm with machine learning and deep learning techniques. They hybridized the machine learning-based one-class SVM and deep learning-based autoencoder methods with DT separately to detect known and unknown spam by identifying anomalies and misuse on the Twitter platform. The DT algorithm effectively detected known spam, while anomaly detection methods were used for classifying regular tweets and unfamiliar spam. The authors conducted experiments using easily extractable Twitter features, which may lack performance for higher-dimensional features. Manasa et al. [28] presented a combination of bidirectional Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNN), and GLoVe vocabulary features for Twitter spam detection. The combinational model detects spam by processing the data in two modules. The initial module focuses on textual tweets, extracting vocabulary features using the GLoVe language model and detecting spam with the LSTM method. The second module incorporates metadata and respective meta-features related to tweets, using the CNN method for spam classification. The final spam detection was performed by combining the decisions of both modules.

The analysis of work pertaining to Twitter spam and spammer detection highlights several critical challenges:

- Spammers can mimic the behavior of legitimate users to attract and trap users for malicious and threatening activities which makes the process of detecting spammers complex with conventional methods.
- The existing studies primarily using standalone machine learning models, which lacks adaptability and robustness to counter evolving spam detection challenges.
- Moreover, existing studies indicate the usability of limited feature attributes and paucity of experiments conducted on standard benchmark Twitter datasets.

Therefore, the present work incorporates the integration of a machine learning based RF algorithm with advanced methods of QC and BGSA to determine Twitter spammers effectively. Additionally, experiments are conducted to analyze the significance of different feature types, and the impact of the spammer to non-spammer ratio is also evaluated.

3. Preliminaries

The section describes the fundamental concepts and methodologies related to the proposed QBGSRF method. These methodologies include the discussion of QC, GSA, and RF methods. The understanding of these preliminaries is essential to comprehend the proposed research methodology.

3.1. Quantum computing (QC)

The QC method leverages principles of quantum mechanics for computations [29]. The quantum computers, in contrast to classical computers, utilize qubits as data units instead of bits [30]. The superposition and entanglement phenomena of qubits allow them to occupy multiple states simultaneously. The states of qubits in QC can be represented by Eq. (1).

$$|\psi\rangle = c_1|0\rangle + c_2|1\rangle \tag{1}$$

Where, $|c_1|^2 + |c_2|^2 = 1$. In Eq. (1), $|0\rangle$ and $|1\rangle$ are the basic states with respect to the complex numbers c_1 and c_2.

The quantum gates can manipulate these qubits to solve the complex problems in an efficient manner compared to classical algorithms. For instance, the Hadamard gate can generate superpositions as described in Eq. (2).

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{2}$$

Additionally, the quantum entanglement attribute of QC entangles the qubits in a correlated state, irrespective of the distance between them. This attribute improves security, error correction, and the ability to solve complex problems [31].

3.2. Gravitational search algorithm (GSA)

The GSA is an optimization algorithm based on the principles of gravity and motion, and operates using a population-based approach [32]. The performance of the GSA is assessed by the masses of the agents, which interact with each other through gravitational force to optimize the solution [33,34]. The force acting on mass agent i by agent j is described by Eq. (3).

$$F_{ij}(t) = G(t)\frac{M_i(t) \times M_j(t)}{R_{ij}(t) + \varepsilon}\left(x_j(t) - x_i(t)\right) \tag{3}$$

Where, $G(t)$ is the gravitational constant, $M_i(t)$ and $M_j(t)$ are the masses with respect to agents i and j respectively, $R_{ij}(t)$ is Euclidean distance between mass agents, ε is a small constant, and $x_i(t)$ and $x_j(t)$ are the positions of the agents i and j, respectively.

The addition of stochastic attributes to incorporate randomness into mass agents changes the force acting on them [35], as expressed in Eq. (4).

$$F_i(t) = \sum_{j \in 1,\ j \neq i}^{k} rand_j F_{ij}(t) \tag{4}$$

Where, $rand_j$ is a random number that lies in the range [0, 1].

With the movement of agents, their position ($x_i(t+1)$) and velocity ($v_i(t+1)$) also updated, which are expressed by Eqs. (5), (6).

$$v_i(t+1) = rand_i \times v_i(t) + a_i(t) \tag{5}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{6}$$

These updates direct the agents towards the optimal solution through an iteration process.

3.3. Random forest (RF)

The RF algorithm is an ensemble learning method that is efficient for both regression and classification [36,37]. During the training process, it creates numerous decision trees and determines the most common class as the classification output and the average prediction of individual trees as the regression output [38,39]. The current work utilizes the RF algorithm for the classification process.

The algorithm classifies Twitter data by considering multiple bootstrap samples from the training dataset. A decision tree [40] is constructed with respect to each bootstrap sample. At each node, features are randomly selected, and the best split is determined on the basis of entropy, which is used to check the impurity of a node. The expression used to evaluate entropy is described in Eq. (7).

$$H = -\sum_{i=1}^{C} p_i \log(p_i) \tag{7}$$

Where, p_i is the probability of a data point to get classified into class i.

The prediction of user accounts is evaluated on the basis of data analysis by each tree of the forest. The final prediction is determined by selecting the class (spammer or non-spammer) on the basis of the majority vote received from all the trees.

4. Dataset, preprocessing and feature selection

This section offers a concise summary of the datasets, preprocessing modules, and selected features used in the detection of Twitter spammers. It outlines the sources and characteristics of the datasets, describes the preprocessing modules used to refine and analyze the data, and identifies the features selected to effectively detect Twitter spammers.

4.1. Datasets

The experiments utilized Twitter data from the 1KS-10KN dataset and the Social Honeypot dataset. The statistics of these datasets, including the count of tweets and users, are presented in Table 1.

The 1KS-10KN dataset was developed by Wang et al. [41] from Twitter data retrieved by Yang et al. [42]. Yang et al. [42] marked a user account as spammer if >10 % of its tweets were spam, with each tweet containing malicious links being marked as a spam tweet. Wang et al. [41] created the 1KS-10KN dataset by randomly choosing a single tweet from every user account, excluding duplicate and empty Twitter accounts. The authors collected 1000 spamming (1KS) and 9828 non-spamming (approx. 10KN) accounts, thus terming the dataset the 1KS-10KN dataset.

The Social Honeypot dataset was generated by Lee et al. [43] by deploying 60 honeypots to collect Twitter data over seven months, from December 30, 2009, to August 2, 2010. The Social Honeypot dataset, similar to the 1KS-10KN dataset, includes data from both spammers and non-spammers. The data was mainly collected based on four parameters: tweets of the top 10 trending topics, textual tweets, tweets containing redirecting links, and tweets replied to specific honeypots using @. The authors incorporated 22,223 spammers after removing 1646 short-term spammers, along with 19,276 non-spamming users.

4.2. Preprocessing

Preprocessing the raw textual Twitter data is crucial for improving the efficiency and efficacy of data analysis. It systematically processes the unprocessed data to narrow down the search space and makes the analysis process easier. This module ensures that the data is adequately prepared for further processing, which results in more reliable and insightful outcomes. The preprocessing stage encompasses the sub-steps of language normalization, tokenization, redirection link analysis, stopwords removal, and stemming [44].

The first sub-step of the preprocessing module is language normalization for translating non-English language tweets to English using an automated language translator. The datasets used include tweets in Korean, Japanese, and Chinese, in addition to English. The Google Translate API was employed for normalizing non-English tweets. The processed data is then tokenized using the NLTK (Natural Language Toolkit) Python library, which splits phrases into individual tokens based on spaces. Subsequently, the tokenized data is analyzed for redirect links. For extracting and analyzing HTML and XML files from web

Table 1
Statistics of twitter datasets.

Attribute	1KS-10KN Dataset	Social Honeypot Dataset
Spam Tweets	1000	2380,059
Non-Spam Tweets	9828	3263,238
Total Count of Tweets	10,828	5643,297
Spamming User	1000	22,223
Non-Spamming User	9828	19,276
Total Count of Users	10,828	41,499

URLs, the Python-based 'requests' library is used in combination with 'beautifulsoup4'. The extracted data is stored for further analysis to detect spam content. The fourth sub-step involves the removal of stop words, which is performed using the inbuilt list of the NLTK library for reducing data noise. The removal of stop words such as 'the', 'but', 'very', 'me', etc. helps to reduce the search space data for analysis and improves the performance efficacy. The final sub-step is stemming, which converts the expanded words to respective root words by eliminating any prefixes or suffixes. Here, data stemming is also conducted using the NLTK library.

4.3. Feature selection

The selection of precise and relevant features is crucial for the proposed method to classify Twitter spammers. The classification of Twitter users directly or indirectly depends on the available features. The present work has selected the features related to Twitter textual content, URL links, activities, user profiles, time, automation, and miscellaneous features [23,44,45]. These categories are prioritized due to direct correlation of their features with the spam behaviour observed in the dataset such as the frequent use of spam keywords, frequent usage of shortened URLs, and suspicious activity patterns. Features such as sentiment analysis are not incorporated as there is no direct relevance of these features with spam detection. Therefore, such features are neglected so that it does not impact on performance or increased complexities. This approach of consideration of only the relevant features allows to achieve optimal performance while minimizing computational resources and complexities.

The selected list of features for Twitter spammer detection is illustrated in Table 2.

a. **Textual features** include the analysis of textual content of the tweets posted by different users. These features are relevant to determine the frequency of certain spam keywords, the presence of commonly used spam phrases, the tweets similarity, etc. The analysis of textual features helps to determine the suspicious patterns used by spammers.
b. **URL features** are incorporated to analyze the tweets containing URL links, as spammers usually redirect these links to malicious or irrelevant websites. These features are related to URLs, including the frequency of links, the presence of shortened URLs, domains with digits in the links, etc. The URL features help to identify tweets that may redirect the users to harmful external websites or unlawful content.
c. **User activity-based features** involve the examination of user behaviour on the Twitter platform. These features include features such as frequency of tweets, retweets, mentions, etc. User activity features determine the abnormal patterns of spammers as they either retweet multiple tweets in a short interval or post tweets in an excessive manner.
d. **User profile features** also help to determine user behaviour, similar to user activity features, but based on the profile information available on the Twitter platform. In user profile features, features like account age, profile description, count of followers, and followings are considered. User accounts that are recently created or those with an imbalanced ratio of followers to followings are usually flagged as the protentional spammer.
e. **Time-based features** are also the significant features, as they are based on the timing of user activities. Feature such as the distribution of tweets over time and time interval between tweets can be considered in this category. Spammers may exhibit non-human and different timing patterns in their activities.
f. **Automation features** incorporate the features that suggest the usage of automated tools to post tweets. Automated tools generally try to mimic human patterns for tweet posting. However, automated tweeting patterns can be frequent and regular with the usage of bots or automated scripts to post tweets.
g. **Miscellaneous features** are additional relevant features apart from the previously mentioned categories that can be useful to detect spammers. These features are the ratio of tweets to retweets, the ratio of account age to tweets, and the ratio of followers to followings, etc. The considered list of miscellaneous features is mentioned in Table 2.

Table 2
A list of selected features for twitter spammer detection.

Textual Features	
Average tweet length	Count of words
Count of spam words	Count of hashtags
Count of user mentions	Count of digits per tweet
Count of retweets	Count of user replies
Tweet comment ratio	Repeated words in tweets
Tweet similarity	Uppercase letters
Frequency of spam keywords	Presence of common spam phrases
URL Features	
Count of URLs	Frequency of same URLs
Count of shortened URLs	Count of direct URLs
Average count of direct URLs	Average count of shortened URLs
Digits in domain links	URL length
URL ratio	URL authenticity
User Activity-based Features	
Frequency of tweets	Frequency of retweets
Time between tweets	Count of tweets per day
Tweet source	Tweet location
Count of repetitions	Frequency of user mentions
User Profile Features	
Age of account	Length of profile name
Count of followers	Count of followings
Length of profile description	Reputation score
Follower ratio	Following ratio
Time-based Features	
Time of tweet posted	Mean time between tweets
Idle time between tweets	Distribution of tweets over time
Automation Features	
Automated tweet ratio	Automated tweet URL ratio
Automated tweet similarity	
Miscellaneous Features	
Ratio of tweets to retweets	Ratio of direct to shortened URLs
Ratio of followers to followings	Ratio of replies to tweets
Ratio of tweets to URLs	Ratio of account age to tweets

5. Proposed QBGSRF method for spammer detection

The section describes the proposed QBGSRF method, which is utilized for Twitter spammer detection. The proposed methodology combines the strengths of QC via quantum mechanics, GSA via gravitational interactions, and RF via heuristic function and ensemble learning to effectively detect Twitter spammers. The integration of the aforementioned methods also aims to mitigate the challenges created by the dynamic and deceptive behaviour of spammers.

The QBGSA method is used for the optimization of the RF hyperparameters due to its superior ability to explore complex and large datasets. The quantum inspired attributes of QBGSA enhances its ability to handle feature optimization effectively which ensures the utilization of only the relevant features and tuning of hyperparameters. The selection of the RF algorithm is due to its robustness and ability to handle imbalanced and noisy datasets. Moreover, the decision tree-based architecture of the RF algorithm is effective in learning non-linear relationships which are common in the spam behaviour. The combination of the QBGSA and RF algorithms helps to maintain the exploration and exploitation balance, which leads to reduced higher performance accuracy and lower computational cost in the process of spam detection.

Methodology Overview:

The proposed QBGSRF method operates as a seven-tuple formulation as presented in Eq. (8).

$$\text{QBGSRF} = \left\langle (O, A \cup \{d\}), RF,\ DT(S), S, QMA_i, F_i^t, \theta_{ij}^{t+1} \right\rangle \tag{8}$$

- $(O, A \cup \{d\})$ indicates the decision table with respect to the incorporated problem of Twitter spammer detection that consists of a specific number of objects (O), attributes (A), and decision attributes (d).
- RF refers to a random forest consisting of z decision trees ($DT(S)$).
- $DT(S)$ are decision trees which are constructed by quantum mass agents (QMA_i) to determine the Twitter spammers.
- QMA_i is count of quantum mass agents.
- S is the set of objects acceptable in a node fulfilling the Taboo list function.
- F_i^t denotes the force exerted on the quantum mass agents (QMA_i).
- θ_{ij}^{t+1} indicates the change in the position of those agents.

The key components of the method are described as follows:

Step 1: Initialization of Parameters

The proposed QBGSRF method begin by initializing the parameters for the QC, BGSA, and RF algorithms. The algorithm involves the initialization of the following parameters:

- **Quantum Mass Agents (QMA_i)**: The population of QMA_i agents ($i = 120$) is initialized by assigning their initial quantum states and rotation angles within the range $\left[0,\ \frac{\pi}{2}\right]$, ensuring the QC formulations.
- **Random Forest Parameters**: The count of the decision trees is set to z=100, and the node splitting rule such as Twoing rule are defined for optimal tree construction.
- **Algorithm Control Parameters**: The maximum number of iterations ($t_{max} = 500$) are defined to control the convergence of the algorithm.
- **Optimization Constants**: The gravitational constant (ϑ_{max}=10) is defined at initial stage which decreases adaptively with iterations, while learning rate ($\propto = 0.01$) controls the speed of the position updates for QMA_i.

The values of these parameters are tuned to balance the exploration and exploitation of the proposed algorithm for the efficient spammer detection.

Step 2: Generation of Initial Population

The initial population of QMA_i agents is generated using quantum principles. Each agent is represented as a string of qubits which follows the normalization condition of $|\sin(\theta_{ij})|^2 + |\cos(\theta_{ij})|^2 = 1$.

The fulfilment of the normalization condition maintains quantum coherence and diversity in the search space. This initialization leverages the inherent randomness of quantum states to ensure the diverse set of candidate solutions which optimizes the search space.

Step 3: Conversion of GSA to BGSA

In QBGSRF method, the GSA algorithm is converted to binary GSA (BGSA) to add the quantum attributes. In this process the binary agents flip the agent state in binary form, and real search space is converted to binary search space as a hypercube. In BGSA, the velocity of the agents affects the probability to select the new binary position [46]. The agents with higher velocity have a higher probability to change the position, and agents with lower velocity have a lower probability to change their binary position, thereby retaining their previous best position [47]. This can be expressed by (9).

$$x_i^{t+1} = \begin{cases} 1, & rand_i < \dfrac{1}{1 + e^{-x_i^{t+1}}} \\ 0, & otherwise \end{cases} \tag{9}$$

Where, $rand_i$ is a uniformly distributed random number in the interval [0, 1].

Step 4: Integration of QC to BGSA

The quantum attributes are added with BGSA method which enables the better exploration of the solution space by introducing randomness and providing a broader diversity in potential solutions. The states of qubits are determined with quantum gates, specifically the rotation gate because of its promising outcomes in the previous studies [48,49]. Eq. (10) provides the solution for the quantum mass agents (QMA_i) using the rotation gate.

$$QMA_i = \begin{bmatrix} \cos(\Delta\theta_i) & -\sin(\Delta\theta_i) \\ \sin(\Delta\theta_i) & \cos(\Delta\theta_i) \end{bmatrix} \tag{10}$$

Where, $\Delta\theta_i$ is the rotation angle with respect to agents ($i = 1,2,\ldots,n$). Each quantum mass agent solution (QMA_i) can be described as the string of qubits. The initial solutions are retained with higher diversity and maintain the normalization criteria of $|\sin(\theta_{ij})|^2 + |\cos(\theta_{ij})|^2 = 1$. The initial solutions are presented with Eq. (11) by considering the $1 < j \le m$ and $\theta_{ij} \in \left[0,\ \frac{\pi}{2}\right]$.

$$QMA_i^{t=0} = \begin{bmatrix} \cos(\theta_{i1}^0) & |\cos(\theta_{i2}^0) & |\ldots & |\cos(\theta_{im}^0) \\ \sin(\theta_{i1}^0) & |\sin(\theta_{i2}^0) & |\ldots & |\sin(\theta_{im}^0) \end{bmatrix} \tag{11}$$

The force acting on these agents (QMA_i) can be evaluated using Eq. (12).

$$F_{ij}^t = \vartheta \sum_{j \in 1,\ j \neq i}^{k} rand_j \frac{M_i^t \times M_j^t}{R_{ij}^t + \mathcal{E}} \left(\theta_j^t - \theta_i^t\right) \tag{12}$$

Where, ϑ indicates gravitational constant whose value decreases from ϑ_{max} to ϑ_{min} on the basis of rotation angle and iterations. θ_j^t and θ_i^t are the quantum states of agents. The distance R_{ij}^t between two quantum agents is evaluated using hamming distance formulations. Further, the acceleration of the agents at time (t) is evaluated using Eq. (13).

$$a_{ij}^t = \frac{F_i^t}{M_{ii}^t} = \vartheta \sum_{k \in Kbest,\ k \neq i} \left[rand_j \times \delta_i^k \times \left(\theta_{kj}^t - \theta_{ik}^t\right)\right] \tag{13}$$

The notation $\mathcal{E}$ is neglected in Eq. (13) due to its constancy. $Kbest_{jk}^t$ is the k^{th} bit of the best mass agents. The symbol δ_i^k indicates the division of the mass ($M_j{}^t$) to distance (R_{ij}). The value of the δ_i^k is calculated by Eqs. (14), (15).

$$\delta_i^k = \begin{cases} \varphi_i^k + 1, & if\ f(\theta_k^t) = f(\theta_{Kbest}^t) \\ \varphi_i^k, & otherwise \end{cases} \tag{14}$$

Where,

$$\varphi_i^k = \begin{cases} 1, & if\ M_k > M_i\ and\ R_{ik} \le \tau \\ 0, & otherwise \end{cases} \tag{15}$$

Where, the notation τ denotes the maximum possible distinct bits among the total count of bits between i^{th} and k^{th} quantum agents who can exert the active force on i^{th} agent.

Step 5: Construction of Random Forest

Each QMA_i constructs a decision tree using the randomness attribute of the RF algorithm, which are ensembled to determine the final decision for the Twitter user to declare as spammer or non-spammer. The attributes of each node in a decision tree are tested by expression $test : O \rightarrow R_{test}$. Here, the list of possible tests is described by $R_{test} = \{r_1, r_2, \ldots r_z\}$. The suitability expression for the $test$ change to $test : A \rightarrow R_{test}$ for the of the attributes $a : O \rightarrow A$.

At each node of the tree, the potential sub-trees $(T_1, T_2, \ldots, T_z)$ can be constructed with test cases $(r_1, r_2, \ldots, r_z)$ under the premise that T_i sub-trees are generated with r_i tests. This can be expressed by the hypoth-

esis of Eq. (16).

$$h(x) = \begin{cases} h_1(x), test(x) = r_1 \\ h_2(x), test(x) = r_2 \\ \quad \vdots \\ h_z(x), test(x) = r_z \end{cases} \tag{16}$$

In the proposed QBGSRF method, the optimal split of the trees can be obtained by calculating the heuristic function ($\eta_{A_i,\ V_j}$) using the Twoing splitting measures with respect to attributes (A_i) and values (V_j). The evaluation of $\eta_{A_i,\ V_j}$ is expressed in Eq. (17).

$$\eta_{A_i,\ V_j} = \frac{P_l P_r}{4}\left[\sum_{d=1}^{D} \left| p\left(d|node_{l(A_i,\ V_j)}\right) - p\left(d|node_{r(A_i,\ V_j)}\right)\right|\right]^2 \tag{17}$$

Where, D represents the largest count of potential decision classes. The notations P_l and P_r are the probabilities associated with left and right nodes. The conditional probabilities for these left and right nodes are denoted as $p\left(d|node_{l(A_i,\ V_j)}\right)$ and $p\left(d|node_{r(A_i,\ V_j)}\right)$, respectively.

Step 6: Update Positions of Agents

The position of the agents is updated by following the principles of QC. Here, the movement of the QMA_i is determined as illustrated by Eqs. (18), (19).

$$\Delta\theta_{ik}^{t+1} = rand_i \times \Delta\theta_{ik}^{t} + a_{ij}^{t} \tag{18}$$

$$\theta_{ij}^{t+1} = \theta_{ik}^{t} + \Delta\theta_{ik}^{t+1} \tag{19}$$

After the movement of QMA_i, the new quantum population is described by Eq. (20), which is determined by changing the rotation angle as per QBGSA strategy.

$$QMA_i^{t+1} = \begin{bmatrix} \cos\left(\theta_{i1}^{t+1}\right) & \left|\cos\left(\theta_{i2}^{t+1}\right)\right. & \left|\ldots\right. & \left|\cos\left(\theta_{im}^{t+1}\right)\right. \\ \sin\left(\theta_{i1}^{t+1}\right) & \left|\sin\left(\theta_{i2}^{t+1}\right)\right. & \left|\ldots\right. & \left|\sin\left(\theta_{im}^{t+1}\right)\right. \end{bmatrix} \tag{20}$$

Step 7: Ensemble Decision for Twitter Spammers

The final match of the Twitter users as spammer or non-spammer is determined by the combined outcomes from each decision tree $DT(S)$. The decision of each $DT(S)$ is evaluated by Eqs. (21), (22).

$$\epsilon(DT(S),\ Dstn) = \sum_{(x,\ y)\in U} Dstn(x,\ y).L(y, DT(S)(x)) \tag{21}$$

Where,

$$L(y,\ DT(S)(x)) = \begin{cases} 1,\ if\ y \neq DT(S)(x) \\ 0,\ if\ y = DT(S)(x) \end{cases} \tag{22}$$

Where, *Dstn* indicate the distribution parameter.

Step 8: Iterative Optimization

The result outcomes of each decision tree with respect to each Twitter user are combined as per the principles of the RF algorithm with voting criteria. The diversity of the attributes helps to construct different potential decision trees with diverse combinations, and hence the ensemble decision forest outcomes are obtained by combining the decision results of all the quantum agents with the completion of iterations. The steps of 3 to 7 are repeated until the maximum number of iteration (t_{max}) is reached or the solution converges.

The pseudo code of the proposed QBGSRF method is described by Algorithm 1.

Algorithm 1

Pseudo code of the proposed QBGSRF method for twitter spammer detection.

Initialize the parameters relative to QC, BGSA, and RF algorithms.
RF=*null*;
t=1;
while ($t \leq t_{max}$) **do**
for (j = 1 *to* DT_i) **do**
DT_{best}= *null*;
spammer_classifier=*choose_objects* //Incorporate QMA_i for the Twitter user data with equal probability.
for (N = 1 *to* QMA_i) **do**
// Construct DT_i by incorporating a subset of attributes at each node using the principles of the QBGSRF method.
DT_{new} = DT_i_*Construction*;
if (Quality (DT_{new}) > Quality (DT_{best}))) **then**
DT_{best}= DT_{new};
end if
end for
Update the movement of the QMA_i using Eqs. (18), (19).
RF.add (DT_{best});
end for
end while
Outcome = Final classification outcome class of Twitter spammers.

Table 3
Confusion matrix components for twitter spammer detection.

Component	Description
True Positive (TP)	Twitter spammer classified as spammer
False Positive (FP)	Twitter spammer classified as non-spammer
True Negative (TN)	Twitter non-spammer classified as non-spammer
False Negative (FN)	Twitter non-spammer classified as spammer

6. Results and discussion

The proposed QBGSRF method is assessed for its performance in the experiments conducted on the 1KS-10KN and Social Honeypot Dataset with performance assessment in terms of recall, precision, and F-measure. The computations of these measures are obtained from the components of the confusion matrix, namely true positive (TP), false positive (FP), true negative (TN), and false negative (FN). These components are described in Table 3, and the assessment measures are formulated in Table 4.

For the experiments, the datasets are partitioned into training and testing ratios of 75:25. The division of Twitter users into training and testing sets is presented in Table 5. To assess the efficacy of the proposed QBGSRF method, the results are also evaluated utilizing the BGSRF method (a combination of BGSA and RF algorithm) and the RF algorithm alone. The confusion matrix results for experiments on the 1KS-10KN and Social Honeypot datasets are provided in Tables 6 and 7, respectively. The performance assessment outcomes for these datasets are summarized in Table 8.

The performance assessment results, as shown in Table 8 indicate the effective performance of the proposed QBGSRF method as well as the presented combinational BGSRF method. In contrast, the results of the RF algorithm are lacking, with F-measure scores of 86.45 % for the 1KS-10KN dataset and 88.89 % for the Social Honeypot dataset. The proposed QBGSRF method attained superior performance results, with F-measure scores of 98.38 % for the 1KS-10KN dataset and 97.87 % for the Social Honeypot dataset.

The experiments are performed on the system with configuration of Intel i7 CPU, NVIDIA RTX 3080 GPU, and 16GB RAM. The overall computational cost and runtime of the algorithms is evaluated in Table 9.

From table 9, it can be noticed that the GPU acceleration has

Table 4
Formulations of the assessment measures for twitter spammer detection.

Assessment Measure	Formulation
Recall	$\frac{TP}{TP+FN}$
Precision	$\frac{TP}{TP+FP}$
F-Measure	$2\times\frac{Recall\times Precision}{Recall+Precision}$

Table 5
Twitter users split as per training and testing proportion of 75:25.

Proportion	Twitter Users	1KS-10KN Dataset	Social Honeypot Dataset
Training (75 %)	Spammers	750	16,667
	Non-spammers	7371	14,457
Testing (25 %)	Spammers	250	5556
	Non-spammers	2457	4819

Table 6
Confusion matrix results for experiments on the 1KS-10KN dataset.

(a) Proposed QBGSRF Method		
	Spammers	Non-spammers
Spammers	TP = 243	FP = 01
Non-spammers	FN = 07	TN = 2456
(b) BGSRF Method		
	Spammers	Non-spammers
Spammers	TP = 240	FP = 02
Non-spammers	FN = 10	TN = 2455
(c) RF Method		
	Spammers	Non-spammers
Spammers	TP = 201	FP = 14
Non-spammers	FN = 49	TN = 2443

Table 7
Confusion matrix results for experiments on the social honeypot dataset.

(a) Proposed QBGSRF Method		
	Spammers	Non-spammers
Spammers	TP = 5410	FP = 89
Non-spammers	FN = 146	TN = 4730
(b) BGSRF Method		
	Spammers	Non-spammers
Spammers	TP = 5361	FP = 127
Non-spammers	FN = 195	TN = 4692
(c) RF Method		
	Spammers	Non-spammers
Spammers	TP = 4822	FP = 471
Non-spammers	FN = 734	TN = 4348

significantly reduced the computation time for QBGSRF method by approximately 55.56 % on the Social Honeypot Dataset, utilizing around 4 GB of GPU memory. CPU memory usage peaked at 10GB during processing. For the 1KS-10KN dataset, the computation time was reduced to approximately 62.7 % with GPU memory usage at 2.5 GB, compared to 6GB memory usage on the CPU. The decision tree construction in RF leveraged multi-threading, whereas QBGSRF benefited from parallel quantum state computations on the GPU. The key advantage of the

Table 8
Performance assessment results.

Method	Recall (%)	Precision (%)	F-Measure (%)
1KS-10KN Dataset			
Proposed QBGSRF	97.20	99.59	98.38
BGSRF	96	99.17	97.56
RF	80.40	93.49	86.45
Social Honeypot Dataset			
Proposed QBGSRF	97.37	98.38	97.87
BGSRF	96.49	97.69	97.08
RF	86.79	91.10	88.89

Table 9
Computational cost and runtime of the experiments.

Method	Runtime (CPU)	Runtime (GPU)	Memory Usage (CPU)	Memory Usage (GPU)
1KS-10KN Dataset				
Proposed QBGSRF	20 min	7.46 min	6GB	2.5GB
BGSRF	15.43 min	N/A	6GB	N/A
RF	11.87 min	N/A	6GB	N/A
Social Honeypot Dataset				
Proposed QBGSRF	45 min	10.26 min	10GB	4GB
BGSRF	27.82 min	N/A	10GB	N/A
RF	21.35 min	N/A	10GB	N/A

QBGSRF method lies in its ability to handle complex dataset pattern in spam detection by incorporating the benefits of quantum state computations and GPU acceleration. While GPU usage for the QBGSRF method significantly reduces the computation time, it adds the computational costs.

The superior performance of the proposed QBGSRF method, as depicted in Tables 8 and 9, is attributed to the careful selection of the hyperparameters. There is the major role of following parameters to achieve the evaluated results:

1. **Quantum Mass Agents (QMA_i)**: The considered population of 120 QMA_i ensured diversity in the optimization process, which enables the algorithm to effectively utilize the feature space for both datasets. For the 1KS-10KN dataset, the higher count of 150 agents was tested but resulted in negligible performance gains (approximately 0.2 %) but increased the runtime by approximately 17 %. On the other hand, the lower count of 100 agents led to an approximately 1.8 % decrease in the f-measure score due to insufficient exploration. For the Social Honeypot dataset, the increase in the count of agents led to an increase in the runtime of approximately 22 % with no efficient performance gain, but reducing the agents to 100 led to a decrease in the F-measure score of approximately 2.2 %. This indicates the selection of 120 agents maintains an optimal balance between the computational efficiency and performance across both datasets.
2. **Maximum Iterations (t_{max})**: The consideration of a maximum count of iterations of 500 allowed the algorithm to effectively converge. The reduction in the value of t_{max} to 300 shortened the runtime by approximately 10 % but also led to the reduction in the F-measure score by 1.9 % for the 1KS-10KN dataset and 2.4 % for the Social Honeypot dataset. This reduction in the value of t_{max} points towards the premature convergence.
3. **Learning Rate ($\propto$)**: The incorporated learning value of 0.01 effectively controls the update in the quantum state, which ensures the stability in feature optimization. The higher value of $\propto$ (such as 0.02, 0.03, etc.) caused instability, while a lower value (such as 0.005) slowed down the optimization process without an efficient increment in the performance.

4. **Random Forest Parameters (z)**: The included value of random forest parameter ($z = 100$) provided an optimal balance between the computation cost and performance of the algorithm. An increase in the count of trees to 150 marginally improved the recall value (approximately 0.5 % for the 1KS-10KN dataset and approximately 0.6 % for the Social Honeypot dataset) but led to an increase in runtime by approximately 14 % and 16 %, respectively. On the other hand, reducing the tree count to 80 reduced the memory usage but decreased precision by approximately 1 % and 1.3 %, respectively, for the 1KS-10KN and Social Honeypot datasets.

Thus, the combination of the above-mentioned tuned hyper-parameters and GPU acceleration enabled the proposed QBGSRF method to process the large datasets efficiently.

Further, two different experiments are conducted to analyze the significance of different feature types, and to assess the impact of the spammer to non-spammer ratio on the performance of the proposed QBGSRF, as well as the other incorporated methods.

6.1. Experiment 1: significance of different feature types

This experiment determines the significance of the distinct feature types incorporated in the process of detecting Twitter spammers. The experiment analyzes the efficacy of each feature category, and their effectiveness is assessed based on the F-measure score. The performance results are calculated for each individual feature as well as combined feature categories using RF, BGSRF, and proposed QBGSRF methods. The F-measure results for the experiments on the 1KS-10KN and Social Honeypot datasets are illustrated in Tables 10 and 11, respectively.

In both datasets, the significance of different features varies due to differences in data size, and accordingly, the importance of feature attributes changes. In both datasets, the performance of the methods is superior for the combined feature categories compared to the performance evaluated for the individual feature categories.

For the 1KS-10KN dataset, the performance outcomes presented in Table 10 indicate the superior significance of the user profile-based features with maximum F-measure among the individual feature categories, and the inferior performance of the time-based features with minimum F-measure. The significance of the incorporated features in descending order is described as follows: user profile, textual, URL, miscellaneous, automation, user activities, and time-based features. The significance order is consistent for the proposed QBGSRF method, as well as for BGSRF and RF methods.

For the Social Honeypot dataset, the performance outcomes illustrated in Table 11 highlight the higher significance of textual features with a higher F-measure, and the lower significance of time-based features with low F-measure among the individual features. The significance of these features in descending order is presented as follows: textual, user activities, user profile, automation, miscellaneous, URL, and time-based features. Similar to the 1KS-10KN dataset, the significance order for the Social Honeypot dataset experiments is also the same for all the proposed QBGSRF method, BGSRF method, and RF method.

The results from experiments on both datasets indicate the inferior significance of the time-based features in comparison to other feature categories. The analysis of feature significance reveals varying importance of feature categories for both datasets. In the case of Social Honeypot dataset, the textual, user activities, and user profile features are among the top three superior features, whereas the significance of user activities changes to one of the inferior feature categories for the 1KS-10KN dataset due to the dataset containing a single tweet per user. Similarly, while URL features are significant for the 1KS-10KN dataset, they rank as the second least significant in the Social Honeypot dataset. These variations in feature significance are influenced by dataset size and the available attributes.

Table 10
Significance of feature categories for experiments on 1KS-10KN dataset.

Feature	RF	BGSRF	QBGSRF
Textual Features	67.32 %	72.47 %	77.35 %
URL Features	54.48 %	63.43 %	69.60 %
User Activities based Features	39.05 %	43.36 %	48.05 %
User Profile Features	69.88 %	76.091 %	81.47 %
Time based Features	31.27 %	35.70 %	38.13 %
Automation Features	42.54 %	44.40 %	51.29 %
Miscellaneous Features	49.53 %	58.61 %	64.77 %
All Features	86.45 %	97.56 %	98.38 %

Table 11
Significance of feature categories for experiments on social honeypot dataset.

Feature	RF	BGSRF	QBGSRF
Textual Features	74.66 %	83.39 %	86.56 %
URL Features	42.17 %	46.17 %	57.58 %
User Activities based Features	67.35 %	72.67 %	78.40 %
User Profile Features	65.12 %	66.65 %	76.43 %
Time based Features	29.59 %	33.78 %	41.02 %
Automation Features	54.83 %	62.27 %	64.47 %
Miscellaneous Features	49.41 %	56.11 %	62.45 %
All Features	88.89 %	97.08 %	97.87 %

6.2. Experiment 2: impact of spammer to non-spammer ratio

This experiment evaluates the impact of the spammer to non-spammer ratio on the effectiveness of the proposed QBGSRF method. The proportion of spammer and non-spammer users is different in the original datasets of 1KS-10KN and the Social Honeypot dataset. In the 1KS-10KN dataset, as the name suggests, the proportion of non-spam users is approximately 10 times that of spam users. For the Social Honeypot dataset, there is also a difference in the spammer and non-spammer user ratio.

For this experiment, four different proportions of spammer to non-spammer users are considered: 1:1, 1:2, 1:5, and 1:10. The respective data for these proportions is described in Table 12. To attain the aforementioned proportions, the data of 1000 spam users and 9828 non-spam users respective to the 1KS-10KN dataset is utilized, while for the Social Honeypot dataset, the data of 19,276 spam and an equal number of non-spam users is considered. In the case of the 1KS-10KN dataset, the spam users are only 1000 in the dataset, therefore, the count of non-spam users is increased with the increasing proportion of non-spam users. Conversely, in the case of the Social Honeypot dataset, the count of non-spam users is considered fixed, and the count of spam users is decreasing with the increasing proportion of non-spam users. This varying proportion of spam and non-spam users will allow to evaluate the proposed method for the diverse experiments.

For performance evaluations, the 75 % of the data is randomly selected for training, while the remaining 25 % is utilized for testing. The impact of the spammer to non-spammer ratio of Twitter users on the performance of the methods is illustrated in Tables 13 and 14 for the 1KS-10KN and Social Honeypot datasets, respectively. These results are assessed using the F-measure as the assessment metric.

The results evaluated in Table 13 for the 1KS-10KN dataset show that the proposed QBGSRF method achieved lower F-measure score of 96.36

Table 12
Statistics of twitter datasets to analyze the impact of spammer to non-spammer ratio.

	1KS-10KN Dataset		Social Honeypot Dataset	
Ratio	Spammer	Non-Spammer	Spammer	Non-Spammer
1:1	1000	1000	19,276	19,276
1:2	1000	2000	9638	19,276
1:5	1000	5000	3855	19,276
1:10	1000	9828	1928	19,276

Table 13
Impact of spamer to non-spammer ratio for experiments on the 1KS-10KN dataset.

Ratio	RF	BGSRF	QBGSRF
1:1	82.57 %	95.93 %	96.36 %
1:2	84.36 %	96.15 %	96.96 %
1:5	85.15 %	96.76 %	97.79 %
1:10	86.33 %	97.58 %	98.38 %

Table 14
Impact of spammer to non-spammer ratio for experiments on the social honeypot dataset.

Ratio	RF	BGSRF	QBGSRF
1:1	88.36 %	97.22 %	98.41 %
1:2	88.11 %	95.88 %	97.25 %
1:5	85.65 %	95.63 %	96.76 %
1:10	84.08 %	93.40 %	95.70 %

% for the 1:1 proportion of spam to non-spam users compared to the original dataset proportion. This value increases to 96.96 % for the 1:2 proportion, 97.79 % for the 1:5 proportion, and 98.38 % for the 1:10 proportion. Here, the proportion 1:10 is the original value of data, which was considered during the initial experiment without any adjustments to the proportion of spam or non-spam users. These findings indicate an improvement in the performance of the proposed method as the proportion of non-spam users increases in the 1KS-10KN dataset.

Similarly, the results in Table 14 for the Social Honeypot dataset show an increase in the F-measure to 98.41 % for the 1:1 proportion, which was only 97.87 % in the original dataset proportion using the proposed QBGSRF method. However, this value decreases to 97.25 % for the 1:2 proportion, 96.76 % for the 1:5 proportion, and 95.70 % for the 1:10 proportion. This indicates the decrease in the performance of the proposed method as the proportion of spam users decreases in the Social Honeypot dataset.

The lower performance of the 1KS-10KN dataset for the 1:1

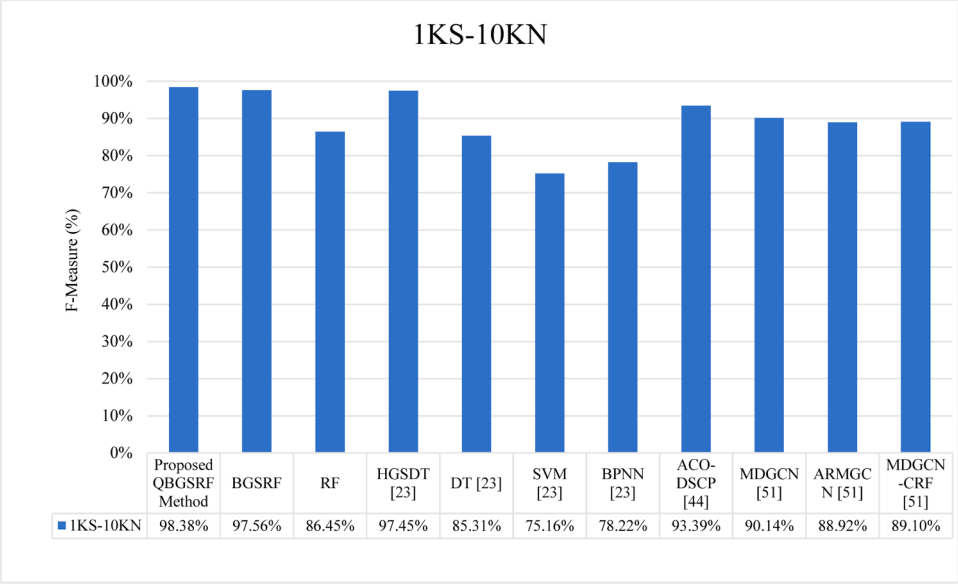


Fig. 2. Comparison of the proposed QBGSRF method with state-of-the-art methods for experiments on the 1KS-10KN dataset.

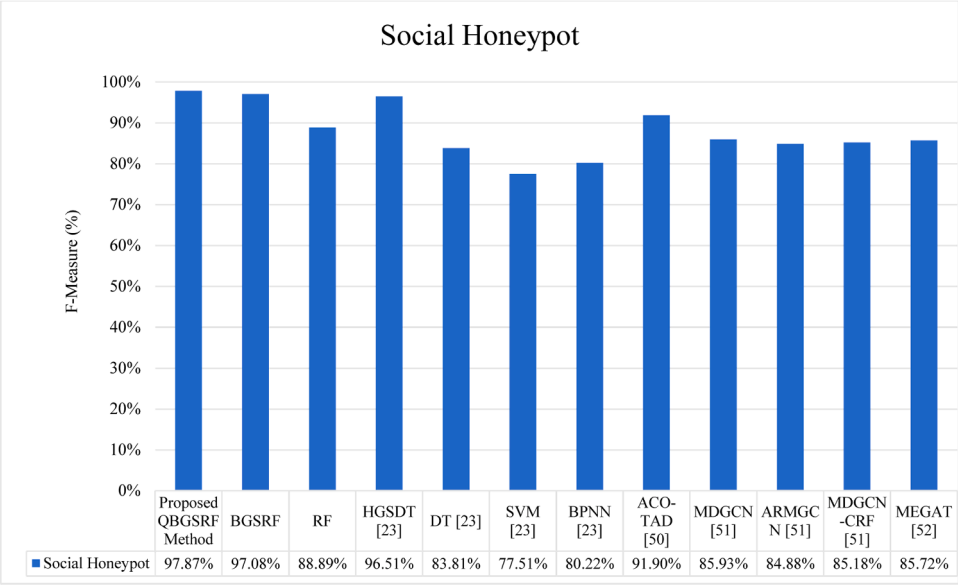


Fig. 3. Comparison of the proposed QBGSRF method with state-of-the-art methods for experiments on the social honeypot dataset.

proportion compared to the 1:10 proportion is due to the limited data availability in the 1KS-10KN dataset, specifically the presence of only a single tweet per user. On the other hand, the proposed method outperformed in the case of the Social Honeypot dataset for the 1:1 proportion compared to the 1:10 proportion.

The other methods (BGSRF and RF) exhibit similar patterns for both the 1KS-10KN and Social Honeypot datasets, as illustrated in Tables 13 and 14, respectively. The evaluation of higher F-measure values using the proposed QBGSRF method ensures the robust behavior of the proposed methodology.

6.3. Comparison

The proposed QBGSRF method is compared with the BGSRF method, the RF algorithm, and several state-of-the-art methods. The incorporated methods are presented by Vives et al. [23], Kumari and Balkishan [44, 50], Deng et al. [51] and Tripathi et al. [52].

Vives et al. [23] evaluated the results for the 1KS-10KN and Social Honeypot datasets using the Hybrid Gravitational Search Algorithm and Decision Tree (HGSDT) method, Decision Tree (DT), Support Vector Machine (SVM), and Back Propagation Neural Networks (BPNN). Kumari and Balkishan [44] conducted the experiments on the 1KS-10KN dataset using the Ant Colony Optimization based Detection of Suspicious Content and Profile (ACODSCP) method. Further, Kumari and Balkishan [50] performed the experiments on the Social Honeypot dataset using the Ant Colony Optimization based Threatening Account Detection (ACOTAD). The performance comparison is illustrated in Figs. 2 and 3 in terms of F-measure for the 1KS-10KN and Social Honeypot datasets, respectively.

Deng et al. [51] evaluated the results for both the 1KS-10KN and Social Honeypot datasets using Markov-Driven Graph Convolutional Network (MDGCN), hybrid Adaptive Reward Markov Random Field with Graph Convolutional Networks (ARMGCN), and integration of MDGCN with conditional random fields (MDGCN-CRF). Tripathi et al. [52] determined the results for Social Honeypot dataset using Markov Enhanced Graph Attention Network (MEGAT).

The comparison graphs illustrated in Figs. 2 and 3 demonstrate the superior performance of the proposed QBGSRF method compared to state-of-the-art methods for Twitter Spammer detection. These results also highlight the improvement in the Twitter Spammer detection in an effective manner compared to other methods.

7. Conclusion

This paper presented a novel QBGSRF method to effectively detect Twitter spammers. The proposed method combined the principles of quantum mechanics via the QC algorithm, the metaheuristic abilities of the BGSA method, and the machine learning method of the RF algorithm. The combined attributes of the aforementioned methods enable quantum agents to quickly determine effective Twitter spammers by constructing decision trees based on the principles of the RF algorithm. The performance results of the proposed QBGSRF method were evaluated for the 1KS-10KN and Social Honeypot datasets, and attained the effective performance f-measure scores of 98.38 % and 97.87 %, respectively. The results are also calculated using BGSRF and RF methods, but these methods lack in comparison to the QBGSRF method.

Additionally, two different experiments were conducted to access the significance of different feature categories and the impact of the Twitter spammer to non-spammer ratio. The results revealed that user profile-based features were most significant for the 1KS-10KN dataset, while textual features were most important for the Social Honeypot dataset. The time-based features were lacking in both datasets compared to other features. Regarding the impact of the spammer to non-spammer ratio, the proposed method noticed a lower performance score in the 1:1 proportion compared to the 1:10 proportion for the 1KS-10KN dataset. Conversely, it determined a higher performance score in the 1:1 proportion compared to the 1:10 proportion for the Social Honeypot dataset.

In the overall scenario, the proposed QBGSRF method outperformed the existing state-of-the-art methods, demonstrating its effectiveness in Twitter spammer detection.

CRediT authorship contribution statement

Kanta Prasad Sharma: Writing – original draft, Software, Methodology, Conceptualization. **Gendal Lal:** Validation, Resources, Funding acquisition, Data curation. **Madhu Shukla:** Writing – original draft, Resources, Project administration, Funding acquisition. **Anupam Yadav:** Validation, Software, Investigation, Data curation. **Jayaprakash B:** Writing – original draft, Validation, Project administration, Investigation. **Bhanu Juneja:** Writing – review & editing, Software, Investigation, Data curation. **Jayant Jagtap:** Validation, Project administration, Investigation, Data curation. **Amrita Singh:** Writing – original draft, Supervision, Methodology, Conceptualization. **A. Bhowmik:** Writing – review & editing, Software, Investigation, Formal analysis. **A. Johnson Santhosh:** Writing – review & editing, Validation, Methodology, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] M.M.D. Khomami, M.R. Meybodi, A. Rezvanian, Efficient identification of maximum independent sets in stochastic multilayer graphs with learning automata, Results Eng. (2024) 103224.
[2] M. Thomas, B.B. Meshram, Chso-DNFNet: spam detection in Twitter using feature fusion and optimized deep neuro fuzzy network, Adv. Eng. Softw. 175 (2023) 103333.
[3] M.A. Salman, M.A. Mahdi, Nifty method for prediction dynamic features of online social networks from users' activity based on machine learning, Results Eng. 20 (2023) 101430.
[4] Stacy Jo Dixon, 2024, Number of global social network users 2017-2028, online available: https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/.
[5] X. Chen, Z.M. Lu, A real-time method to predict social media popularity, Int. J. Mod. Phys. C 28 (12) (2017) 1750144.
[6] E. Rosenberg, C. Tarazona, F. Mallor, H. Eivazi, D. Pastor-Escuredo, F. Fuso-Nerini, R. Vinuesa, Sentiment analysis on Twitter data towards climate action, Results Eng. 19 (2023) 101287.
[7] Stacy Jo Dixon, 2024, Global social networks ranked by number of users 2024, online available: https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/.
[8] H. Kwak, C. Lee, H. Park, S. Moon, What is twitter, a social network or a news media?, in: Proceedings of the 19th International Conference on World wide web, 2010, pp. 591–600.
[9] Stacy Jo Dixon, 2024, X/Twitter: distribution of global audiences 2024, by age group. Online available: https://www.statista.com/statistics/283119/age-distribution-of-global-twitter-users/.
[10] Stacy Jo Dixon, 2024, X/Twitter: distribution of Global Audiences 2024, by Gender. Online Available: https://www.statista.com/statistics/828092/distribution-of-users-on-twitter-worldwide-gender/.
[11] Statista Research Department, 2024, X/Twitter: countries with the largest audience 2024. Online Available: https://www.statista.com/statistics/242606/number-of-active-twitter-users-in-selected-countries/.
[12] P. Manasa, A. Malik, I. Batra, Detection of twitter spam using GLoVe vocabulary features, bidirectional LSTM and convolution neural network, SN Comput. Sci. 5 (2) (2024) 206.
[13] S.B. Abkenar, M.H. Kashani, M. Akbari, E. Mahdipour, Learning textual features for twitter spam detection: a systematic literature review, Expert. Syst. Appl. 228 (2023) 120366.
[14] S. Jindal, M. Sachdeva, A.K.S. Kushwaha, A novel quantum-behaved binary firefly algorithm with gravitational search algorithm to optimize the features for human activity recognition, Int. J. Mod. Phys. C 33 (11) (2022) 2250146.

[15] Asha, Deep neural networks-based classification optimization by reducing the feature dimensionality with the variants of gravitational search algorithm, Int. J. Mod. Phys. C 32 (10) (2021) 2150137.
[16] H. Tajalizadeh, R. Boostani, A novel stream clustering framework for spam detection in twitter, IEEe Trans. Comput. Soc. Syst. 6 (3) (2019) 525–534.
[17] X. Wang, Q. Kang, J. An, M. Zhou, Drifted Twitter spam classification using multiscale detection test on KL divergence, IEEe Access. 7 (2019) 108384–108394.
[18] G. Lingam, R.R. Rout, D.V. Somayajulu, S.K. Ghosh, Particle swarm optimization on deep reinforcement learning for detecting social spam bots and spam-influential users in twitter network, IEEe Syst. J. 15 (2) (2020) 2281–2292.
[19] K.N. Güngör, O. Ayhan Erdem, İ.A. Doğru, Tweet and account based spam detection on twitter, Artificial Intelligence and Applied Mathematics in Engineering Problems:, in: Proceedings of the International Conference on Artificial Intelligence and Applied Mathematics in Engineering (ICAIAME 2019), Springer International Publishing, 2020, pp. 898–905.
[20] S.B. Abkenar, E. Mahdipour, S.M. Jameii, M. Haghi Kashani, A hybrid classification method for twitter spam detection based on differential evolution and random forest, Concurr. Comput.: Pract. Exp. 33 (21) (2021) e6381.
[21] S.B.S. Ahmad, M. Rafie, S.M. Ghorabie, Spam detection on Twitter using a support vector machine and users' features by identifying their interactions, Multimed. Tools. Appl. 80 (8) (2021) 11583–11605.
[22] K.U. Santoshi, S.S. Bhavya, Y.B. Sri, B. Venkateswarlu, Twitter spam detection using naïve bayes classifier, in: Proceedings of the 2021 6th International Conference on Inventive Computation Technologies (ICICT), IEEE, 2021, pp. 773–777.
[23] L. Vives, G.S. Tuteja, A.S. Manideep, S. Jindal, N. Sidhu, R. Jindal, A. Bhatt, A novel hybrid approach of gravitational search algorithm and decision tree for twitter spammer detection, Int. J. Mod. Phys. C 33 (05) (2022) 2250060.
[24] F. Concone, G.L. Re, M. Morana, S.K. Das, Spade: multi-stage spam account detection for online social networks, IEEe Trans. Dependable Secure Comput. 20 (4) (2022) 3128–3143.
[25] M. Diqi, TwitterGAN: robust spam detection in twitter using novel generative adversarial networks, Int. J. Inf. Technol. 15 (6) (2023) 3103–3111.
[26] M. Thomas, B.B. Meshram, Chso-DNFNet: spam detection in Twitter using feature fusion and optimized deep neuro fuzzy network, Adv. Eng. Softw. 175 (2023) 103333.
[27] J. Choi, B. Jeon, C. Jeon, Scalable learning framework for detecting new types of twitter spam with misuse and anomaly detection, Sensors 24 (7) (2024) 2263.
[28] P. Manasa, A. Malik, I. Batra, Detection of twitter spam using GLoVe vocabulary features, bidirectional LSTM and convolution neural network, SN Comput. Sci. 5 (2) (2024) 206.
[29] M. Fazilat, N. Zioui, J. St-Arnaud, A novel quantum model of forward kinematics based on quaternion/Pauli gate equivalence: application to a six-jointed industrial robotic arm, Results Eng. 14 (2022) 100402.
[30] O.H.M. Ross, A review of quantum-inspired metaheuristics: going from classical computers to real quantum computers, IEEe Access. 8 (2019) 814–838.
[31] S.K. Sood, Quantum computing review: a decade of research, IEEe Trans. Eng. Manag. 71 (2023) 6662–6676.
[32] F.W. Ipeayeda, M.O. Oyediran, S.A. Ajagbe, J.O. Jooda, M.O. Adigun, Optimized gravitational search algorithm for feature fusion in a multimodal biometric system, Results Eng. 20 (2023) 101572.
[33] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, Inf. Sci. 179 (13) (2009) 2232–2248 (Ny).
[34] Hashemi A., Dowlatshahi M.B. and Nezamabadi-Pour H., 2021. Gravitational search algorithm: theory, literature review, and applications. Handbook of AI-based Metaheuristics, pp.119–150.
[35] N.M. Sabri, M. Puteh, M.R. Mahmood, An overview of Gravitational Search Algorithm utilization in optimization problems, in: Proceedings of the 2013 IEEE 3rd International Conference on System Engineering and Technology, IEEE, 2013, pp. 61–66.
[36] H.T. Chung, C.C. Tsai, K.K. Jen, Y.S. Huang, Y.C. Ferng, C.Y. Lo, T.W. Chen, K. H. Chang, A.C. Yeh, Optimization of process parameters of selective laser melted nickel-based superalloy for densification by random forest regression algorithm and response surface methodology, Results Eng. 22 (2024) 102182.
[37] J. Zhou, Y. Dai, M. Tao, M. Khandelwal, M. Zhao, Q. Li, Estimating the mean cutting force of conical picks using random forest with salp swarm algorithm, Results Eng. 17 (2023) 100892.
[38] S.J. Rigatti, Random forest, J. Insur. Med. 47 (1) (2017) 31–39.
[39] M. Schonlau, R.Y. Zou, The random forest algorithm for statistical learning, Stata J. 20 (1) (2020) 3–29.
[40] F.F. Fadoul, A.A. Hassan, R. Çağlar, Integrating autoencoder and decision tree models for enhanced energy consumption forecasting in microgrids: a meteorological data-driven approach in Djibouti, Results Eng. 24 (2024) 103033.
[41] Wang B., Zubiaga A., Liakata M. and Procter R., 2015. Making the most of tweet-inherent features for social spam detection on Twitter. arXiv preprint arXiv: 1503.07405.
[42] C. Yang, R.C. Harkreader, G. Gu, Die free or live hard? empirical evaluation and new design for fighting evolving twitter spammers, in: Proceedings of the Recent Advances in Intrusion Detection: 14th International Symposium, RAID 2011, Menlo Park, CA, USA, September 20-21, 2011, Springer Berlin Heidelberg, 2011, pp. 318–337. . Proceedings 14.
[43] K. Lee, B. Eoff, J. Caverlee, Seven months with the devils: a long-term study of content polluters on twitter, in: Proceedings of the International AAAI Conference on Web and Social Media 5, 2011, pp. 185–192.
[44] A. Kumari, Balkishan, An ant colony optimisation-based framework for the detection of suspicious content and profile from text corpus, Int. J. Intell. Syst. Technol. Appl. 20 (1) (2021) 1–24.
[45] Z. Alom, B. Carminati, E. Ferrari, A deep learning model for Twitter spam detection, Online Soc. Netw. Media 18 (2020) 100079.
[46] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, BGSA: binary gravitational search algorithm, Nat. Comput. 9 (2010) 727–745.
[47] T. Chakraborti, A. Chatterjee, A. Halder, A. Konar, Automated emotion recognition employing a novel modified binary quantum-behaved gravitational search algorithm with differential mutation, Expert. Syst. 32 (4) (2015) 522–530.
[48] Y.W. Jeong, J.B. Park, S.H. Jang, K.Y. Lee, A new quantum-inspired binary PSO: application to unit commitment problems for power systems, IEEE Trans. Power Syst. 25 (3) (2010) 1486–1495.
[49] D. Zouache, F. Nouioua, A. Moussaoui, Quantum-inspired firefly algorithm with particle swarm optimization for discrete optimization problems, Soft Comput. 20 (2016) 2781–2799.
[50] A. Kumari, Balkishan, Detection of threatening user accounts on Twitter social media database, Int. J. Intell. Eng. Inform. 7 (5) (2019) 457–489.
[51] L. Deng, C. Wu, D. Lian, Y. Wu, E. Chen, Markov-driven graph convolutional networks for social spammer detection, IEEe Trans. Knowl. Data Eng. 35 (12) (2022) 12310–12322.
[52] A. Tripathi, M. Ghosh, K.K. Bharti, Markov enhanced graph attention network for spammer detection in online social network, Knowl. Inf. Syst. (2024) 1–20.