Received 23 January 2025, accepted 3 March 2025, date of publication 5 March 2025, date of current version 14 March 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3548529

RESEARCH ARTICLE

An Ensemble-Based Auto Insurance Fraud Detection Using BQANA Hyperparameter Tuning

AFSANEH GHEYSARBEIGI, MORTEZA RAKHSHANINEJAD, MOHAMMAD FATHIAN, AND FARNAZ BARZINPOUR
School of Industrial Engineering, Iran University of Science and Technology, Tehran 13114-16846, Iran
Corresponding author: Mohammad Fathian (fathian@iust.ac.ir)

ABSTRACT The prevalence of insurance fraud in the auto industry poses significant financial challenges and undermines customer trust. Despite the application of machine learning methods to reduce these losses, current literature lacks effective tuned algorithms for detecting fraud in insurance claims. To address this gap, this study proposes an ensemble-based method with a weighted voting strategy for auto insurance fraud detection. The study uses the Binary Quantum-Based Avian Navigation Optimizer Algorithm (BQANA) to optimize the hyperparameters of Support Vector Machines (SVM), Random Forest (RF), and XGBoost classifiers, which are combined into an ensemble. To address the dataset's imbalance, random undersampling was applied to create five legitimate-to-fraudulent claim ratios: A:A, 1:1, 2:1, 4:1, and 8:1. The performance of BQANA was compared with Genetic Algorithms and Simulated Annealing for hyperparameter tuning. The results indicate that the ensemble model with BQANA-optimized hyperparameters outperforms other methods, particularly at a 1:1 ratio, achieving 99.94% Accuracy, 98.93% Precision, 100% Recall, and a 99.46% F1-score. These metrics surpass those obtained without optimization or with traditional tuning methods. This research highlights the efficacy of the BQANA algorithm in optimizing hyperparameters for classification models. By combining these optimized classifiers into an ensemble, the study significantly enhances predictive accuracy in car insurance fraud detection, offering notable improvements over conventional methods.

INDEX TERMS Insurance fraud detection, machine learning, ensemble learning, metaheuristic, hyperparameter tuning.

I. INTRODUCTION

The financial sector is progressively contending with an escalation in fraudulent activities, particularly within the domain of auto insurance [1]. The increase in the number of vehicles and corresponding insurance policies has been paralleled by a rise in fraudulent claims, resulting in significant economic burdens. Auto insurance fraud is estimated to impose an annual financial burden exceeding $40 billion, resulting in an incremental cost of approximately $400 to $700 in premiums for the average American household [2]. Significantly, research shows that while 21% to 36% of auto insurance claims may involve fraud, only about 3% receive legal scrutiny, showing that many fraudulent claims remain undetected and potentially lead to larger undisclosed financial losses [3]. The insurance industry is huge, with more than 7,000 companies and over $1 trillion in yearly premiums, making it harder to fight fraud [4]. Auto insurance fraud typically involves exaggerated claims, false claims [5], intentional injury [6], multiple claims, or other forms of misrepresenting insurance-related information [7]. However, research shows that fraudulent behavior among insurers and auto repair shops is related to each other, resulting from their mutual perceptions of each other's actions. For example, if both parties engage in fraudulent activities, both will benefit. However, each party refrains from such activities, neither can benefit and the risk of detection increases [8]. Fraudsters often believe that insurance companies and police lack the expertise to detect fraud, leading to low interest in such crimes [9]. This belief is reinforced by two factors:

The associate editor coordinating the review of this manuscript and approving it for publication was Chao Tong.

insurance companies frequently pay illegitimate claims, and insurance fraud is rarely prosecuted [4].Traditional manual verification methods are inadequate, often producing false alarms. Consequently, it is essential to employ precise and intelligent methods, such as data mining, to effectively detect and reduce fraudulent cases [10].

Various methods are employed to detect auto insurance fraud, including statistical techniques [11], machine learning (ML) [12], and deep learning (DL) [13]. Statistical methods, such as regression and hypothesis testing, are used to identify and analyze anomalies in auto insurance data. Moreover, ML methods use intelligent algorithms to detect fraudulent activities in real-time by analyzing relevant data, while DL methods leverage artificial neural networks (ANN) [14] to automatically identify complex patterns and features in large datasets. Each of these methods has its own pros and cons when applied to different types of data, making the choice of method crucial for effective fraud detection [12].

In the field of auto insurance fraud detection, three main challenges are commonly encountered: unbalanced datasets, a high number of false positives, and ML models with untuned hyperparameters. The imbalance between fraudulent and non-fraudulent samples in datasets can lead to biased models and poor fraud detection [15]. To address this, two main methods are used: oversampling and undersampling. Oversampling techniques like synthetic minority oversampling technique (SMOTE) [16] increase the number of fraudulent cases, while undersampling methods, such as random undersampling [17], reduce the number of non-fraudulent cases. Both methods aim to balance the dataset and improve the performance and reliability of predictive models [18].

Additionally, a high number of false positives can occur, meaning legitimate claims are incorrectly flagged as fraudulent. This not only leads to customer dissatisfaction but also increases operational costs for insurance companies. Advanced anomaly detection techniques and robust validation processes can help mitigate this issue by refining model accuracy [19]. Moreover, ML models with untuned hyperparameters often result in suboptimal performance. Hyperparameter tuning is essential for enhancing the accuracy and efficiency of fraud detection models [20]. Properly tuned models can better differentiate between fraudulent and non-fraudulent claims, thereby reducing both false positives and false negatives. Our motivation is to address these challenges by incorporating balancing techniques, an ensemble learning model, and a metaheuristic-based hyperparameter tuning algorithm to enhance the accuracy and robustness of auto insurance fraud detection systems. This approach aims to develop a comprehensive solution that effectively tackles the limitations of existing methods.

Using an undersampled dataset as the input for ensemble learning, which integrates multiple base classifiers, can effectively mitigate the challenges of unbalanced datasets and high false positives in auto insurance fraud detection [21]. By reducing the number of non-fraudulent cases through techniques like random undersampling, the dataset becomes more balanced, reducing bias and enhancing model reliability. Ensemble learning, which combines multiple base classifiers with distinct features, further improves detection accuracy [22]. When these base classifiers have their hyperparameters tuned with metaheuristic algorithms, their individual performances are optimized. This optimized performance, along with the ensemble method employing weighted voting between the results of these base classifiers, ensures that the most accurate predictions are prioritized. This method collectively addresses the three primary challenges unbalanced datasets, high false positives, and untuned hyperparameters resulting in a more effective fraud detection system.

This study introduces an ensemble learning framework designed for the detection of auto insurance fraud, utilizing a publicly available dataset of 15,420 car insurance claims from Kaggle, recorded between 1994 and 1996. The dataset includes 32 features, combining categorical (e.g., accident area, policy type) and numerical (e.g., age, deductible) data, offering a rich foundation for analysis. However, with only 6% of claims classified as fraudulent and 94% as legitimate, the dataset presents a significant class imbalance, posing a key challenge for fraud detection. To address this, the proposed framework incorporates three base classifiers—random forest (RF), extreme gradient boosting (XGBoost), and support vector machines (SVM)—combined into an ensemble model with a weighted voting strategy, and leverages balancing techniques to mitigate the impact of the imbalance.

The hyperparameters of each base classifier are fine-tuned using a metaheuristic optimization algorithm named binary quantum-based avian navigation optimizer algorithm (BQANA). For comparative analysis, additional hyperparameter tuning methods, including simulated annealing (SA) and genetic algorithms (GA), are employed, with their evaluation outcomes assessed. To further explore the impact of data imbalance, the study generates four subsets with varying ratios of fraudulent to legitimate claims. Each classifier is assigned a specific weight based on its performance during classification, enhancing the accuracy of fraud identification in the final ensemble voting and prediction stages. By integrating these techniques, the framework effectively addresses the challenges of unbalanced datasets, high false positives, and untuned hyperparameters, providing a robust solution for fraud detection in the auto insurance sector. The contributions of this work are summarized as follows:

- Utilization of the BQANA metaheuristic optimization algorithm for hyperparameter tuning to enhance classifier performance.
- Development of an ensemble learning framework with RF, XGBoost, and SVM, using a weighted voting

strategy to combine base classifier results, effectively reducing false positives in fraud detection.
- Application of balancing techniques, such as random undersampling, to mitigate the significant class imbalance present in the dataset.

The rest of this paper is organized into four sections. In Section II, we review previous studies on auto insurance fraud detection. Section III discusses the research method employed in this study. The obtained results and the model's effectiveness compared to other methods are presented in Section IV. Finally, in Section V, we provide practical conclusions and implications of the findings.

II. LITERATURE REVIEW

The detection of insurance fraud remains a complex and persistent challenge in contemporary society, requiring the continuous development of innovative algorithms and methods to address it effectively. As outlined in the study by [23], fraud can be defined as the misuse of professional positions for personal gain through the intentional misappropriation of organizational resources. Additionally, the study by [24] emphasizes the growing issue of financial fraud, particularly within the banking and finance sectors, where complex organizational structures and international capital flows are exploited for illegitimate gains. This manipulation undermines economic stability and violates legal frameworks and ethical standards, thus making the development of advanced fraud detection systems increasingly vital.

Given the significant economic impact of auto insurance, numerous researchers have investigated innovative techniques to detect fraud in this domain, underscoring the need for continued improvement. These include ML techniques such as Multi-Layer Perceptron (MLP), Decision Trees (DT), SVM [25], Logistic Regression (LR), ANN, and AdaBoost, Stochastic gradient descent (SGD) methods [18], Bagging [26], and other ensemble approaches. The application of ML techniques in various aspects of the finance domain has been the subject of extensive research in recent years.

The study by [12] examines the performance of several ML models, including SVM, RF, DT, Adaboost, K-Nearest Neighbor (KNN), LR, Naïve Bayes (NB), and MLP. Their findings reveal that the DT model improves the overall Accuracy of the fraud detection system. Similarly, [27] investigates the use of various ML models for insurance fraud detection, including RF, Adaboost, XGBoost, KNN, and LR. Among the models tested, the RF algorithm demonstrated the highest Accuracy and F1 score in detecting insurance fraud, thereby highlighting its strong classification performance.

Reference [28] investigated the application of various ML algorithms to detect fraudulent vehicle insurance claims. The research evaluated the performance of several models, including AdaBoost, XGBoost, NB, SVM, LR, DT, ANN, and RF, finding that AdaBoost and XGBoost outperformed the other models by achieving a classification Accuracy of 84.5%. In contrast, the LR classifiers showed poor performance, both with balanced and unbalanced data. These insights emphasize the importance of selecting appropriate algorithms to enhance fraud detection systems. Furthermore, [29] examined the use of RF, LR, and ANN for fraud detection, revealing that the RF method demonstrated superior performance, achieving an Accuracy of 98.21%, Precision of 98.08%, Recall of 100%, and F1-score of 99.03%. In addition, [30] proposed an ensemble model combining basic ML algorithms (RF, DT, XGBoost, and LR) with a meta-heuristic method called Particle Swarm Optimization (PSO). After balancing the classes using SMOTE, the proposed ensemble model improved the overall Accuracy to 99%.

Several recent studies have also focused on hyperparameter tuning techniques to optimize ML models in fraud detection contexts. Researchers have adopted both exact and meta-heuristic methods, each offering distinct advantages. Exact methods, such as Grid Search (GS), as discussed in [31], provide a systematic approach to hyperparameter optimization but can be computationally intensive. In contrast, [32] employed ML techniques, specifically employing GA to optimize hyperparameters. Their findings showed that incorporating GA results into the LR model increased Accuracy to 94%.

Moreover, [33] offers a thorough exploration of GA and XGBoost, focusing on hyperparameter optimization to enhance fraud detection systems in smart grid environments. The experimental findings showed a significant boost in model performance, raising Accuracy from 0.82 to 0.978. Similarly, [34] compares the proposed PSO method with GS, demonstrating that PSO can produce superior solutions more rapidly. Incorporating PSO results into a Deep Neural Network (DNN) model led to an Accuracy of 94.93%. Building on these advancements, the recent study by [35] introduces a PSO-XGBoost framework tailored for automobile insurance fraud detection. This framework leverages the optimization capabilities of PSO to fine-tune XGBoost hyperparameters, achieving a notable 95% accuracy. By enhancing model precision and interpretability, this approach provides actionable insights for early fraud prevention, further demonstrating the effectiveness of PSO in optimizing machine learning models for complex fraud detection challenges.

Data preprocessing and class imbalance handling techniques have been critical in developing effective fraud detection methods. Specifically, [36], [37], and [38] emphasized the use of SMOTE, Random Under-Sampling (RUS), and Random Over-Sampling (ROS) to address imbalanced datasets. Building on these efforts, [39] propose methods to address imbalanced datasets and missing values, which are common challenges in real-world insurance fraud detection. Their framework integrates data imputation techniques, such as KNN and multivariate imputation, with ensemble learning methods, including Random Forest, XGBoost, and stacking classifiers. By incorporating advanced resampling techniques like SMOTE and ADASYN, the model achieved superior accuracy and F1-scores, demonstrating its effectiveness in detecting fraudulent claims and further highlighting the importance of robust preprocessing strategies. Moreover, feature selection methods, such as GA, Firefly Algorithm Optimization (FFA), PSO, ANOVA, and Chi-2 [40], [41], are frequently employed to identify the most relevant features for robust fraud detection. Additionally, some studies have utilized unsupervised learning techniques, including K-means and C-means clustering [42], [43], to enhance detection capability.

Despite these advances, several gaps remain. First, unbalanced datasets continue to pose a significant challenge [44], often leading to skewed performance metrics and overlooked fraudulent instances [45]. Second, a high number of false positives can undermine practitioner trust and inflate investigation costs [46]. Finally, many existing ML models are not rigorously tuned, resulting in suboptimal performance when dealing with complex fraud patterns [47].

To address these challenges, this study systematically analyzes different class ratios to identify the optimal approach for balancing our dataset, thereby mitigating skewed detection outcomes. We also reduce false positives through advanced feature selection, leading to more precise fraud detection. In addition, our proposed methodology adopts an ensemble learning algorithm that employs a weighted voting strategy to improve predictive performance in the insurance fraud detection field. Furthermore, we leverage BQANA to fine-tune the hyperparameters of each base classifier incorporated into the ensemble model, thereby enhancing overall detection accuracy. By tackling unbalanced data, high false-positive rates, and untuned model parameters simultaneously, our research fills a critical gap in the literature and establishes a more robust framework for detecting fraudulent activities in the auto insurance domain.

Finally, this research synthesizes key approaches in Table 1, specifically examining the BQANA technique for hyperparameter optimization in auto insurance fraud detection. Through this review, we aim to advance the field and offer valuable insights that inform future developments.

III. MATERIAL AND METHODS

A. DATASET

The current research utilizes a comprehensive dataset obtained from an insurance company's car claim records. This dataset, available on the Kaggle platform (Link to Dataset), contains a substantial collection of 15,420 insurance claims recorded from January 1994 to December 1996 [49]. For accessibility and reproducibility, details on data availability are provided in the Data and Code Availability section at the end of the paper.

The dataset contains 32 features, in addition to a target feature, as detailed in Table 2. Each sample in the dataset is shown by a binary target feature, which is essential for classifying the claims into fraudulent or non-fraudulent categories. The dataset is identified by its variety, and contains 25 categorical and 8 numerical features, providing a rich foundation for analysis.

The employed dataset shows a clear imbalance in the distribution of fraudulent cases, which account for only 6% (923 claims) of the total, in contrast to the 94% (14,497 claims) that are non-fraudulent. This data imbalance shows a fundamental challenge in fraud detection and is a significant factor to consider when selecting appropriate modeling and evaluation techniques.

In preparation for the analysis, the dataset underwent a partitioning process to facilitate the model's training, testing, and validation phases. Specifically, the data was randomly divided into three subsets: 60% was allocated for training the model, enabling it to learn and adapt to the patterns within the data; 20% was reserved for testing, providing an evaluation of the model's predictive performance on unseen data; and the remaining 20% was allocated for validation purposes, allowing an additional layer of evaluation to fine-tune the model's hyperparameters. This partitioning strategy is essential for the valid evaluation of fraud detection models. All codes were implemented in Python 3.6.15. The computations were performed on a system equipped with an Intel Core i3-3220 processor and 4 GB DDR3 RAM.

B. PREPROCESSING

The preprocessing step commenced with a meticulous examination of the initial dataset to detect and eliminate redundant features that could hinder the model's learning efficiency and jeopardize result Accuracy. The features named "Policy Number" and "Age" were removed from the dataset, as "Policy Number" was used solely as distinct identifiers for each claim, and "Age" duplicated the information already provided by the "Age of policyholder" feature [4]. Upon a detailed review of prior studies outlined in Table 3, it was observed that 10 features have no impact on the model's accuracy and efficiency. Consequently, these irrelevant features were methodically eliminated from the dataset to enhance the speed and Accuracy of fraud detection in auto insurance. The final set of features that we removed from the dataset in our study are shown in the last row of Table 3. Moreover, to meet the algorithmic requirement for numerical input, the classification features underwent a transformation process assigning each category a unique integer value, streamlining computational processing and analysis.

Additionally, to meet the algorithmic requirement for numerical input, the categorical features underwent a transformation process, where each category was assigned a unique integer value. This transformation step reduces the computational processing and eases the analysis of the data.

Another significant issue that researchers are involved with is the challenge of data imbalance [15]. Previous scientific literature has documented various strategies, including both undersampling and oversampling techniques, to address the challenge of data imbalance in the context of fraud detection. Previous research has explored the effectiveness of methods such as SMOTE [58], Adaptive Synthetic

TABLE 1. Studies reviewed in the field of fraud detection.

Article	Aim																									Dataset	
	Prediction													Hyperparameter Tuning				Preprocessing									
	Machine Learning																										
	Supervised												Unsupervised					Feature Selection				Balancing					
	LR	MLP	SVM	RF	DT	NB	KNN	Bagging	XGBoost	CatBoost	AdaBoost	ANN	K-Means	GA	PSO	BQANA	GS	GA	Boruta	PSO	ANOVA/Chi2	SMOTE	ROS	RUS	This Study	Other
[42]													*													*
[48]		*	*		*																	*			*	
[49]	*	*	*	*	*	*					*	*													*	
[50]			*				*					*										*			*	
[51]		*	*	*	*																	*			*	
[18]	*		*	*	*	*	*					*										*			*	
[25]		*	*		*																				*	
[17]	*		*	*		*	*																	*		*
[23]						*						*												*	*	
[26]	*		*		*	*		*																		*
[32]	*		*	*	*							*		*			*									*
[38]	*		*	*	*	*	*		*		*	*											*	*		*
[52]	*		*	*			*		*		*											*				*
[20]	*			*	*				*			*								*	*	*				*
[41]	*			*	*	*						*						*								*
[12]	*		*	*	*	*	*				*	*														*
[4]	*		*			*													*							*
[40]	*		*	*					*			*								*	*	*				*
[29]	*			*								*														*
[27]	*		*	*			*		*		*											*				*
[28]	*			*			*		*																*	
[53]			*	*	*	*					*	*														*
[54]	*		*		*		*					*										*				*
[55]	*			*	*	*	*															*	*	*		*
[56]				*	*				*	*															*	
[35]	*		*	*		*			*			*			*								*	*	*	
[39]	*			*		*	*	*	*													*	*	*	*	*
This Study			*	*					*							*								*	*	

Sampling Approach (ADASYN) [59], and Tabular Generative Adversarial Networks (TGAN) [60] in decreasing the imbalance within the dataset. These techniques have been explored and evaluated for their ability to effectively handle the disproportionate representation of fraudulent and non-fraudulent cases, which is a fundamental issue in fraud detection.

In alignment with the findings of previous study [27], we opted for the RUS to rectify the balance in our dataset. Our objective was to systematically assess which undersampling ratio produces the most effective results during the training phase and, subsequently, leads to the best performance in prediction. Accordingly, the dataset was divided into training, testing, and validation subsets. Specifically, the entire dataset was split randomly, with 60% of the samples constituting the training set, 20% allocated to testing, and the remaining 20% reserved for validation.

Table 4 illustrates the five different undersampling ratios we applied to the training set. In Ratio A:A (no undersampling), all training samples remained intact, resulting in 8,699 normal and 553 fraudulent cases. By contrast, the 1:1 ratio reduces the number of normal samples to 553, creating a perfectly balanced subset. The 2:1 ratio allows for 1,106 normal and 553 fraudulent samples, while the 4:1 and 8:1 ratios further increase the number of normal samples to 2,212 and 4,424, respectively, against the same 553 fraud cases. Our intent in gradually modifying the class distribution is to identify the ratio that optimizes the detection of fraudulent activities while minimizing misclassification errors. Meanwhile, the testing and validation sets, each comprising 2,899 normal and 185 fraud samples, remain unchanged, ensuring an unbiased evaluation of the trained model.

Another significant aspect in the field of data preprocessing is the identification of important features. Based on previous research, various feature selection techniques have been employed, as documented in Table 5. These include Boruta's algorithm [4] and meta-heuristic methods such as Ant Colony Optimization (ACO), PSO, and GA [57].

The results presented in Table 3 and Table 5 shows that certain features, such as "Rep Number", "Deductible", and "Policy Type", have been recognized as significant factors in some research studies while being deemed less significant in others. When these features are recognized as important, the model shows increased Accuracy levels, prompting the researchers to categorize these three features as significant.

Consequently, in the current study, we have utilized a set of 23 features after removing 10 less significant features identified through the aforementioned feature selection techniques. This feature engineering process aims to increase the speed and Accuracy of the fraud detection model by focusing on the most relevant features.

C. MODELING

After the data preprocessing steps on the primary dataset are completed, the focus is on developing a fraud detection model. Through a comprehensive review of the existing literature in the field of fraud detection, the researchers

TABLE 2. Description of the dataset.

Row	Feature	Description	Type
1	Month	The month of the accident	Categorical
2	Week of month	Week of the month when the accident occurred	Numerical
3	Day of week	The day of the week when the accident occurred	Categorical
4	Month claimed	The month the claim is made	Categorical
5	Week of month claimed	The week of the month when the claim was made	Numerical
6	Day of week claimed	The day of the week when the claim occurred	Categorical
7	Year	1994, 1995, and 1996	Numerical
8	Make	What company did the car manufacturing belong to? (19 companies)	Categorical
9	Accident area	Urban or rural accident	Categorical
10	Gender	Being Female or male	Categorical
11	Marital status	Being Married, divorced, single, or widowed	Categorical
12	Age	How old is the policyholder?	Numerical
13	Fault	Defect related to third party or policyholder	Categorical
14	Policy type	What is the type of policy? (1–9)	Categorical
15	Vehicle category	Being a Utility, sport, or sedan	Categorical
16	Vehicle price	What is the price of the car? (6 groups)	Categorical
17	Rep. number	ID of the representative who reviewed the claim (16 IDs)	Numerical
18	Deductible	The amount to be deducted before the damages are paid	Numerical
19	Driver rating	What is the driving skill of the driver? (4 groups)	Numerical
20	Days: policy accident	The days left of the policy when the incident occurred	Numerical
21	Days: policy claim	Days remaining on the policy when the claim is submitted	Categorical
22	Past No. of Claims	The number of claims that have occurred in the past	Categorical
23	Age of vehicle	How old is the vehicle? (8 groups)	Categorical
24	Age of policyholder	How old is the policyholder? (9 groups)	Categorical
25	Policy report filed	To be yes or no	Categorical
26	Witness presented	To be yes or no	Categorical
27	Agent type	Being External or internal	Categorical
28	Number of supplements	How many supplements are there?	Categorical
29	Address change claim	The number of times that address has changed	Categorical
30	Number of cars	How many cars are there?	Categorical
31	Base policy (BP)	Collision, liability, or all risks are included	Categorical
32	Policy Number	The policy number has been allocated to the client	Numerical
33	Class	Fraudulent or not Fraudulent	Categorical

TABLE 3. Removed features.

Row	Ref.	Removed Features
1	[4]	Month, Week of month, Day of week, Make, Day of week claimed, Month claimed, Week of month claimed, Age, Rep Number, Deductible, Past No. of Claims, Address change claim, No of Cars, and Policy Number
2	[49]	Policy Number, Rep Number, Month, Week of month, Day of week, Day of week claimed, Week of month claimed, and Month claimed
3	[57]	Month claimed, Day of week claimed, and PolicyType
4	[27]	Age, Policy Number, and Rep Number
5	**Removed Features in This Study**	Month, Week of month, Day of week, Make, Day of week claimed, Month claimed, Week of month claimed, Age, No of Cars, and Policy Number

have explored various modeling techniques, including LR, DT, SVM, NB [49], CatBoost, XGBoost, RF [56], KNN, and AdaBoost [27]. Based on insights from the literature, we selected SVM, RF, and XGBoost for this study, as shown in Table 6. To enhance their performance, we optimized the hyperparameters of these classifiers using the BQANA

TABLE 4. Undersampling training set.

Dataset	Ratio	Normal	Fraud
Training	A:A	8699	553
	1:1	553	553
	2:1	1106	553
	4:1	2212	553
	8:1	4424	553
Testing	-	2899	185
Validation	-	2899	185

method. Once the optimal hyperparameters were determined, the models were trained using five data imbalance ratios (A:A, 1:1, 2:1, 4:1, and 8:1) as described in Table 4.

Subsequently, we constructed an ensemble model where each of the three base classifiers was assigned a weight reflecting its relative importance. Weights were calculated using 10-fold cross-validation results. Each classifier within the ensemble algorithm was allocated a weight between 0 and 1 based on its performance on the validation dataset, measured by Accuracy, Precision, Recall, and F1-score.

TABLE 5. Methods used to select important features.

Techniques Used	Ref.	Selected Features
Boruta	[4]	Age of policyholder, Age of Vehicle, Base Policy, Fault, Marital Status, Vehicle Category, Sex, Days Policy Claim, and Policy Type
PSO, ACO, GA	[57]	Fault, RepNumber, PastNumberofClaims, BasePolicy, Deductible, and AgeofPolicyHolder
This Study	-	Age of policyholder, Age of Vehicle, Base Policy, Fault, Marital Status, Vehicle Category, Gender, Address change claim, Days Policy Claim, Policy Type, RepNumber, Past No. of Claims, Deductible, Accident area, Vehicle price, Driver rating, Days policy accident, Policy report field, Year, Witness presented, Agent type, Number of supplements, and Class

These weights were then used in a weighted voting technique to determine the ensemble model's results on the test set.

Given the five training datasets with varying ratios of fraudulent to normal samples, we developed five distinct ensemble models, denoted as $Eclf_1$, $Eclf_2$, …, $Eclf_5$. The base classifiers in the first ensemble algorithm are denoted as $clf_1, clf_2 \ldots, clf_3$. For example, Equation (1) and Equation (2) determines the weight of the first base classifier within the first ensemble algorithm. The result from Equation (1) is used in Equation (2) to calculate the weights of each classifier within the ensemble model. In each ensemble, the sum of the base classifiers' weights is constrained to equal one. The notations used in Equations (1) and (3) are as follows: Accuracy (A), Precision (P), Recall (R), and F1-Score (F1).

$$D_{clf_1} = \frac{A_{clf_1} \times P_{clf_1} \times R_{clf_1} \times F1_{clf_1}}{\sum_{i=2}^{3} \left(A_{clf_i} \times P_{clf_i} \times R_{clf_i} \times F1_{clf_i}\right)} \tag{1}$$

$$W_{clf_1} = \frac{D_{clf_1}}{\sum_{i=1}^{3} D_{clf_i}} \tag{2}$$

Then, as below, for all the ensembles, calculate the weights of the classifiers inside them:

$$Eclf_1 - \text{Classifiers Weight} = \{W_{clf_1}, W_{clf_2}, W_{clf_3}\}$$
$$Eclf_2 - \text{Classifiers Weight} = \{W_{clf_4}, W_{clf_5}, W_{clf_6}\}$$
$$Eclf_3 - \text{Classifiers Weight} = \{W_{clf_7}, W_{clf_8}, W_{clf_9}\}$$
$$Eclf_4 - \text{Classifiers Weight} = \{W_{clf_{10}}, W_{clf_{11}}, W_{clf_{12}}\}$$
$$Eclf_5 - \text{Classifiers Weight} = \{W_{clf_{13}}, W_{clf_{14}}, W_{clf_{15}}\}$$

The voting strategy used the calculated weights to determine each ensemble algorithm's outcome. For example, the weight of the first ensemble was computed using Equations (3) and (4), with the result from Equation (3) serving as input for Equation (4).

$$D_{Eclf_1} = \frac{A_{Eclf_1} \times P_{Eclf_1} \times R_{Eclf_1} \times F1_{Eclf_1}}{\sum_{i=2}^{5} \left(A_{Eclf_i} \times P_{Eclf_i} \times R_{Eclf_i} \times F1_{Eclf_i}\right)} \tag{3}$$

$$W_{Eclf_1} = \frac{D_{Eclf_1}}{\sum_{i=1}^{5} D_{Eclf_i}} \tag{4}$$

Afterward, we calculated the weight for all 5 ensembles:

$$\text{Ensembles Weight} = \{W_{Eclf_1}, W_{Eclf_2}, W_{Eclf_3}, W_{Eclf_4}, W_{Eclf_5}\}$$

Ultimately, the ensemble algorithm with the maximum calculated weight was selected for evaluation based on the criteria of Accuracy, Precision, Recall, and F1-score using the test dataset. The flowchart of the proposed method is illustrated in Figure 1.

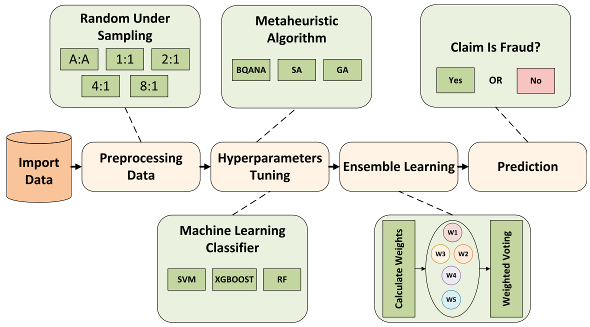

FIGURE 1. The flowchart of the proposed method.

In the field of ML, the definition of hyperparameters is significant for any algorithm prior to starting the training step. The optimization of these hyperparameters plays a significant role in increasing the model's performance on the test dataset. Each algorithm is determined by a unique set of predefined hyperparameters. Therefore, we can describe hyperparameter tuning as an endeavor to determine the optimal values of the hyperparameters for the learning algorithm that submits the best model performance. Previous studies have explored several types of hyperparameter tuning methods, such as RS, GS [32], and meta-heuristic algorithms. Through an examination of prior research in the domain of fraud detection, various methods such as GA, Differential Evolution (DE), Artificial Bee Colony (ABC), Grey Wolf Optimizer (GWO), PSO, Teaching-Learning-Based Optimization (TLBO), and GS [34] has been used. In this study, we use BQANA, which has not been studied in past studies in the field of car insurance fraud detection.

First in the study [61] Quantum-based avian navigation optimizer algorithm (QANA) was introduced. Then in the study [62] a binary version of QANA named BQANA was introduced. In this paper, using BQANA, the superior features of several high-dimensional datasets were identified to perform classification operations. Then in the study [63] and [64] QANA was used to solve engineering problems. Although QANA is powerful and used in versatile fields of engineering studies, this algorithm and its binary versions are not used for hyperparameter tuning in any of the published papers. We aim to optimize classifiers including XGBoost, RF, and SVM across the ensemble using the BQANA optimization algorithm.

QANA is a population-based meta-heuristic algorithm, that draws inspiration from the navigational patterns of migratory birds during extended aerial journeys. This algorithm, QANA, is designed with a multi-flock framework and quantum-driven navigation, incorporating two mutation techniques and a qubit-crossover operator to enhance efficient exploration of the search domain. At first, the initial step involves dividing the migratory bird population into multiple flocks in a random manner. Subsequently, the algorithm imitates the flight formation of migratory birds to share acquired information among search agents through the utilization of a V-echelon communication structure. Assuming V represents a collection of n individuals within the flock f_q, comprising a header (H) and two subgroups known as right-line (R) and left-line (L) arranged in a V-shaped configuration.

The flocks employ a quantum-based navigation method for search space exploration, incorporating a Success-based Population Distribution (SPD) strategy, two mutation methods known as "DE/quantum/I" and "DE/quantum/II," and a qubit-crossover operator. Each flock dynamically switches between these mutation techniques, with f_m representing the flocks utilizing M_m in iteration t (as shown in Equation (5)). The variable τ_{ij} is set to 1 if M_m enhances a_j of the i-th flock in the set f_m; otherwise, it is assigned a value of 0.

$$SR_m(t) = \left(\left(\sum_{i \in f_m} \frac{\sum_{j=1}^{n} \tau_{ij}}{n} \right) \Big/ |f_m| \right) \times 100 \tag{5}$$

The quantum mutation strategies are defined by Equations (6) and (7). Here, $x_i(t)$ represents the current position of search agent a_i in the iteration t, $x_{V_{echelon}}(t)$ denotes the position of the subsequent search agent after a_i, and $x_{best}(t)$ indicates the best search agent's location. Random selections from short-term memory (STM) and long-term memory (LTM) are denoted by $x_{j \in STM}(t)$ and $x_{j \in LTM}(t)$ respectively. Equation (8) is utilized to compute the trial vector $v_H(t+1)$ as the leader in the V-echelon structure, where L and U represent the lower and upper boundaries of the search space. Additionally, S_i denotes the quantum orientation of the bird a_i, and incorporates a parameter adaptation mechanism based on a historical record of successful parameters.

$$\begin{aligned} v_i(t+1) = {} & x_{best}(t) + S_i(t) \times \left(x_{V_{echelon}}(t) - x_{j \in LTM}(t)\right) \\ & + S_i(t) \times \left(x_{V_{echelon}}(t) - x_{best}(t)\right) \\ & + S_i(t) \times \left(x_{j \in LTM}(t) - x_{j \in STM}(t)\right) \end{aligned} \tag{6}$$

$$\begin{aligned} v_i(t+1) = {} & S_i(t) \times \left(x_{best}(t) - x_{V_{echelon}}(t)\right) \\ & + S_i(t) \times \left(x_i(t) - x_{j \in LTM}(t) - x_{j \in STM}(t)\right) \end{aligned} \tag{7}$$

$$v_H(t+1) = S_i(t) \times x_{best} + (L + (U - L) \times rand(0,1)) \tag{8}$$

To generate the trial vector $u_i(t+1)$, the mutant vector $v_i(t+1)$ is combined with its parent $x_i(t)$ using Equation (9), with $|\varphi_i\rangle_d$ representing the qubit-crossover probability for the d-th dimension. Each iteration involves the calculation of a qubit-crossover $|\varphi_i\rangle_d$ for each dimension of the trial vector $u_i(t+1)$ through Equation (10), where the parameter $|\varphi_R\rangle_d$ is a random integer acting as a coefficient for adjusting the length of the vector $|\varphi_i\rangle_d$ within the Bloch sphere [61].

$$u_{id}(t+1) = \begin{cases} x_{id}(t+1), & |\varphi_i|_d < rand \\ v_{id}(t+1), & |\varphi_i|_d \geq rand \end{cases} \tag{9}$$

$$\begin{aligned} |\varphi_i\rangle_d &= |\varphi_R\rangle_d \times \left(\cos\left(\frac{\theta}{2}\right) |0\rangle + e^{i\varphi} \sin\left(\frac{\theta}{2}\right) |1\rangle \right), \\ \theta, \varphi &= rand \times \frac{\pi}{2} \end{aligned} \tag{10}$$

As per the findings from a prior study [61], QANA demonstrates superior performance compared to other established optimizers across diverse continuous search space benchmark assessments. When compared to its rivals, QANA surpasses them in terms of both exploration and exploitation capabilities. Consequently, the foundational components of the conventional QANA are adapted to formulate its binary counterpart. During the binary QANA's formulation, the initial solutions are generated at random within the interval [0, 1]. Following this initialization, the iterative procedure is carried out until the predefined termination criterion, typically the maximum number of iterations is met. Based on this study [62], using the threshold method for binary conversion of continuous solutions yields significantly improved outcomes compared to transfer functions like S-shaped, V-shaped, U-shaped, and Z-shaped.

The performance of SVM, RF, and XGBoost models within an ensemble model relies on the precise selection of optimal hyperparameters, so in this study, our objective is to employ BQANA to select optimal hyperparameters set for each base classifiers inside the ensemble. A list of possible parameters associated with these three models is outlined in Table 6.

The SVM model includes c, kernel, degree, gamma, shrinking, and tol hyperparameters [34]. In the case of the RF model, fine-tuning involved hyperparameters such as solver type, n_estimators, criterion, max_depth, min_samples_split, min_samples_leaf, max_features, bootstrap, and max_features [34]. The tuning of the XGBoost model centered on hyperparameters such as learning rate, n_estimators, min_weight_fraction_leaf, max_depth, min_impurity_decrease, colsample_bytree, reg_alpha, reg_lambda, and subsample [33]. The range of the relevant hyperparameters was identified through a review of the documentation available on the Scikit-learn (https://scikit-learn.org) platform as well as related scientific literature.

D. FITNESS EVALUATION

To optimize the hyperparameters of the XGBoost, SVM, and RF classifiers, we employed the BQANA algorithm. This algorithm at first generates proposed solutions for the model hyperparameters in the first iteration. Each solution length is equal to a number of hyperparameters of classifiers. Subsequently, the fitness function calculates the mean of Accuracy, Precision, Recall, and F1-score for each of these solutions and selects the best solutions from the first iteration.

TABLE 6. XGBoost, Svm, Rf hyperparameters to be optimized by BQANA, SA, and GA.

Model	Hyperparameters	Range of Value
SVM	C	1-100
	Kernel	linear, poly, rbf, and sigmoid
	Degree	1-5
	Gamma	0.05-1
	Shrinking	true, false
	Tol	10^{-5} - 10^{-1}
RF	n_estimators	100-300
	Criterion	gini, entropy
	max_depth	10-200
	min_samples_split	10-100
	min_samples_leaf	1-10
	max_features	0-1
	Bootstrap	true, false
XGBoost	learning_rate	0.01-0.3
	n_estimators	50-1000
	min_weight_fraction_leaf	1-10
	max_depth	3-10
	min_impurity_decrease	0-5
	colsample_bytree	0.01-1
	reg_alpha	0-1
	Subsample	0.5-1
	reg_lambda	1-100

However, we repeated this process to select the optimal solutions for a total of 10 iterations. We employed four classification metrics to assess the predictive abilities of the generated solutions by BQANA: Accuracy, Precision, Recall, and F1-score. Following an fitness evaluation of the generated solutions, we rank them based on the mean performance across the four metrics.

The accuracy metric is defined as the ratio of the number of correct predictions to the total number of predictions [65], relies on True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) to determine the model's accuracy, as outlined below.

$$\text{Accuracy} = \frac{TN + TP}{TN + FP + FN + TP} \tag{11}$$

The precision formula signifies the ratio of positive predictions that were accurately identified.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{12}$$

The recall formula, provided below, indicates the proportion of actual positive cases that were correctly recognized.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{13}$$

The balance between Precision and Recall can lead to a situation where a model performs well on one metric but poorly on the other. The F1-score tackles this challenge by taking both metrics into account simultaneously [4]. It is formulated as follows:

$$F1\text{-Score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Sensitivity}}} \tag{14}$$

E. COMPARISON

In this study, we employed the BQANA meta-heuristic algorithm to optimize the hyperparameters of the SVM, RF, and XGBoost models, as well as their ensemble, and then compared the results against two other metaheuristic methods (SA and GA).

In Section III, we explained the mechanism of BQANA for hyperparameter tuning. In comparison, SA begins with a defined initial temperature and a cooling coefficient, leading to a gradual reduction in temperature with each iteration. At each iteration, SA evaluates a new set of randomly selected hyperparameters. If this new set results in a better fitness function value, it is used for subsequent iterations. As the algorithm progresses, the likelihood of accepting suboptimal solutions decreases. The fitness function within SA assesses the efficacy of a solution in relation to the defined problem [66].

GA, on the other hand, starts by creating an initial population of potential solutions, typically represented as binary strings, though other data structures may also be used. An objective function evaluates each individual's performance within the population. Higher-quality solutions are more likely to be selected for reproduction. During the crossover phase, the genetic information of two parent solutions merges to produce new offspring, aiming to create an improved solution that incorporates beneficial features of the parents. The mutation step introduces random changes in the offspring's genetic composition to maintain genetic diversity and prevent the algorithm from becoming too uniform or trapped in local optima [67].

The hyperparameter values used for BQANA, SA, and GA are presented in Table 7. These parameters were selected based on guidelines from previous studies, specifically [61] for BQANA, [66] for GA, and [67] for SA, which detail standard parameter settings and best practices for each metaheuristic method. Similar to BQANA, SA and GA also employ an iterative evaluation of the fitness function to identify an optimal solution. In our experiments, we ran each of these algorithms (BQANA, SA, and GA) ten times, with each run consisting of 100 iterations. The best results obtained from these ten runs for each algorithm are illustrated in Figures 2, 3, and 4. For the comparative evaluation, we used the "carclaims.txt" insurance dataset, partitioning it into three subsets for training (60%), testing (20%), and validation (20%). We began by fine-tuning the hyperparameters of the SVM, RF, and XGBoost classifiers using BQANA for each of the specified data ratios (A:A, 1:1, 2:1, 4:1, and 8:1). After determining the optimal hyperparameters, we trained each classifier separately and then combined them into an ensemble learning model, with the goal of further improving fraud detection performance.

We first evaluated the performance of each base classifier and the ensemble model across the five data ratios (A:A, 1:1, 2:1, 4:1, and 8:1) before hyperparameter tuning. Next, we employed the meta-heuristic algorithms BQANA, SA,

TABLE 7. Hyperparameters and values used in three metaheuristic optimization algorithms BQANA, GA, and SA.

Model	Hyperparameters	Value	Ref.
BQANA	Iteration	100	[61]
	Number of search agents	100	
	Number of flocks	2	
	LTM size	2	
	STM size	10	
	Number of runs	30	
GA	Iteration	100	[66]
	Crossover probability	1.0, 0.6	
	Population size	100	
	Mutation rate	0.1	
	Penalty coefficient	0.5	
	In rank-based selection	0.25	
	Number of runs	30	
SA	Initial Temperature (T_0)	1000	[67]
	Cooling Rate	$\alpha = 0.9$	
	Maximum number of iterations	100	
	Neighborhood Function	Swap	

and GA to optimize the hyperparameters of these classifiers, and re-evaluated their performance using the same data ratios.

IV. RESULTS AND DISCUSSION

In this section, we present the outcomes of applying three metaheuristic algorithms—BQANA, SA, and GA—to fine-tune the hyperparameters of SVM, RF, and XGBoost models across five dataset ratios (A:A, 1:1, 2:1, 4:1, and 8:1). After determining the optimal hyperparameters for each model-dataset combination, we leveraged the tuned SVM, RF, and XGBoost classifiers as base learners in an ensemble approach, aiming to enhance overall performance. Tables 8, 9, and 10 summarize the best hyperparameter values identified for SVM, RF, and XGBoost, respectively, categorized by the metaheuristic optimization algorithms (BQANA, SA, and GA) and the training set ratios used during the tuning process.

The effectiveness and performance of the metaheuristic algorithms in tuning the hyperparameters of the ML models are depicted in Figures 2, 3, and 4. The iterative process of the BQANA algorithm, as shown in Figure 2, begins by evaluating the fitness value of the initial solution and refining this solution over 100 iterations to identify the optimal hyperparameter configuration. Similarly, Figures 3 and 4 illustrate the fitness values at each iteration for the SA and GA algorithms, respectively. These figures provide a clear visualization of how each algorithm converges toward improved solutions over the course of 100 iterations, demonstrating their effectiveness in optimizing ML model performance through hyperparameter tuning.

We first assessed the performance of each base classifier and the ensemble model across the five data ratios prior to hyperparameter tuning, as shown in Table 11. Subsequently, we applied the metaheuristic algorithms BQANA, SA, and GA to optimize the hyperparameters of these classifiers.

TABLE 8. Hyperparameter values for the SVM model tuned using BQANA, GA, and SA across different training set ratios.

Model	Train Set Ratio	Hyperparameters	BQANA	GA	SA
SVM	A:A	C	10	20	15
		Kernel	Rbf	Linear	Poly
		Degree	3	2	4
		Gamma	0.1	0.2	0.15
		Shrinking	TRUE	TRUE	FALSE
		Tol	1e-03	1e-04	1e-05
	1:1	C	12	25	18
		Kernel	Sigmoid	Poly	Sigmoid
		Degree	4	1	5
		Gamma	0.08	0.05	0.2
		Shrinking	FALSE	FALSE	TRUE
		Tol	1e-02	1e-02	1e-02
	2:1	C	30	15	30
		Kernel	Linear	Rbf	Linear
		Degree	2	4	2
		Gamma	0.25	0.25	0.3
		Shrinking	TRUE	TRUE	TRUE
		Tol	1e-04	5e-03	1e-03
	4:1	C	50	30	22
		Kernel	Poly	Linear	Rbf
		Degree	1	5	3
		Gamma	0.3	0.2	0.1
		Shrinking	TRUE	FALSE	FALSE
		Tol	1e-05	1e-03	5e-04
	8:1	C	70	35	25
		Kernel	Rbf	Poly	Rbf
		Degree	5	2	3
		Gamma	0.5	0.15	0.05
		Shrinking	FALSE	TRUE	TRUE
		Tol	1e-03	1e-02	1e-01

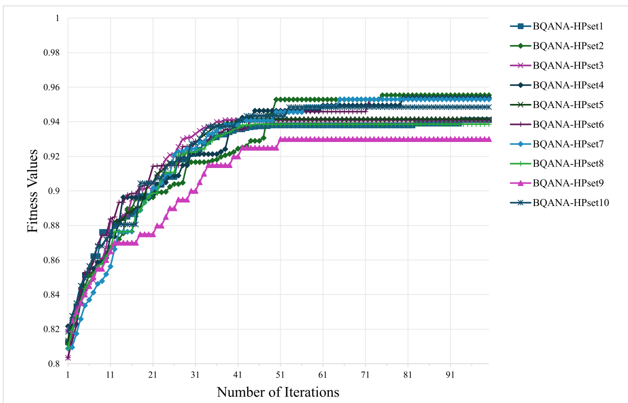

FIGURE 2. BQANA hyperparameter selection optimization plot.

Following this tuning process, we re-evaluated their performance using the same dataset ratios. The comparative results, highlighting the impact of hyperparameter optimization on each classifier and the ensemble model, are presented in Table 12.

TABLE 9. Hyperparameter values for the RF model tuned using BQANA, GA, and SA across different training set ratios.

Model	Train Set Ratio	Hyperparameters	BQANA	GA	SA
RF	A:A	n_estimators	250	200	300
		Criterion	Gini	Entropy	Gini
		max_depth	50	100	150
		min_samples_split	20	30	40
		min_samples_leaf	6	5	4
		max_features	0.8	0.6	0.7
		Bootstrap	TRUE	FALSE	TRUE
	1:1	n_estimators	220	180	270
		Criterion	Entropy	Gini	Entropy
		max_depth	60	90	130
		min_samples_split	25	35	45
		min_samples_leaf	4	7	3
		max_features	0.7	0.8	0.75
		Bootstrap	TRUE	TRUE	FALSE
	2:1	n_estimators	230	190	280
		Criterion	Gini	Gini	Entropy
		max_depth	70	80	140
		min_samples_split	30	40	35
		min_samples_leaf	6	5	4
		max_features	0.75	0.7	0.65
		Bootstrap	FALSE	TRUE	TRUE
	4:1	n_estimators	240	210	260
		Criterion	Entropy	Gini	Gini
		max_depth	80	110	120
		min_samples_split	35	50	30
		min_samples_leaf	3	6	5
		max_features	0.8	0.75	0.7
		Bootstrap	TRUE	FALSE	FALSE
	8:1	n_estimators	260	220	290
		Criterion	Gini	Entropy	Entropy
		max_depth	90	120	160
		min_samples_split	40	45	25
		min_samples_leaf	5	4	6
		max_features	0.85	0.65	0.8
		Bootstrap	TRUE	TRUE	TRUE

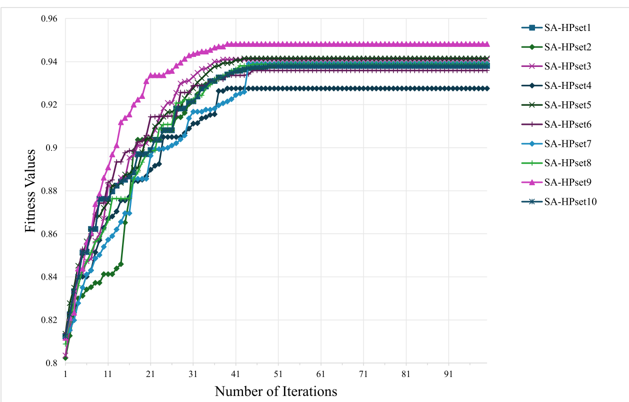

FIGURE 3. SA hyperparameter selection optimization plot.

Tables 11 and 12 summarize the evaluation metrics, including Accuracy, Precision, Recall, and F1-score, before and after hyperparameter tuning. These results demonstrate the effectiveness of the metaheuristic algorithms employed in this study—BQANA, SA, and GA—in optimizing the hyperparameters of the ML models. The improvement in performance across the classifiers and ensemble model highlights the significance of these methods in accurately distinguishing between fraudulent and legitimate insurance claims.

TABLE 10. Hyperparameter values for the XGBoost model tuned using BQANA, GA, and SA across different training set ratios.

Model	Train Set Ratio	Hyperparameters	BQANA	GA	SA
XGBoost	A:A	learning_rate	0.2	0.15	0.1
		n_estimators	500	600	700
		min_weight_fraction_leaf	5	3	4
		max_depth	7	8	6
		min_impurity_decrease	1	2	1
		colsample_bytree	0.7	0.6	0.5
		reg_alpha	0.3	0.4	0.2
		Subsample	0.8	0.7	0.6
		reg_lambda	50	45	55
	1:1	learning_rate	0.1	0.2	0.25
		n_estimators	600	500	750
		min_weight_fraction_leaf	4	3	6
		max_depth	6	5	7
		min_impurity_decrease	2	1	2
		colsample_bytree	0.6	0.7	0.8
		reg_alpha	0.2	0.3	0.4
		Subsample	0.7	0.8	0.9
		reg_lambda	45	50	60
	2:1	learning_rate	0.15	0.1	0.2
		n_estimators	550	650	600
		min_weight_fraction_leaf	3	5	4
		max_depth	8	6	7
		min_impurity_decrease	2	1	3
		colsample_bytree	0.7	0.6	0.5
		reg_alpha	0.3	0.2	0.3
		Subsample	0.8	0.7	0.9
		reg_lambda	50	60	45
	4:1	learning_rate	0.2	0.25	0.1
		n_estimators	700	600	800
		min_weight_fraction_leaf	4	3	5
		max_depth	7	8	6
		min_impurity_decrease	1	2	1
		colsample_bytree	0.6	0.7	0.8
		reg_alpha	0.4	0.3	0.2
		Subsample	0.9	0.8	0.7
		reg_lambda	55	50	45
	8:1	learning_rate	0.25	0.2	0.1
		n_estimators	750	650	700
		min_weight_fraction_leaf	3	5	4
		max_depth	6	7	8
		min_impurity_decrease	1	2	1
		colsample_bytree	0.8	0.6	0.7
		reg_alpha	0.2	0.3	0.4
		Subsample	0.7	0.8	0.9
		reg_lambda	60	55	50

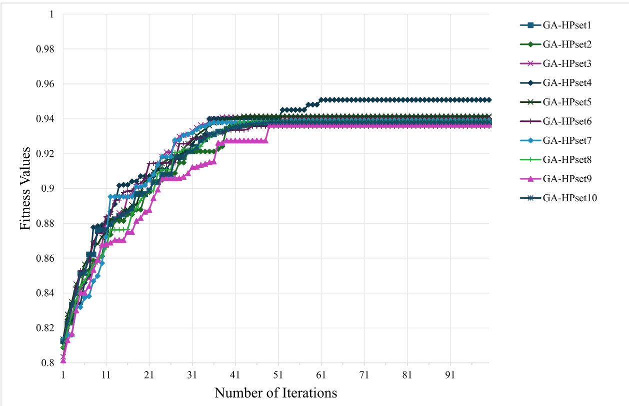

FIGURE 4. GA hyperparameter selection optimization plot.

TABLE 11. Evaluation metrics of ML models before hyperparameter tuning.

Model	Train Set Ratio	Accuracy	Precision	Recall	F1-score
SVM	A:A	99.64	95.31	98.92	97.08
	1:1	99.71	95.83	99.46	97.61
	2:1	99.58	94.79	98.38	96.55
	4:1	99.51	94.27	97.84	96.02
	8:1	99.45	93.75	97.30	95.49
RF	A:A	99.68	95.81	98.92	97.34
	1:1	99.74	96.34	99.46	97.87
	2:1	99.64	95.31	98.92	97.08
	4:1	99.58	94.79	98.38	96.55
	8:1	99.51	94.27	97.84	96.02
XGBoost	A:A	99.64	95.31	98.92	97.08
	1:1	99.71	95.83	99.46	97.61
	2:1	99.64	95.31	98.92	97.08
	4:1	99.58	94.79	98.38	96.55
	8:1	99.51	94.27	97.84	96.02
Ensemble	A:A	99.74	96.34	99.46	98.13
	1:1	99.77	96.84	99.46	98.13
	2:1	99.71	95.83	99.46	97.61
	4:1	99.64	95.31	98.92	97.08
	8:1	99.58	94.79	98.38	96.55

In terms of Accuracy, the ensemble method with BQANA achieved the best performance at a 1:1 ratio, recording an Accuracy of 99.94%. This surpasses the Accuracy of 99.87% and 99.84% achieved by SA and GA, respectively, emphasizing the superior optimization capability of BQANA. In contrast, when hyperparameter tuning was not applied, the ensemble model with a 1:1 ratio achieved an Accuracy of 99.77%, which is lower than the results obtained with BQANA. This comparison highlights the added value of hyperparameter optimization.

For Recall, the ensemble model using BQANA at a 1:1 ratio achieved a perfect score of 100%, while GA and SA produced Recall scores of 99.46% and 98.92%, respectively. This result further underscores BQANA's ability to correctly identify fraudulent claims while minimizing false negatives. Without hyperparameter optimization, the Recall at a 1:1 ratio was also 99.46%, but the results with BQANA demonstrated an enhanced capacity for identifying suspicious claims with fewer false positives.

The Precision and F1-score of the ensemble method using BQANA at a 1:1 ratio were also noteworthy, achieving scores of 98.93% and 99.46%, respectively. These metrics consistently outperformed those obtained by SA and GA across all ratios. In comparison, the best Precision and F1-score achieved without hyperparameter tuning were 96.84% and 98.13%, which are lower than the values achieved with BQANA.

Figure 5 visually illustrates the superior performance of the BQANA algorithm across evaluation metrics, particularly for the ensemble model at the 1:1 ratio. Figures 6 and 7 depict the corresponding results for the SA and GA algorithms, respectively, confirming that BQANA consistently outperforms both methods across all metrics and ratios. These findings establish BQANA as a highly effective metaheuristic algorithm for optimizing hyperparameters in ML models for fraud detection, leading to substantial improvements in overall performance.

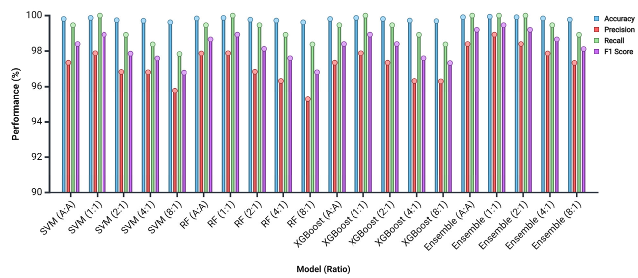

FIGURE 5. BQANA results compare plot.

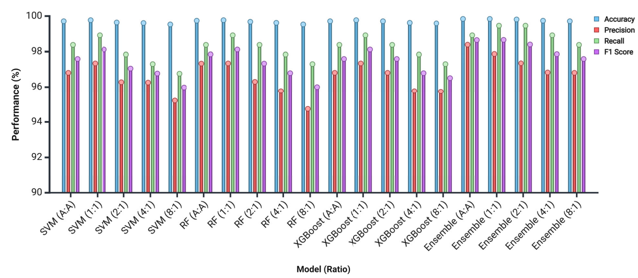

FIGURE 6. SA results compare plot.

We also evaluated the computational efficiency of the BQANA algorithm compared to SA and GA. The runtime required for hyperparameter tuning across the different machine learning models (SVM, RF, XGBoost) and training set ratios (A:A, 1:1, 2:1, 4:1, and 8:1) is detailed in Table 13. The results show that BQANA, while achieving superior performance metrics, requires slightly more computation time than SA and GA due to its quantum-inspired complexity. For instance, when applied to the SVM model with a 1:1 ratio, the runtime for BQANA was approximately 512 seconds, compared to 398 seconds for GA and 312 seconds for SA.

TABLE 12. Evaluation metrics of ML models after hyperparameter tuning.

Model	Train Set Ratio	Accuracy			Precision			Recall			F1-score		
		BQANA	GA	SA	BQANA	GA	SA	BQANA	GA	SA	BQANA	GA	SA
SVM	A:A	99.81	99.74	99.71	97.35	96.83	96.81	99.46	98.92	98.38	98.40	97.86	97.59
	1:1	99.87	99.81	99.77	97.88	97.35	97.34	100	99.46	98.92	98.93	98.40	98.12
	2:1	99.74	99.68	99.64	96.83	96.30	96.28	98.92	98.38	97.84	97.86	97.33	97.05
	4:1	99.71	99.64	99.61	96.81	96.28	96.26	98.38	97.84	97.30	97.59	97.05	96.77
	8:1	99.62	99.56	99.53	95.77	95.24	95.21	97.84	97.30	96.76	96.79	96.26	95.98
RF	A:A	99.84	99.77	99.74	97.87	97.34	97.33	99.46	98.92	98.38	98.66	98.12	97.85
	1:1	99.87	99.81	99.77	97.88	97.35	97.34	100	99.46	98.92	98.93	98.40	98.12
	2:1	99.77	99.71	99.68	96.84	96.32	96.30	99.46	98.92	98.38	98.13	97.60	97.33
	4:1	99.72	99.65	99.62	96.32	95.79	95.77	98.92	98.38	97.84	97.60	97.07	96.79
	8:1	99.62	99.56	99.53	95.29	94.76	94.74	98.38	97.84	97.30	96.81	96.28	96.00
XGBoost	A:A	99.81	99.74	99.71	97.35	96.83	96.81	99.46	98.92	98.38	98.40	97.86	97.59
	1:1	99.87	99.81	99.77	97.88	97.35	97.34	100	99.46	98.92	98.93	98.40	98.12
	2:1	99.81	99.74	99.71	97.35	96.83	96.81	99.46	98.92	98.38	98.40	97.86	97.59
	4:1	99.72	99.65	99.62	96.32	95.79	95.77	98.92	98.38	97.84	97.60	97.07	96.79
	8:1	99.69	99.62	99.59	96.30	95.77	95.75	98.38	97.84	97.30	97.33	96.79	96.51
Ensemble	A:A	99.91	99.87	99.84	98.40	98.40	98.40	100	99.46	98.92	99.20	98.92	98.65
	1:1	**99.94**	99.87	99.84	**98.93**	97.88	97.87	**100**	99.46	98.92	**99.46**	98.93	98.66
	2:1	99.91	99.84	99.81	98.40	97.87	97.35	100	99.46	98.92	98.93	98.40	98.12
	4:1	99.84	99.77	99.74	97.87	97.34	96.83	99.46	98.92	98.92	98.66	98.12	97.86
	8:1	99.77	99.71	99.71	97.34	96.81	96.81	98.92	98.38	98.38	98.12	97.59	97.59

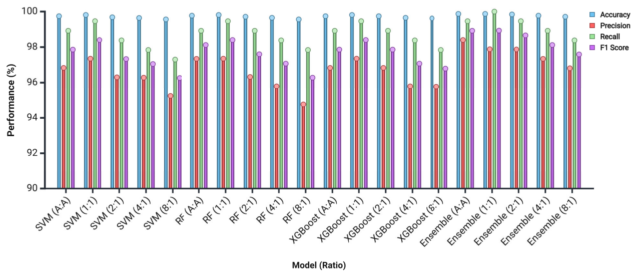

FIGURE 7. GA results compare plot.

This difference is consistent across other models and ratios, as detailed in Table 13, reflecting the trade-off between computational efficiency and enhanced model performance.

The results clearly underscore the importance of selecting optimal hyperparameters over relying on default settings or traditional approaches such as Grid Search, Random Search, or manual tuning. Hyperparameter optimization through metaheuristic methods not only reduces the manual effort involved in deploying machine learning models but also enhances their predictive accuracy and robustness. Furthermore, this approach supports the development of advanced methodologies for solving complex real-world problems, such as insurance fraud detection.

To further validate the effectiveness of the proposed method, we compared the best-performing model obtained in this study—an ensemble model with hyperparameters tuned using BQANA at a 1:1 training set ratio—with models developed in other state-of-the-art studies. The comparison, presented in Table 14, highlights the superiority of our proposed method in classifying insurance claims accurately.

TABLE 13. Runtime comparison for hyperparameter tuning across machine learning models and train set ratios.

Model	Train Set Ratio	SA Runtime (seconds)	GA Runtime (seconds)	BQANA Runtime (seconds)
SVM	A:A	1823	1987	2214
	1:1	312	398	512
	2:1	635	748	915
	4:1	1238	1504	1812
	8:1	1627	1794	2018
RF	A:A	1518	1712	1938
	1:1	267	342	465
	2:1	527	648	789
	4:1	1034	1269	1527
	8:1	1421	1645	1856
XGBoost	A:A	1332	1527	1736
	1:1	234	318	428
	2:1	478	604	792
	4:1	893	1047	1208
	8:1	1234	1412	1603

Our methodology leverages extensive simulations involving various machine learning models (SVM, RF, XGBoost) and hyperparameter tuning algorithms (BQANA, SA, and GA) to identify the optimal model configuration.

The comparative studies involve diverse methodologies, including PSO-based hyperparameter optimization for XGBoost [35], stacking models with oversampling techniques to address class imbalance [39], ensemble CNN models for feature extraction and classification [68], hybrid data augmentation approaches for resampling [27], fuzzy clustering techniques integrated with advanced classifiers [56],

TABLE 14. Comparison of evaluation metrics with related studies.

Study	Model	Accuracy	Precision	Recall	F1-score
This Study	**BQANA-Ensemble (1:1)**	**99.94**	**98.93**	**100.00**	**99.46**
N. Ding et al. [35]	PSO-XGBoost	95.00	95.40	95.00	95.00
A. A. Khalil et al. [39]	Stacking	100.00	91.00	91.00	91.00
Y. Abakarim et al. [68]	Ensemble-CNN	98.00	99.34	98.54	98.94
Z. S. Rubaidi et al. [27]	RF-Hybrid Data Augmentation	97.50	95.60	99.50	97.50
S. K. Majhi. [56]	MWOA-FCM-CATBoost	86.38	61.74	96.25	75.22
M. A. Caruana et al. [23]	NN	93.70	50.00	1.00	2.00

and neural networks optimized for imbalanced datasets [23]. Despite their effectiveness in specific scenarios, the results show that our ensemble model with BQANA tuning outperforms these approaches across multiple evaluation metrics (Accuracy, Precision, Recall, and F1-score), demonstrating its robustness and adaptability to the complex challenge of insurance fraud detection.

The advantages of using metaheuristic algorithms, particularly BQANA, extend beyond superior performance metrics to their potential real-world applicability. For instance, the ensemble model tuned with BQANA achieved an Accuracy of 99.94% and a perfect Recall of 100% at the 1:1 ratio, outperforming SA and GA by notable margins. The balanced dataset provided by the 1:1 ratio amplifies the effectiveness of hyperparameter optimization, enabling the model to accurately detect fraudulent claims while minimizing false positives. These results, supported visually by Figures 5, 6, and 7, highlight the consistent superiority of the BQANA-tuned ensemble method compared to other approaches. Furthermore, the computational efficiency of BQANA, as shown in Table 13, demonstrates its feasibility for practical deployment. While BQANA incurs slightly higher computational costs compared to SA and GA, this is justified by its substantial performance gains, making it a viable solution for real-world scenarios where accuracy is paramount.

The robustness of the proposed framework is particularly relevant in addressing challenges commonly faced in real-world settings, such as noisy or incomplete datasets. The adaptability of metaheuristic algorithms, demonstrated through evaluations across multiple training set ratios, ensures that the models maintain robust performance even in imbalanced or imperfect data conditions. Moreover, the automated nature of hyperparameter tuning minimizes the reliance on manual intervention, making the framework scalable for integration into existing insurance fraud detection workflows.

The practical implications of this study are significant, especially for insurance companies. By accurately identifying fraudulent claims, the proposed framework has the potential to reduce financial losses and administrative burdens. However, challenges such as handling real-world data inconsistencies and ensuring seamless integration with existing IT infrastructures must be addressed. For instance, incorporating preprocessing steps to handle noisy or missing data and optimizing runtime for larger datasets will be crucial for operational deployment. Additionally, the ensemble approach requires compatibility with existing fraud detection systems, which may necessitate customization or incremental integration strategies. Despite these challenges, the demonstrated effectiveness of the BQANA-tuned ensemble model underscores its potential as a transformative tool for modern insurance fraud detection systems.

V. CONCLUSION

Insurance fraud remains a critical challenge, necessitating sophisticated detection systems. This study introduced a robust ensemble learning framework with hyperparameter optimization using the BQANA algorithm, outperforming GA and SA in enhancing model performance. Using the imbalanced Carclaims dataset, the ensemble model tuned with BQANA and a 1:1 ratio achieved 99.94% accuracy, demonstrating superior results across all evaluation metrics. These findings underscore the significance of addressing data imbalances, employing metaheuristic-based hyperparameter tuning, and leveraging ensemble techniques to improve fraud detection systems.

Despite its promising outcomes, this study is limited to a single dataset, which may affect the generalizability of the findings. Additionally, while BQANA achieved the best predictive performance, it incurred slightly higher computational costs compared to GA and SA. Future work should focus on validating the proposed framework on diverse datasets, enhancing computational efficiency, and integrating deep learning techniques to further advance fraud detection capabilities.

DATA AND CODE AVAILABILITY

The dataset and code used in this study are publicly available in the Figshare repository under the DOI: 10.6084/m9.figshare.28207571.

REFERENCES

[1] E. W. T. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and X. Sun, "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature," *Decis. Support Syst.*, vol. 50, no. 3, pp. 559–569, Feb. 2011.

[2] L. Maiano, A. Montuschi, M. Caserio, E. Ferri, F. Kieffer, C. Germanò, L. Baiocco, L. R. Celsi, I. Amerini, and A. Anagnostopoulos, "A deep-learning–based antifraud system for car-insurance claims," *Expert Syst. Appl.*, vol. 231, Nov. 2023, Art. no. 120644.

[3] A. Singhal, N. Singhal, Divya, and K. Sharma, "Machine learning methods for detecting car insurance fraud: Comparative analysis," in *Proc. 3rd Int. Conf. Intell. Technol. (CONIT)*, Jun. 2023, pp. 1–5.

[4] F. Aslam, A. I. Hunjra, Z. Ftiti, W. Louhichi, and T. Shams, "Insurance fraud detection: Evidence from artificial intelligence and machine learning," *Res. Int. Bus. Finance*, vol. 62, Dec. 2022, Art. no. 101744.

[5] F. Gao, J. W. Lien, and J. Zheng, "Exaggerating to break-even: Reference-dependent moral hazard in automobile insurance claims," *Available SSRN*, vol. 2021, pp. 1–51, Feb. 2021.
[6] A. Ahmed, A. F. M. Sadullah, and A. S. Yahya, "Errors in accident data, its types, causes and methods of rectification-analysis of the literature," *Accident Anal. Prevention*, vol. 130, pp. 3–21, Sep. 2019.
[7] H. Lando, "Optimal rules of negligent misrepresentation in insurance contract law," *Int. Rev. Law Econ.*, vol. 46, pp. 70–77, Jun. 2016.
[8] A. M. Macedo, C. V. Cardoso, J. S. M. Neto, and C. A. da Costa Brás da Cunha, "Car insurance fraud: The role of vehicle repair workshops," *Int. J. Law, Crime Justice*, vol. 65, Jun. 2021, Art. no. 100456.
[9] J. Jung and B.-J. Kim, "Insurance fraud in korea, its seriousness, and policy implications," *Frontiers Public Health*, vol. 9, Nov. 2021, Art. no. 791820.
[10] W. Hilal, S. A. Gadsden, and J. Yawney, "Financial fraud: A review of anomaly detection techniques and recent advances," *Expert Syst. Appl.*, vol. 193, May 2022, Art. no. 116429.
[11] T. Badriyah, L. Rahmaniah, and I. Syarif, "Nearest neighbour and statistics method based for detecting fraud in auto insurance," in *Proc. Int. Conf. Appl. Eng. (ICAE)*, Oct. 2018, pp. 1–5.
[12] L. Rukhsar, W. H. Bangyal, K. Nisar, and S. Nisar, "Prediction of insurance fraud detection using machine learning algorithms," *Mehran Univ. Res. J. Eng. & Technol.*, vol. 41, no. 1, pp. 33–40, Jan. 2022.
[13] C. Gomes, Z. Jin, and H. Yang, "Insurance fraud detection with unsupervised deep learning," *J. Risk Insurance*, vol. 88, no. 3, pp. 591–624, Sep. 2021.
[14] S. Mirjalili, "Evolutionary algorithms and neural networks," in *Evolutionary Algorithms and Neural Networks: Theory and Applications* (Studies in Computational Intelligence), vol. 780. Cham, Switzerland: Springer, 2019, pp. 43–53.
[15] M. Rakhshaninejad, M. Fathian, B. Amiri, and N. Yazdanjue, "An ensemble-based credit card fraud detection algorithm using an efficient voting strategy," *Comput. J.*, vol. 65, no. 8, pp. 1998–2015, Aug. 2022.
[16] I. M. Nur Prasasti, A. Dhini, and E. Laoh, "Automobile insurance fraud detection using supervised classifiers," in *Proc. Int. Workshop Big Data Inf. Secur. (IWBIS)*, Oct. 2020, pp. 47–52.
[17] D. Trisanto, N. Rismawati, M. Mulya, and F. Kurniadi, "Effectiveness undersampling method and feature reduction in credit card fraud detection," *Int. J. Intell. Eng. Syst.*, vol. 13, no. 2, pp. 173–181, Apr. 2020.
[18] B. Itri, Y. Mohamed, B. Omar, and Q. Mohamed, "Empirical oversampling threshold strategy for machine learning performance optimisation in insurance fraud detection," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 10, pp. 1–6, 2020.
[19] X. Liang, Y. Gao, and S. Xu, "ASE: Anomaly scoring based ensemble learning for highly imbalanced datasets," *Expert Syst. Appl.*, vol. 238, Mar. 2024, Art. no. 122049.
[20] S. Dalal, B. Seth, M. Radulescu, C. Secara, and C. Tolea, "Predicting fraud in financial payment services through optimized hyper-parameter-tuned XGBoost model," *Mathematics*, vol. 10, no. 24, p. 4679, Dec. 2022.
[21] A. Y. Kusdiyanto and Y. Pristyanto, "Machine learning models for classifying imbalanced class datasets using ensemble learning," in *Proc. 5th Int. Seminar Res. Inf. Technol. Intell. Syst. (ISRITI)*, Dec. 2022, pp. 648–653.
[22] A. A. Feitosa-Neto, J. C. Xavier-Júnior, A. M. P. Canuto, and A. C. M. Oliveira, "A study of model and hyper-parameter selection strategies for classifier ensembles: A robust analysis on different optimization algorithms and extended results," *Natural Comput.*, vol. 20, no. 4, pp. 805–819, Dec. 2021.
[23] M. A. Caruana and L. Grech, "Automobile insurance fraud detection," *Commun. Statist., Case Stud., Data Anal. Appl.*, vol. 7, no. 4, pp. 520–535, Oct. 2021.
[24] A. Ali, S. Abd Razak, S. H. Othman, T. A. E. Eisa, A. Al-Dhaqm, M. Nasser, T. Elhassan, H. Elshafie, and A. Saif, "Financial fraud detection based on machine learning: A systematic literature review," *Appl. Sci.*, vol. 12, no. 19, p. 9637, Sep. 2022.
[25] S. Subudhi and S. Panigrahi, "Use of optimized fuzzy C-means clustering and supervised classifiers for automobile insurance fraud detection," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 32, no. 5, pp. 568–575, Jun. 2020.
[26] M. K. Severino and Y. Peng, "Machine learning algorithms for fraud prediction in property insurance: Empirical evidence using real-world microdata," *Mach. Learn. Appl.*, vol. 5, Sep. 2021, Art. no. 100074.
[27] Z. S. Rubaidi, B. Ben Ammar, and M. Ben Aouicha, "Vehicle insurance fraud detection based on hybrid approach for data augmentation," *J. Inf. Assurance & Secur.*, vol. 18, no. 5, pp. 135–146, 2023.
[28] H. I. Okagbue and O. Oyewole, "Prediction of automobile insurance fraud claims using machine learning," *Sci. Temper*, vol. 14, no. 3, pp. 756–762, Sep. 2023.
[29] E. Nabrawi and A. Alanazi, "Fraud detection in healthcare insurance claims using machine learning," *Risks*, vol. 11, no. 9, p. 160, Sep. 2023.
[30] B. P. Verma, V. Verma, and A. Badholia, "Hyper-tuned ensemble machine learning model for credit card fraud detection," in *Proc. Int. Conf. Inventive Comput. Technol. (ICICT)*, Jul. 2022, pp. 320–327.
[31] O. R. Sanchez, M. Repetto, A. Carrega, and R. Bolla, "Evaluating ML-based DDoS detection with grid search hyperparameter optimization," in *Proc. IEEE 7th Int. Conf. Netw. Softwarization (NetSoft)*, Jun. 2021, pp. 402–408.
[32] M. Tayebi and S. E. Kafhali, "Hyperparameter optimization using genetic algorithms to detect frauds transactions," in *Proc. Int. Conf. Artif. Intell. Comput. Vis.* Springer, Jan. 2021, pp. 288–297.
[33] A. Mehdary, A. Chehri, A. Jakimi, and R. Saadane, "Hyperparameter optimization with genetic algorithms and XGBoost: A step forward in smart grid fraud detection," *Sensors*, vol. 24, no. 4, p. 1230, Feb. 2024.
[34] M. Tayebi and S. El Kafhali, "Performance analysis of metaheuristics based hyperparameters optimization for fraud transactions detection," *Evol. Intell.*, vol. 17, no. 2, pp. 921–939, Apr. 2024.
[35] N. Ding, X. Ruan, H. Wang, and Y. Liu, "Automobile insurance fraud detection based on PSO-XGBoost model and interpretable machine learning method," *Insurance, Math. Econ.*, vol. 120, pp. 51–60, Jan. 2025.
[36] P. Mrozek, J. Panneerselvam, and O. Bagdasar, "Efficient resampling for fraud detection during anonymised credit card transactions with unbalanced datasets," in *Proc. IEEE/ACM 13th Int. Conf. Utility Cloud Comput. (UCC)*, Dec. 2020, pp. 426–433.
[37] C. G. Tekkali and K. Natarajan, "Smart fraud detection in e-transactions using synthetic minority oversampling and binary Harris hawks optimization," in *Proc. Comput., Mater. & Continua*, Jan. 2023, vol. 75, no. 2, pp. 3171–3187.
[38] M. Hanafy and R. Ming, "Using machine learning models to compare various resampling methods in predicting insurance fraud," *J. Theor. Appl. Inf. Technol.*, vol. 99, no. 12, pp. 2819–2833, 2021.
[39] A. A. Khalil, Z. Liu, A. Fathalla, A. Ali, and A. Salah, "Machine learning based method for insurance fraud detection on class imbalance datasets with missing values," *IEEE Access*, vol. 12, pp. 155451–155468, 2024.
[40] X. Li, "Identifying the optimal machine learning model for predicting car insurance claims: A comparative study utilising advanced techniques," *Academic J. Bus. Manag.*, vol. 5, no. 3, pp. 112–120, 2023.
[41] E. Ileberi, Y. Sun, and Z. Wang, "A machine learning based credit card fraud detection using the GA algorithm for feature selection," *J. Big Data*, vol. 9, no. 1, p. 24, Dec. 2022.
[42] A. Ghorbani and S. Farzai, "Fraud detection in automobile insurance using a data mining based approach," *Int. J. Mechatronics, Elektrical Comput. Technol.*, vol. 8, no. 27, pp. 3764–3771, 2018.
[43] H. Ahmad, B. Kasasbeh, B. Al-Dabaybah, and E. Rawashdeh, "EFN-SMOTE: An effective oversampling technique for credit card fraud detection by utilizing noise filtering and fuzzy c-means clustering," *Int. J. Data Netw. Sci.*, vol. 7, no. 3, pp. 1025–1032, 2023.
[44] H. Kaur, H. S. Pannu, and A. K. Malhi, "A systematic review on imbalanced data challenges in machine learning: Applications and solutions," *ACM Comput. Surv.*, vol. 52, no. 4, pp. 1–36, Jul. 2020.
[45] S. Das, S. Datta, and B. B. Chaudhuri, "Handling data irregularities in classification: Foundations, trends, and future challenges," *Pattern Recognit.*, vol. 81, pp. 674–693, Sep. 2018.
[46] F. G. Rebitschek, G. Gigerenzer, and G. G. Wagner, "People underestimate the errors made by algorithms for credit scoring and recidivism prediction but accept even fewer errors," *Sci. Rep.*, vol. 11, no. 1, p. 20171, Oct. 2021.
[47] M. E. Lokanan and V. Maddhesia, "Supply chain fraud prediction with machine learning and artificial intelligence," *Int. J. Prod. Res.*, vol. 63, no. 1, pp. 286–313, Jan. 2025.
[48] S. Subudhi and S. Panigrahi, "Effect of class imbalanceness in detecting automobile insurance fraud," in *Proc. 2nd Int. Conf. Data Sci. Bus. Anal. (ICDSBA)*, Sep. 2018, pp. 528–531.
[49] B. Itri, Y. Mohamed, Q. Mohammed, and B. Omar, "Performance comparative study of machine learning algorithms for automobile insurance fraud detection," in *Proc. 3rd Int. Conf. Intell. Comput. Data Sci. (ICDS)*, Oct. 2019, pp. 1–4.
[50] S. Padhi and S. Panigrahi, "Decision templates based ensemble classifiers for automobile insurance fraud detection," in *Proc. Global Conf. Advancement Technol. (GCAT)*, Oct. 2019, pp. 1–5.

[51] S. Harjai, S. K. Khatri, and G. Singh, "Detecting fraudulent insurance claims using random forests and synthetic minority oversampling technique," in *Proc. 4th Int. Conf. Inf. Syst. Comput. Netw. (ISCON)*, Nov. 2019, pp. 123–128.
[52] N. S. Patil, S. Kamanavalli, S. Hiregoudar, S. Jadhav, S. Kanakraddi, and N. D. Hiremath, "Vehicle insurance fraud detection system using robotic process automation and machine learning," in *Proc. Int. Conf. Intell. Technol. (CONIT)*, Jun. 2021, pp. 1–5.
[53] S.-Z. Shareh Nordin, Y. B. Wah, N. K. Haur, A. Hashim, N. Rambeli, and N. A. Jalil, "Predicting automobile insurance fraud using classical and machine learning models," *Int. J. Electr. Comput. Eng.*, vol. 14, no. 1, p. 911, Feb. 2024.
[54] M. Zhu, Y. Zhang, Y. Gong, C. Xu, and Y. Xiang, "Enhancing credit card fraud detection a neural network and SMOTE integrated approach," 2024, *arXiv:2405.00026*.
[55] M. Abdul Salam, K. M. Fouad, D. L. Elbably, and S. M. Elsayed, "Federated learning model for credit card fraud detection with data balancing techniques," *Neural Comput. Appl.*, vol. 36, no. 11, pp. 6231–6256, Apr. 2024.
[56] S. K. Majhi, "Fuzzy clustering algorithm based on modified whale optimization algorithm for automobile insurance fraud detection," *Evol. Intell.*, vol. 14, no. 1, pp. 35–46, Mar. 2021.
[57] S. Subudhi and S. Panigrahi, "Detection of automobile insurance fraud using feature selection and data mining techniques," *Int. J. Rough Sets Data Anal.*, vol. 5, no. 3, pp. 1–20, Jul. 2018.
[58] E. Ileberi, Y. Sun, and Z. Wang, "Performance evaluation of machine learning methods for credit card fraud detection using SMOTE and AdaBoost," *IEEE Access*, vol. 9, pp. 165286–165294, 2021.
[59] M. Zakariah, S. A. AlQahtani, and M. S. Al-Rakhami, "Machine learning-based adaptive synthetic sampling technique for intrusion detection," *Appl. Sci.*, vol. 13, no. 11, p. 6504, May 2023.
[60] X. Zhao and S. Guan, "CTCN: A novel credit card fraud detection method based on conditional tabular generative adversarial networks and temporal convolutional network," *PeerJ Comput. Sci.*, vol. 9, Oct. 2023, Art. no. e1634.
[61] H. Zamani, M. H. Nadimi-Shahraki, and A. H. Gandomi, "QANA: Quantum-based avian navigation optimizer algorithm," *Eng. Appl. Artif. Intell.*, vol. 104, Sep. 2021, Art. no. 104314.
[62] M. H. Nadimi-Shahraki, A. Fatahi, H. Zamani, and S. Mirjalili, "Binary approaches of quantum-based avian navigation optimizer to select effective features from high-dimensional medical data," *Mathematics*, vol. 10, no. 15, p. 2770, Aug. 2022.
[63] M. H. Nadimi-Shahraki, "An effective hybridization of quantum-based avian navigation and bonobo optimizers to solve numerical and mechanical engineering problems," *J. Bionic Eng.*, vol. 20, no. 3, pp. 1361–1385, May 2023.
[64] R. R. Mostafa, O. Kisi, R. M. Adnan, T. Sadeghifar, and A. Kuriqi, "Modeling potential evapotranspiration by improved machine learning methods using limited climatic data," *Water*, vol. 15, no. 3, p. 486, Jan. 2023.
[65] A. Singh, A. Jain, and S. E. Biable, "Financial fraud detection approach based on firefly optimization algorithm and support vector machine," *Appl. Comput. Intell. Soft Comput.*, vol. 2022, pp. 1–10, Jun. 2022.
[66] I.-S. Oh, J.-S. Lee, and B.-R. Moon, "Hybrid genetic algorithms for feature selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1424–1437, Nov. 2004.
[67] A. Kuznetsov, M. Karpinski, R. Ziubina, S. Kandiy, E. Frontoni, O. Peliukh, O. Veselska, and R. Kozak, "Generation of nonlinear substitutions by simulated annealing algorithm," *Information*, vol. 14, no. 5, p. 259, Apr. 2023.
[68] Y. Abakarim, M. Lahby, and A. Attioui, "A bagged ensemble convolutional neural networks approach to recognize insurance claim frauds," *Appl. Syst. Innov.*, vol. 6, no. 1, p. 20, Jan. 2023.

AFSANEH GHEYSARBEIGI received the M.S. degree in information technology engineering, majoring in e-commerce (machine learning in car insurance fraud detection) from Iran University of Science and Technology, Iran, in 2024. Her research interests include decision support systems, industrial engineering, artificial intelligence, machine learning, and deep learning.

MORTEZA RAKHSHANINEJAD received the M.S. degree in information technology engineering, majoring in e-commerce (machine learning in fraud detection systems) from Iran University of Science and Technology, Iran, in 2019. His research interests include applied machine learning, information systems, computational biology, bioinformatics, and big data.

MOHAMMAD FATHIAN received the M.S. and Ph.D. degrees in industrial engineering from Iran University of Science and Technology, Tehran. He is currently a Professor with the School of Industrial Engineering, Iran University of Science and Technology. He is working in the areas of information technology and industrial engineering. He has more than 90 journal articles and five books in the areas of industrial engineering and information technology.

FARNAZ BARZINPOUR received the Ph.D. degree from Tarbiat Modares University, in 2004. She is currently an Associate Professor of industrial engineering with Iran University of Science and Technology. She teaches operations research, meta-heuristics algorithms, principles of logistics and supply chain engineering, research methodology, and integer programming. Her research interests include optimization and meta-heuristic algorithms, health systems engineering, and supply chain management.