*I. J. Mathematical Sciences and Computing*, **2024, 3, 8-25**
Published Online on September 8, 2024 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijmsc.2024.03.02

# An Automated System for Detecting Property Insurance Fraud Using Machine Learning

**Kazi Md. Tawsif Rahman**
Department of CSE of the Chittagong University of Engineering and technology, Chittagong-4349, Bangladesh
E-mail: u1804113@student.cuet.ac.bd
ORCID iD: https://orcid.org/0009-0005-8509-3270

**Chowdhury Mahfuzul Hoq***
Department of CSE of the Chittagong University of Engineering and technology, Chittagong-4349, Bangladesh
E-mail: chowdhurymhoq@gmail.com
ORCID iD: https://orcid.org/0000-0002-3006-4596
*Corresponding author

Received: 20 June, 2024; Revised: 24 July, 2024; Accepted: 16 August, 2024; Published: 08 September, 2024

**Abstract:** Detecting property insurance fraud is critical for reducing financial losses and ensuring fair claim processing. Traditional methods of detecting insurance fraud had several drawbacks, including no feature selection process, no hyper parameter tuning, lower accuracy, and class imbalance problems. To address the aforementioned shortcomings, this paper examines advanced ML (machine learning) techniques for accurately detecting property insurance fraud. To determine the best model for predicting fraudulent activities, this paper tested several machine learning models, including Gradient Boosting, classical ML classifiers, and Stacking Ensemble methods. To address class imbalance and improve model performance, the selected model incorporates proper feature selection, hyper parameter tuning, and SMOTE techniques (synthetic minority over-sampling). The Stacking Ensemble method outperformed the other ML models, achieving an accuracy of 96% and a recall of 94%. The experimental results show that the proposed stacking ensemble-based prediction scheme improves accuracy by 3.4% and recall by 2.7% over previous works. This article also includes a web application for assisting with property insurance fraud, which includes ML-based fraud prediction, question submission, answer checking, and blog post access. According to the findings, more than 54% of users expressed satisfaction with the web application's usefulness for detecting property fraud.

**Index Terms:** Insurance fraud, machine learning, web application, Ensemble technique, Stacking, SMOTE.

## 1. Introduction

Property insurance fraud can be defined as any act that attempts to defraud a property insurance process by providing false information. Early detection of insurance fraud can help companies and customers avoid serious losses. Currently, insurance fraud costs consumers hundreds of billions of dollars. According to [1], the global rate of claims fraud is nearly 4%, with the Asia Pacific region having the highest incidence. In 2021, the United States reported suspected fraud in 20% [2] of insurance claims, an increase over the previous year. This fraudulent activity cost American consumers USD 308.6 billion [3] per year, affecting both insurance companies and ordinary policyholders who lost money. Insurance fraud in the property and business sectors receives little media and prosecutorial attention. For many years, the insurance industry has been plagued by fraudulent claims. These frauds result in significant losses for all parties involved. Fraud can result in significant financial loss, increased premium costs, increased inefficiencies during process execution, and a loss of trust. Despite fraud detection systems, insurance companies continue to face inefficient and time-consuming processes. Traditional methods were unable to adapt to dynamic situations due to the presence of human workers and manual processes. Prolonged investigations delay payouts and reduce customer satisfaction.

Fraudulent claims can lower profits and encourage similar behavior among policyholders [4, 5]. By 2024, the gross written premium market for property insurance in Bangladesh is expected to reach USD1.89 billion. This means that people in Bangladesh are willing to invest in insurance for protection and peace of mind. The gross written premium is expected to rise by 4.80% annually between 2024 and 2028, reaching USD 2.28 billion in 2028. Bangladesh's increasing demand for property insurance suggests that the sector has room for further growth [6]. Bangladesh's

property insurance market is expected to expand significantly due to increased awareness, economic development, and demand for protection against property-related risks. That is why creating an automated system can help protect against financial fraud. Insurance fraud has a significant financial impact on the insurance industry.

Fraudulent claims cause significant losses in the property insurance industry. To reduce economic losses in the insurance industry, a machine learning-based property insurance fraud prediction system with high accuracy is required. Traditional fraud detection methods, while somewhat effective, frequently rely on manual processes and static rule-based systems, which are limited in their ability to adapt to changing fraudulent behaviors [7, 8, 9, and 10]. This emphasizes the importance of automated methods that can adapt and learn.

Currently, there are some literary works [11, 12, 13, 14, 15, 16, 17, and 18] on the development of property insurance fraud prediction systems. Existing works on the property insurance fraud prediction had several limitations, including a lack of an important feature selection process, a lack of a reliable dataset, increased false positives, a lack of proper hyper parameter tuning technique usage, a lack of normalization and data cleaning processes, a failure to use ensemble models such as stacking and bagging. The existing works suffered from issues like lower prediction accuracy and imbalanced dataset issues. They did not optimize the machine learning algorithm parameters to achieve the best prediction results. The existing works did not examine several necessary features for the property insurance fraud prediction system, including claim type, claim date, claim severity, discrepancies, suspicious circumstances, delayed reporting, witness statements, and surveillance evidence. Insurance fraud datasets are often skewed, with far fewer fraudulent claims than legitimate ones. Machine learning approaches that deal with unbalanced data can increase property insurance fraud detection rates while decreasing false positives.

The existing research did not investigate the ideal balance of model complexity and interpretability. The existing research did not investigate data imbalance issues. They did not offer an automated system for property insurance fraud prediction assistance using any mobile or web application. To address these limitations, the goal of this paper is to create an automated system for detecting property insurance fraud with high accuracy by utilizing machine learning techniques. The system has three major components: data collection, preprocessing, and machine learning model implementation. By incorporating machine learning models into insurance operations, the system aims to improve fraud detection capabilities, operational efficiency, and the interests of both insurers and policyholders. This work aims to solve the data imbalance issues by using SMOTE techniques (synthetic minority over-sampling). This paper tries to increase the property insurance fraud prediction accuracy result by using the Stacking Ensemble method with proper feature selection and the hyper parameter tuning method. Our second goal is to develop a web application that includes property insurance fraud prediction, question assistance, and suggestion features for general users. The key contributions of this article are depicted as follows:

(i) This paper employs several ML algorithms like SVM, XGBoost, naïve bayes, MLP, stacking ensemble, classical ML classifiers, boosting, and decision trees to accurately predict property insurance fraud incidents.

(ii) This article creates a valid dataset of property insurance claims and applies advanced preprocessing techniques to address data imbalances and improve model training activities. The proposed property insurance fraud prediction model also incorporates appropriate feature selection, adaptive hyper parameter tuning, and proper normalization methods.

(iii) This paper determines the best property insurance fraud prediction model by comparing the accuracy value, recall value, RMSE, MSE, and MAE values for different machine learning models. This paper also compares the proposed property insurance fraud prediction model to previous research.

(iv) This article develops a web application with features like property insurance fraud prediction, question submission, answer assistance, home page, suggestions, and blog post submission, among others. This article also evaluates the web application's suitability by requesting user feedback on its features.

Section two summarizes previous research on property insurance fraud prediction. Section three presents the main methodology of our proposed machine learning-based property insurance fraud prediction model, including evaluation results. Section four illustrates the web application's features. Section five reports on the web application's customer survey results. Section six summarizes the paper's findings and discusses research challenges.

## 2. Related Works

The advancement of machine learning technologies has enabled novel solutions in a variety of industries, including property insurance. This section summarizes some related works on machine learning-based fraud prediction systems.

In [11], the author predicted healthcare insurance fraud using the Random Forest (RF) and XGBoost classifiers and a Medicare dataset. They discovered that when combined with random undersampling to address class imbalance, the XGBoost algorithm outperformed the Random Forest method in terms of recall (87%) and overall accuracy (86%). The authors also identified key fraud prediction features, such as the average insurance claim amount reimbursed, mean duration, and specific claims. The authors' contribution is to apply the XGBoost algorithm to healthcare insurance fraud detection and show that it outperforms the widely used Random Forest method. However, they acknowledged that the

results could be influenced by the training data and suggested looking into additional provider-specific features to improve performance. They did not work on property insurance fraud detection and data unbalancing issue.

In [12], the authors employed nine predictive models for detecting fraud in a Brazilian insurance company. In their study, the random forest model outperformed other classical methods, achieving an accuracy of 84.56%. However, their work had some limitations, including data credibility and potential bias. The data from a major Brazilian insurance company may not be completely generalizable to other regions or insurance companies, compromising the external validity of the findings. Other machine learning algorithms or ensemble techniques may also be able to deliver competitive or superior results. They did not use under sampling to correct the data imbalance. Their dataset is small, and they did not look into many important features for property fraud prediction.

The authors of [13] present a study on using machine learning techniques, specifically the Naive Bayes Classifier and the Gradient Boosting Classifier, to detect auto insurance fraud, a major issue that costs insurance companies billions of dollars each year. To validate their findings, the author applied a variety of machine learning models. Gradient boosting outperformed other techniques, achieving 90% accuracy, 92% precision, and 89% recall. This work is limited to automobile fraud and does not include other types of property fraud. The accuracy of their proposed model is also reduced.

In [14], the authors used a decision tree algorithm to detect insurance fraud. Their evaluation results indicated that the decision tree classifier has a maximum accuracy of 79%. Adaboost, a boosting algorithm, also performed well, with an accuracy of 78%, trailing only the DT algorithm. The authors concluded that DT and Adaboost are promising approaches for detecting insurance fraud because they can handle complex data while producing interpretable results. The study used a small dataset of only 110 customer records with a limited number of attributes. The models' performance may vary when applied to larger and more diverse datasets.

The authors of [15] presented a system for predicting auto insurance fraud based on the random forest algorithm. The proposed methodology involved data collection, preprocessing, feature selection, and classification using ten different machine learning algorithms. The algorithms' performance was evaluated using the F-measure, K-measure, and Root Mean Squared Log Error (RMSE). Their simulation results showed that the classical random forest method had the highest accuracy of 89.57%, outperforming other machine learning models. They did not look into several key aspects of insurance fraud prediction, such as claim type, claim date, claim severity, discrepancies, suspicious circumstances, delayed reporting, witness statements, and surveillance data. They did not use any appropriate hyper parameter tuning techniques.

In [16], the authors used a random forest classifier to identify fraudulent vehicle insurance claims. Their work is limited to the automotive industry. The data imbalance problem was investigated using the oversampling technique [17]. They did not investigate several aspects of fraud prediction, including suspicious circumstances, delayed reporting, witness statements, and surveillance footage. The authors used the Random Forests classification algorithm to predict insurance fraud and achieved an overall accuracy of 94.33%. In [18], the author used not only a genetic algorithm (GA), but also a fuzzy c-means clustering (FCM) scheme combined with some classical classifiers to predict vehicle insurance fraud. The GAFCM clustering algorithm is used to under sample the class samples, resulting in a balanced training dataset.

In their work, the SVM classifier performs the best, with a recall value of 83.21% and a precision value of 88.45%. However, the fraud prediction accuracy result of their work is insufficient and they did not investigate the various factors associated with property insurance fraud prediction.

The authors of [19] tested multiple classical ML classifiers for predicting Egyptian vehicle insurance fraud. The dataset had an extremely unbalanced class distribution. Their results showed that the stochastic gradient boosting model performed best, with an accuracy of 87.15% and an AUC of 88.4%. The study emphasized the importance of managing class imbalance and using appropriate evaluation metrics in addition to accuracy, especially for imbalanced datasets like insurance fraud detection problems. The study did not investigate the impact of feature selection or dimensionality reduction techniques, which could improve model performance even more.

The authors of [20] proposed a machine learning-based automated prediction system for detecting insurance fraud. In their work, the stacking model achieved 91% accuracy. Stacking, which combines the outputs of multiple base classifiers, can frequently improve overall predictive performance when compared to a single classifier. Despite the fact that the stacking ensemble model employs a limited set of base classifiers, including more diverse classifiers may improve the ensemble's performance even further. Advanced data preprocessing techniques, such as feature engineering and dealing with imbalanced data through various sampling/weighting methods, could be tested. The comparison with other ensemble models, such as gradient boosting, bagging, or deep learning methods, is lacking. The authors of [24] used a decision tree-based classifier to forecast health insurance fraud. They did not, however, address the issues of data imbalance, feature engineering, or hyper parameter tuning. They also did not consider various property and insurance factors when predicting fraud. The article in [25] used the quantum SVM technique to predict housing insurance fraud. Their work has limitations, including a low accuracy rate (92.67 percent) and a small dataset. They also failed to investigate the various required property insurance factors for fraud prediction. Several of the studies mentioned above did not look into data imbalance, normalization, and encoding, scaling, or important feature selection issues. The majority of studies employed a small dataset and a limited number of features. They also had lower accuracy results. Some works did not employ hyper parameter tuning techniques.

There is also a lack of an efficient prediction model selection method that uses classical ML classifiers, stacking, and bagging techniques. They also did not assess the model's performance across multiple metrics, including accuracy, MAPE and MSE error values, precision, recall, and ROC-AUC.

Differ from existing works, this article introduces a suitable ML model selection-based property insurance fraud prediction system that achieves high accuracy by addressing data imbalance, hyper parameter tuning, scaling, and dataset cleaning issues. This work compares several classifiers and selects the best one with a high accuracy value. This article also includes a web application for predicting property insurance fraud, complete with fraud detection, question submission, answer checking, and suggestions.

## 3. Proposed Scheme

This section presents our proposed machine learning-based property insurance fraud prediction system, along with evaluation results and a step-by-step discussion.

### *3.1. Overview of proposed machine learning based property insurance fraud prediction system*

Figure 1 delivers a high-level overview of the steps involved in predicting fraudulent claims in property insurance. The process begins with providing the data pipeline with the necessary datasets. We began by gathering pertinent insurance claim information from various sources. The following stage involved cleaning and preparing this data for analysis, ensuring its accuracy and suitability for machine learning applications. Following preprocessing, we used feature extraction algorithms to identify and select the most significant features that could indicate fraudulent behavior. We then used the processed data to create and train a variety of machine learning models. We evaluated 12 machine learning classifiers, including classical ML classifiers such as RF, stacking ensemble model, XGBoost, and naïve bayes, to determine the most effective model selection. The stacking ensemble method is used in this study because it reduces bias and variation, increases model variety, and improves the interpretability of the final forecast by combining predictions from multiple base models. We also used other classical ML classifiers to see how our proposed stacking ensemble model outperformed existing fraud prediction models. Our dataset includes both numerical and categorical values. Due to nature of the dataset, the proposed 12 machine learning classifiers would be effective in predicting insurance fraud from such a dataset. These models were then evaluated against common benchmarks to determine their effectiveness and utility in detecting fraud. We evaluated the performance of the compared ML classifiers using a variety of metrics, including accuracy, confusion matrix, precision, recall, and some error metrics. Then, based on the highest accuracy, we chose the best property insurance fraud prediction model for deployment. This paper also includes the initiation of a web application for detecting property insurance fraud in Bangladesh. Figure 2 depicts a flowchart of the web application's working steps. The front-end user interface is created using the Streamlit platform [21], while the back-end functionality is written in Python. The application stores and retrieves data using the MongoDB platform, a NoSQL database [22]. This design facilitates efficient data management and seamless integration of the user interface and machine learning model for fraud detection. Data is saved and retrieved from the MongoDB database using Python's PyMongo package [23], ensuring that the application and database communicate seamlessly. The web application includes features such as registration, property insurance fraud prediction using machine learning, question submission, home page, answer checking, suggestion checking, and blog post creation, among others. The developed web application allows the user to check for fraudulent insurance fraud activity. The user can also post questions about insurance fraud and receive responses.

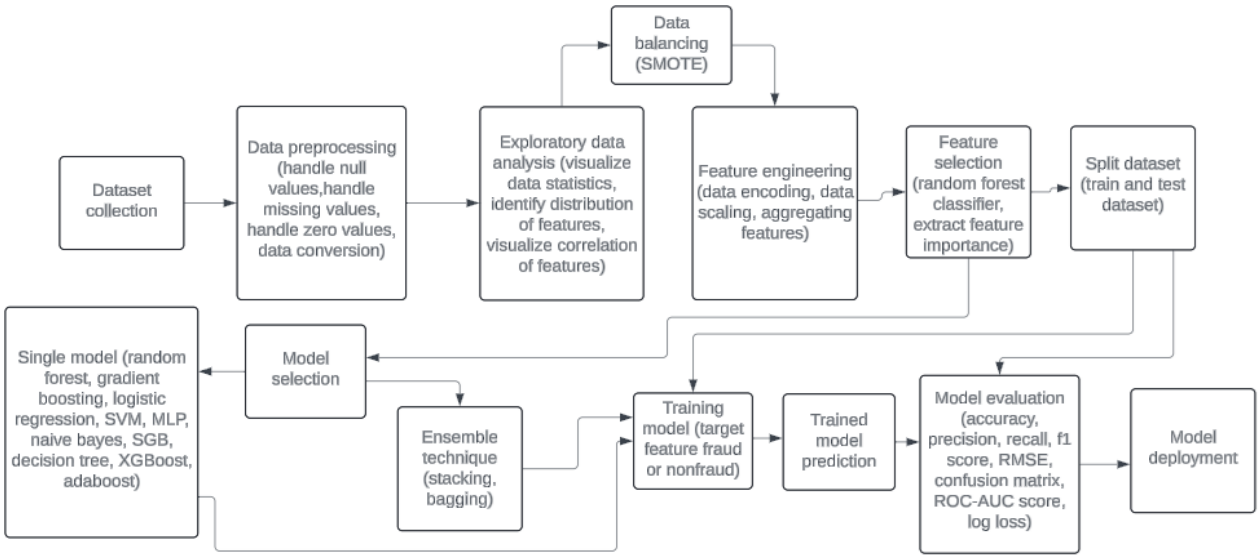



Fig. 1. Proposed Machine Learning Model Framework for Property Insurance Fraud Prediction

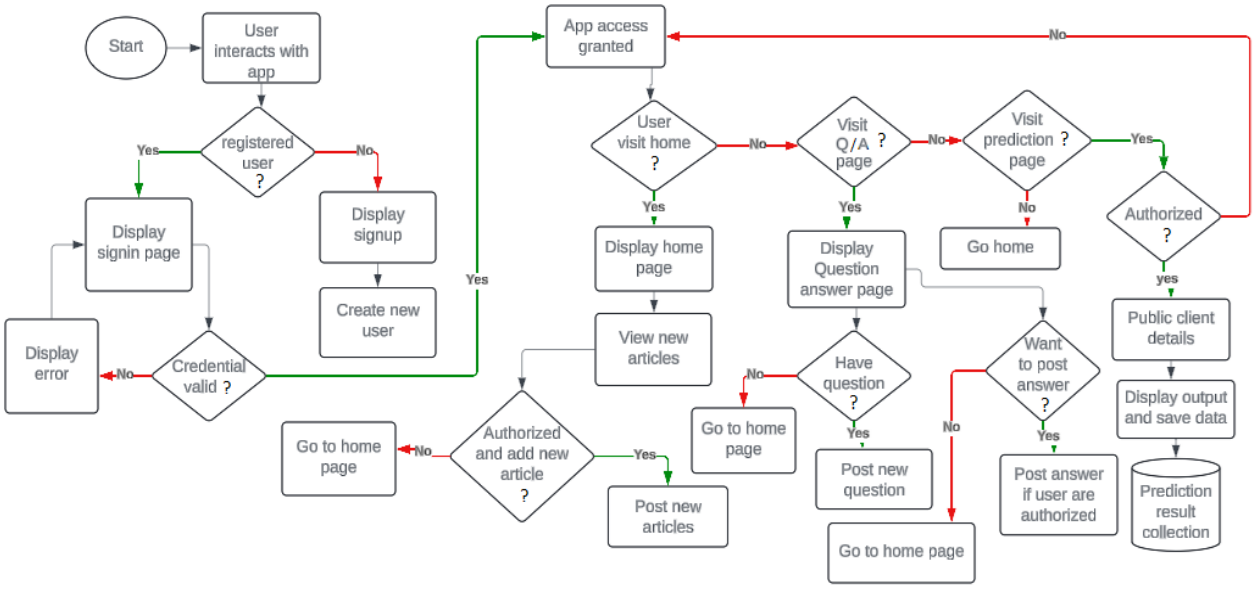



Fig. 2. Flowchart Model for the Proposed Property Insurance Fraud Assistance Web Application

| Age | Gender | Occupatio | Address | cy Age (Ye | verage ty | rage Amo | vious Clai | m Freque | Claim Dat | cedent Da | Claim Type | aim Amou | aim Sever |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 45 | M | Teacher | Dhaka | 10 | Fire, Thef | 20000 | 2 | 0.2 | 5/1/2024 | 4/29/202 | Fire | 15000 | Major |
| 38 | F | Engineer | Chattogra | 5 | Theft | 15000 | 0 | 0 | 4/25/202 | 4/22/202 | Theft | 10000 | Minor |
| 50 | M | Businessn | Sylhet | 8 | Water Da | 10000 | 1 | 0.125 | 5/5/2024 | 5/3/2024 | Water | 12000 | Minor |
| 60 | F | Retired | Barisal | 12 | Fire, Storr | 25000 | 3 | 0.25 | 4/20/202 | 4/18/202 | Storm | 20000 | Major |
| 29 | M | Shopkeep | Dhaka | 2 | Fire | 8000 | 0 | 0 | 5/3/2024 | 5/1/2024 | Fire | 7500 | Minor |
| 35 | F | Doctor | Dhaka | 6 | Fire, Thef | 22000 | 1 | 0.167 | 5/6/2024 | 5/4/2024 | Theft | 18000 | Major |
| 42 | M | Lawyer | Sylhet | 5 | Water Da | 12000 | 2 | 0.4 | 5/7/2024 | 5/5/2024 | Water | 15000 | Major |

| aphic Lo | us Circum | cial Diffic | erated Da | screpanci | ved Repo | ess State | illance Evi | raud or No |
|---|---|---|---|---|---|---|---|---|
| Low-risk | No | No | No | No | No | Yes | Yes | Non-Fraud |
| High-risk | Yes | Yes | Yes | Yes | Yes | No | No | Fraud |
| Moderate | No | No | No | No | No | Yes | Yes | Non-Fraud |
| Low-risk | No | Yes | Yes | Yes | Yes | No | No | Fraud |
| High-risk | Yes | Yes | Yes | Yes | Yes | No | No | Fraud |
| Low-risk | No | No | No | No | No | Yes | Yes | Non-Fraud |
| Moderate | Yes | Yes | Yes | Yes | Yes | No | No | Fraud |

Fig. 3. Sample dataset for property insurance fraud prediction

### *3.2. Data collection and data set preparation*

The first step in developing our prediction model is to collect data on property insurance claims. The dataset was obtained from Crystal Insurance Company Limited, a well-known insurance company in Bangladesh. We conducted an online interview with Crystal Insurance Company Limited officials about data entries, dataset validity, and important factors related to fraudulent claims, and property insurance fraud patterns. Figure 3 depicts a portion of the property insurance fraud prediction dataset. The dataset has 420 data entries. The dataset includes some additional features that correspond to the patterns provided for potential fraud, such as exaggerated damages, financial difficulties, and suspicious circumstances. The dataset includes 23 columns (features). Figure 4 displays the features associated with the insurance fraud prediction dataset. Our dataset has the following features: coverage type, coverage amount, previous claims, claim amount, claim severity, geographic location, delayed reporting, witness statements, surveillance evidence, and financial difficulties, among others. The dataset is validated by two insurance company specialists and a university based scientist. We want to mention that the small dataset (420 data entries) may limit the generalizability of the findings. We attempted to contact several insurance companies and authorities about the large number of verified dataset. Only one insurance company wanted to provide the dataset. However, other insurance authorities refuse to provide insurance data due to a lack of records and customer data protection issues. Ensemble learning techniques like stacking, bagging, and boosting combine multiple weak models to improve generalization performance. Stacking ensemble combines predictions from multiple base models to reduce bias and variation, increase model variety, and improve forecast interpretability. Stacking ensemble models reduces variance and bias, prevents over fitting. Furthermore, the differences between training and testing errors are small, indicating that our work would not suffer from this over fitting problem.

```
[69] names = df.columns
     print(names)

Index(['Age', 'Gender', 'Occupation', 'Address', 'Policy Age (Years)',
       'Coverage type', 'Coverage Amount', 'Previous Claims',
       'Claim Frequency', 'Claim Date', 'Incedent Date', 'Claim Type',
       'Claim Amount', 'Claim Severity', 'Geographic Location',
       'Suspicious Circumstances', 'Financial Difficulties',
       'Exaggerated Damages', 'Discrepancies', 'Delayed Reporting',
       'Witness Statements', 'Surveillance Evidence', 'Fraud or Not'],
      dtype='object')
```

Fig. 4. Dataset features visualization

```
df.isnull().sum()

Age                         0
Gender                      0
Occupation                  0
Address                     0
Policy Age (Years)          0
Coverage type               0
Coverage Amount             0
Previous Claims             0
Claim Frequency             0
Claim Date                  0
Incedent Date               0
Claim Type                  0
Claim Amount                0
Claim Severity              0
Geographic Location         0
Suspicious Circumstances    0
Financial Difficulties      0
Exaggerated Damages         0
Discrepancies               0
Delayed Reporting           0
Witness Statements          0
Surveillance Evidence       0
Fraud or Not                0
dtype: int64
```

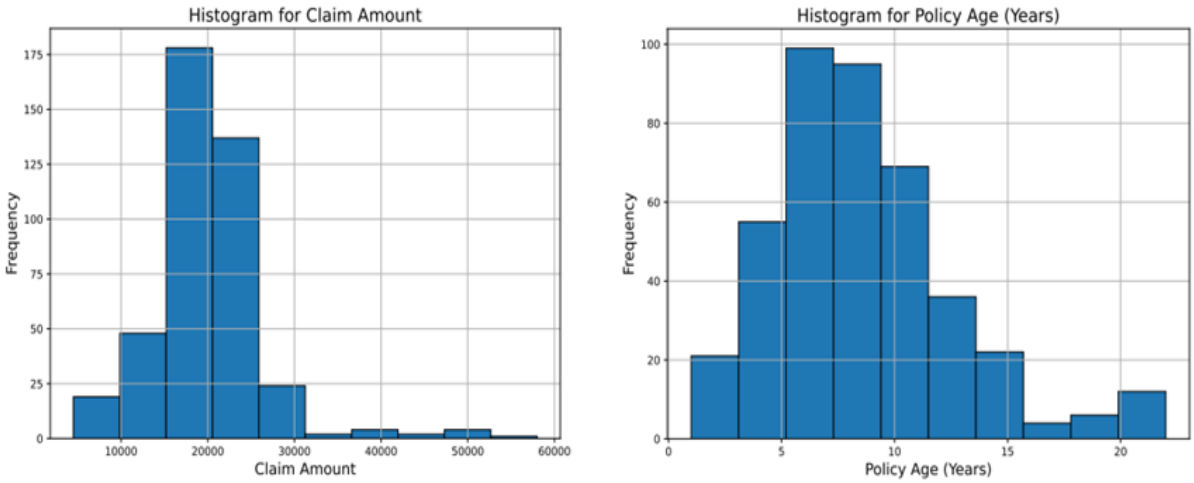
Fig. 5. Checking for null values in the dataset

### *3.3. Data Cleaning, Data Preprocessing, and Visualization*

Data cleaning and preprocessing are critical components of our machine learning-based prediction system. Handling missing and null values is a vital component of data preparation. In this step of our system, we checked for missing and null values associated with the prepared insurance fraud prediction dataset. Figure 5 shows that the null values in each column are zero. This indicates that the dataset has no null values. The dataset contains date values. Dates sometimes need to be correctly formatted. Figure 6 shows how date entries are handled and formatted correctly. In this step, we also removed any unnecessary columns from the dataset.

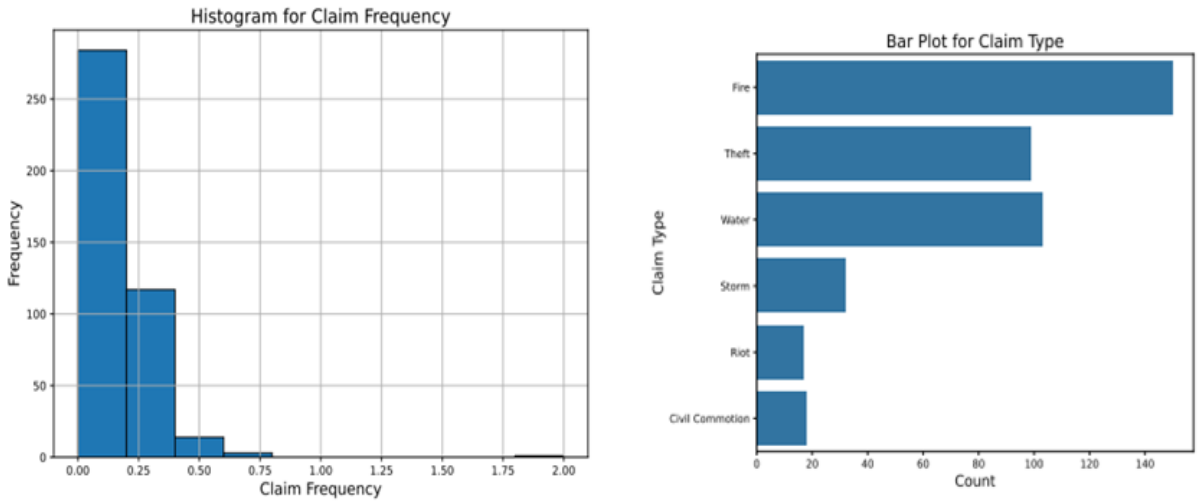
```
[160] df['Claim Date'] = pd.to_datetime(df['Claim Date'], format='%d/%m/%Y')
      df['Incedent Date'] = pd.to_datetime(df['Incedent Date'], format='%d/%m/%Y')
      df['Claim Date']

      0     2024-05-01
      1     2024-04-25
      2     2024-05-05
      3     2024-04-20
      4     2024-05-03
              ...
      414   2025-10-17
      415   2025-10-18
      416   2025-10-19
      417   2025-10-20
      418   2025-10-21
      Name: Claim Date, Length: 419, dtype: datetime64[ns]
```

Fig. 6. Formatting date values of our dataset




(a) (b)

Fig. 7. Claim amount and policy age data visualization of our dataset




(a) (b)

Fig. 8. Claim frequency and claim type data visualization of our dataset

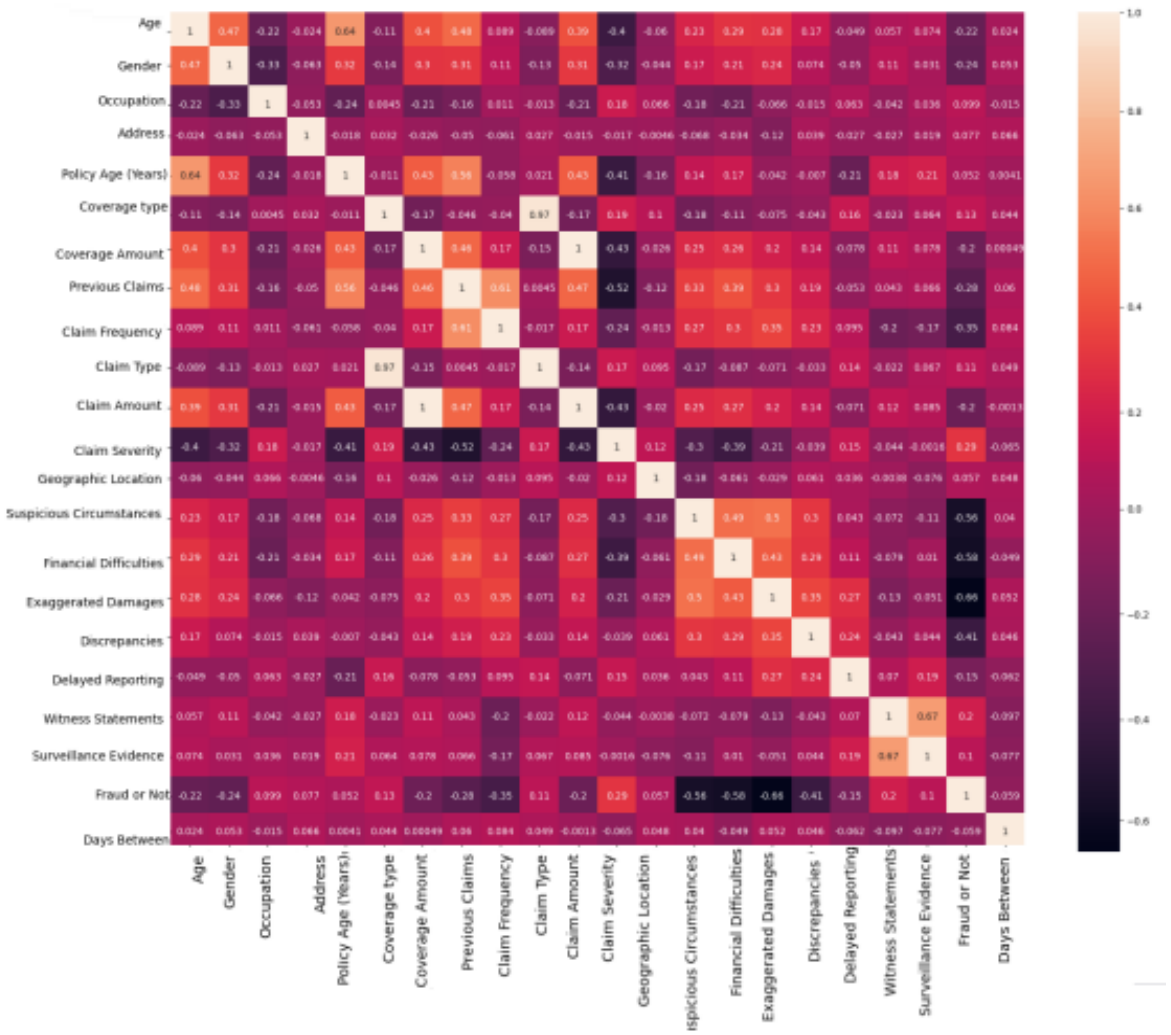



Fig. 9. Correlation Heat map

```
smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X, y)
```

```
print(y_resampled.value_counts())
```

```
Fraud or Not
1    312
0    312
Name: count, dtype: int64
```

Fig. 10. Applying SMOTE to solve class imbalance

Visualizing data distributions can help detect feature importance. We used Exploratory Data Analysis (EDA) for key property analysis. Figure 7 depicts the claim amount and policy age feature value plots from our dataset. The policy age histogram (figure 7(b)) is right-skewed, indicating a higher prevalence of policies with younger policy ages. This could indicate a large number of recently issued policies or a tendency for policyholders to renew or update their coverage frequently. The claim amount histogram (figure 7(a)) shows a unimodal distribution, with the highest frequency of claims falling between 20,000 and 30,000. This range may correspond to common claim amounts or coverage limits for the types of policies offered. Figure 8 depicts the claim frequency and claim type feature value plots from our dataset. The claim frequency histogram also has a strongly right-skewed distribution, with a high proportion of low claim frequencies (around 0). This indicates that the majority of policyholders have a low claim frequency, which may be advantageous from an insurance standpoint. Figure 8(b) shows that the most fraudulent claims are for theft, fire incidents, water issues, storms, and riots. Figure 9 depicts the correlation coefficients between pairs of features. It allows you to visually analyze the relationships and patterns between multiple features at the same time. Darker and lighter colors in the map depict stronger and weaker correlations, respectively.

```
from sklearn import preprocessing
label_encoder = preprocessing.LabelEncoder()
for col in df.select_dtypes(include=['object']).columns:

    label_encoder.fit(df[col].unique())
    df[col] = label_encoder.transform(df[col])

    print(f"{col}: {df[col].unique()}")
```

```
Gender: [1 0]
Occupation: [22 10  3 17 20  8 14  2 11 12 21  6  7  4 13 15 16  0  1  9 18 19  5]
Address: [4 1 9 0 7 3 5 8 2 6]
Coverage type: [3 6 7 2 1 5 4 0]
Claim Type: [1 4 5 3 2 0]
Claim Severity: [0 1]
Geographic Location: [2 0 4 3 5 1]
Suspicious Circumstances: [0 1]
Financial Difficulties: [0 1]
Exaggerated Damages: [0 1]
Discrepancies: [0 1]
Delayed Reporting: [0 1]
Witness Statements: [1 0]
Surveillance Evidence: [1 0]
Fraud or Not: [1 0]
```

Fig. 11. Encoding Categorical Values using Label Encoding

Unbalanced datasets include more than half of the items from a single class. Most ML algorithms perform well with a balanced datasets because they aim to improve overall classification accuracy or other metrics. To solve the problem of imbalanced target features, we used the SMOTE data balancing technique. Figure 10 depicts the process of using smote to address class imbalance issues. While SMOTE is extremely effective at mitigating class imbalance issues, it does have some drawbacks and limitations. One disadvantage of SMOTE is that it does not account for the majority class when creating synthetic samples. It can also create synthetic samples in between samples to represent noise.

### *3.4. Feature selection and normalization*

Data encoding and feature selection is a critical step in preparing data for ML model. Conversion of a categorical variable to machine learning-friendly formats improves model accuracy. We used Label encoding to handle categorical values, as depicted in Figure 11. The figure depicts the application of data encoding to categorical features in the dataset, as well as the resulting feature values. Scaling Data scaling is an important preprocessing step in machine learning, especially when using algorithms that depend on the magnitude of feature values. The Standard Scaler is a popular technique for scaling features. Figure 12 shows how we normalized the data using Standard Scaler. Using the Standard Scalar helps to normalize feature values, this improves the performance and reliability of ML models. Feature selection can be termed as the selection of relevant characteristics from a dataset based on specific criteria. Figure 13 depicts the key feature extraction process using a machine learning classifier model (e.g., random forest model). Figure 14 shows the feature's importance in descending order. The graph aids in understanding which features contribute higher to the prediction model's decision-making process, providing insights into the factors that predict the target variable.

This visualization is critical for feature selection and understanding model behavior, particularly in fields such as fraud detection, where identifying key indicators is essential. Then we chose the top ten features based on their feature importance. The most prominent characteristics are financial difficulties, exaggerated damages, suspicious circumstances, claim frequency, discrepancies, claim amount, address, coverage amount, age, and policy age. Figure 15 depicts the results of selecting the top ten features from our dataset.

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
array([[ 0.79417816,  0.81786082,  1.13072491, ..., -0.86719899,
        -1.19725072,  0.43695628],
       [ 0.79417816,  0.81786082,  1.13072491, ...,  0.08102734,
         0.69848671,  0.43695628],
       [ 0.79417816,  0.81786082,  1.13072491, ..., -0.73173808,
        -0.92643109, -2.2885585 ],
       ...,
       [ 0.79417816, -1.22270193, -0.88438841, ..., -0.18989447,
        -0.11397219,  0.43695628],
       [ 0.79417816,  0.81786082,  1.13072491, ...,  0.48741005,
        -0.11397219,  0.43695628],
       [-1.2591633 ,  0.81786082, -0.88438841, ..., -0.18989447,
         0.69848671,  0.43695628]])
```

Fig. 12. Applying scalar function to the dataset

```
model = RandomForestClassifier(random_state=42)
model.fit(X_resampled, y_resampled)
```

```
RandomForestClassifier(random_state=42)
```

```
[86]
importances = model.feature_importances_
feature_names = X.columns
feature_importance_df = pd.DataFrame({'Feature': feature_names, 'Importance': importances})
feature_importance_df = feature_importance_df.sort_values(by='Importance', ascending=False)
```

Fig. 13. Extracting Feature Importance Values

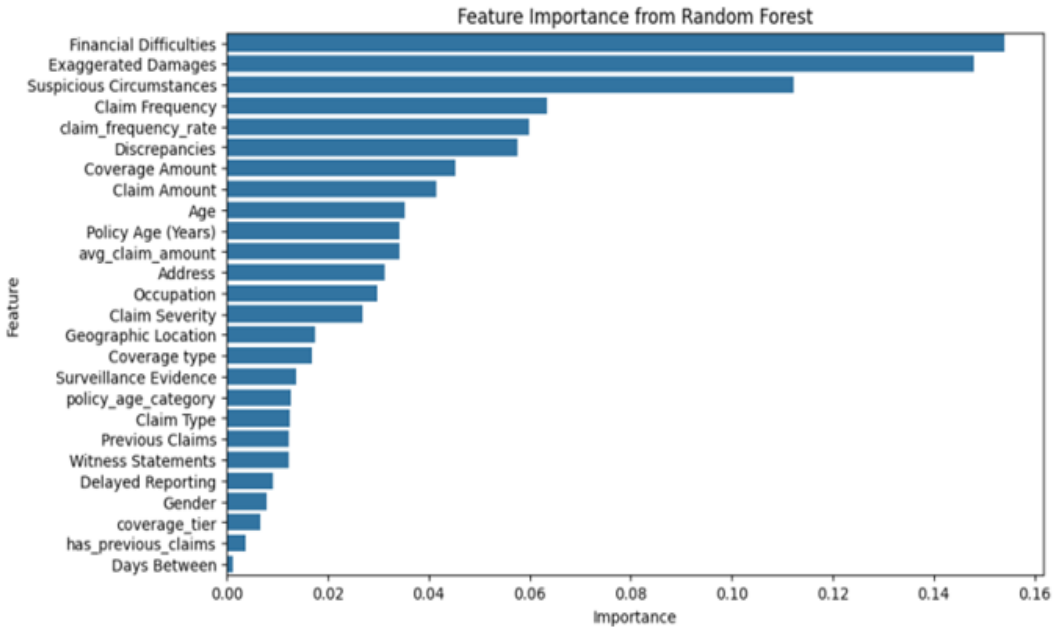



Fig. 14. Feature Importance Results in Descending Order

```
X_selected = X_resampled[top_features]
X_selected
```

| | Suspicious Circumstances | Financial Difficulties | Exaggerated Damages | Discrepancies | Claim Frequency | Age | Address | Claim Amount | Policy Age (Years) | Surveillance Evidence |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 0.200000 | 45 | 4 | 15000.000000 | 10 | 1 |
| **1** | 1 | 1 | 1 | 1 | 0.000000 | 38 | 1 | 10000.000000 | 5 | 0 |
| **2** | 0 | 0 | 0 | 0 | 0.125000 | 50 | 9 | 12000.000000 | 8 | 1 |
| **3** | 0 | 1 | 1 | 1 | 0.250000 | 60 | 0 | 20000.000000 | 12 | 0 |
| **4** | 1 | 1 | 1 | 1 | 0.000000 | 29 | 4 | 7500.000000 | 2 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **619** | 1 | 0 | 0 | 0 | 0.161912 | 46 | 7 | 26238.236152 | 20 | 1 |
| **620** | 1 | 1 | 1 | 1 | 0.291072 | 44 | 0 | 13377.728886 | 7 | 1 |
| **621** | 0 | 1 | 1 | 1 | 0.237570 | 36 | 0 | 24000.000000 | 8 | 1 |
| **622** | 1 | 1 | 1 | 1 | 0.417074 | 38 | 2 | 17000.000000 | 4 | 0 |
| **623** | 1 | 1 | 0 | 1 | 0.269481 | 46 | 3 | 21000.000000 | 11 | 1 |

624 rows × 10 columns

Fig. 15. Dataset with Only Selected Features

### *3.5. Model selection, hyper parameter tuning, training, testing, and cross validation*

The next step is to select the most beneficial model for predicting property insurance fraud. To do so, we compared the performance of twelve machine learning classifiers. We have used scikit-learn's train-test-split function for the dataset training and testing split up, as illustrated in figure 16. This split is critical for determining the model's performance. The test-size parameter determines how much data is given for testing (20%). After splitting, the resulting array shapes are as follows: X_train has 499 samples with 10 features each, X_test has 125 samples with the same 10 features, and y_train and y_test have 499 and 125 target values, respectively. Tuning hyperparameters is critical for achieving the best machine learning parameters. Choosing the best hyper-parameters can be time-consuming, especially when the goal functions are expensive or many parameters must be adjusted. In Figure 17, we used the GridSearchCV hyper parameter optimization technique. We also performed tenfold cross validation. After hyper parameter tuning, we determined the optimal parameters for each ML model. Following the selection of a suitable ML model, these optimal parameters will be used to predict property insurance fraud incidents. Figure 18 shows the best parameter selection for random forest classifier using GridSearchCV. We have used 20 percent data for testing. The split criteria we used in this work is gini impurity. Note that the best ML model for prediction will be chosen based on its accuracy, precision, f1 score, and error value. We tested various classifiers for prediction model selection, including random forest, bagging model, gradient boosting, stacking, XGBoost, decision tree, SVM, MLP, logistic regression, naïve bayes, and adaboost.

```
[51] X_train, X_test, y_train, y_test = train_test_split(X_selected, y_resampled, test_size=0.2

[52] X_train.shape, X_test.shape, y_train.shape, y_test.shape

((499, 10), (125, 10), (499,), (125,))
```

Fig. 16. Dataset splitting during model training and testing

Figure 19 depicts the confusion matrix values for all of the models we used to predict insurance fraud claims. The confusion matrix hints the classification performance of a model by indicating some values such as true positives or TP value, true negatives or TN value, false positives or FP value, and false negatives or FN value. Figure 19 shows that the stacking method has the highest true positive value and the lowest false positive value when compared to the other methods. The Stacking Model has the highest accuracy score (95%). The accuracy value of logistic regression is the lowest due to its lower true positive value (58) and higher false positive value (8).

Figure 20 depicts an evaluation of 12 machine learning models, including popular methods such as Random Forest, Gradient Boosting, Support Vector Machine, and Multilayer Perception in terms of accuracy value, precision value, recall value, MAE value (Mean Absolute Error), RMSE value (Root Mean Square Error), log loss, and ROC-AUC (Receiver Operating Characteristic - Area Under Curve). The Stacking model's success may be attributed to its ability to combine the characteristics of multiple base models, resulting in more robust and accurate forecasts. This method appears to have effectively addressed the flaws in individual models, resulting in improved overall performance across the analyzed criteria.

```
for model_name, (model, param_grid) in models.items():
    print(f"Training {model_name}...")
    skf = StratifiedKFold(n_splits=5)
    grid_search = GridSearchCV(estimator=model, param_grid=param_grid,
                               cv=skf, scoring='f1_weighted', n_jobs=-1)
    grid_search.fit(X_train, y_train)

    best_model = grid_search.best_estimator_
    best_params = grid_search.best_params_
    best_score = grid_search.best_score_

    print(f"Best parameters for {model_name}: {best_params}")
    print(f"Best cross-validation score for {model_name}: {best_score}")
    bestT[model_name] = best_model
    # Evaluate on test data
    y_pred = best_model.predict(X_test)
    print(f"F1-score on test data for {model_name}: {f1_score(y_test, y_pred, average='weighted')}")
    print(f"Classification report for {model_name}:\n{classification_report(y_test, y_pred)}")
    print(f"Confusion matrix for {model_name}:\n{confusion_matrix(y_test, y_pred)}")
```

Fig. 17. Applying GridSearchCV for Hyper-parameter Tuning

```
param_grid = {
    'n_estimators': [100, 200, 300],
    'max_features': ['auto', 'sqrt', 'log2'],
    'max_depth': [4, 6, 8, 10, 12],
    'criterion': ['gini', 'entropy']
}
grid_search = GridSearchCV(estimator=RandomForestClassifier(random_state=42), param_grid=param_grid, cv=5, n_jobs=-1, verbose=2)
grid_search.fit(X_train, y_train)
best_rf = grid_search.best_estimator_
y_pred_rf = best_rf.predict(X_test)

print("Best parameters for Random Forest:", grid_search.best_params_)
print(f"Random Forest Accuracy: {accuracy_score(y_test, y_pred_rf):.2f}")
print("Random Forest Confusion Matrix:")
print(confusion_matrix(y_test, y_pred_rf))
print("Random Forest Classification Report:")
print(classification_report(y_test, y_pred_rf))
```

```
Fitting 5 folds for each of 90 candidates, totalling 450 fits
/usr/local/lib/python3.10/dist-packages/sklearn/ensemble/_forest.py:424: FutureWarning: `max_features='auto'` has been deprecate
  warn(
Best parameters for Random Forest: {'criterion': 'gini', 'max_depth': 12, 'max_features': 'auto', 'n_estimators': 200}
```

Fig. 18. Best parameter selection (hyper parameter tuning for random forest classifier)

| Model | TP | FP | FN | TN |
|---|---|---|---|---|
| Random Forest | 66 | 2 | 4 | 53 |
| Bagging Model | 66 | 2 | 4 | 53 |
| Gradient Boosting | 66 | 4 | 4 | 51 |
| XGBoost | 62 | 1 | 8 | 54 |
| Stacking | 66 | 1 | 4 | 54 |
| Logistic Regression | 58 | 8 | 12 | 47 |
| Support Vector Machine | 62 | 5 | 8 | 50 |
| Naive Bayes | 65 | 9 | 5 | 46 |
| Multi-Layer Perceptron | 65 | 3 | 5 | 52 |
| Decision Tree Model | 63 | 2 | 7 | 53 |
| Stochastic Gradient Boosting | 66 | 2 | 4 | 53 |
| AdaBoost | 62 | 7 | 8 | 48 |

Fig. 19. TP, FP, FN, and TN entries of Confusion matrix for different ML models

| Model | Accuracy | Precision | Recall | F1 Score | MAE | RMSE | Log Loss | ROC-AUC |
|---|---|---|---|---|---|---|---|---|
| Random Forest | 0.95 | 0.97 | 0.94 | 0.96 | 0.05 | 0.22 | 1.73 | **0.988** |
| Bagging | 0.95 | 0.97 | 0.94 | 0.96 | 0.05 | 0.22 | 1.73 | 0.987 |
| Gradient Boosting | 0.94 | 0.94 | 0.94 | 0.94 | 0.06 | 0.25 | 2.31 | 0.985 |
| XGBoost | 0.93 | 0.98 | 0.89 | 0.93 | 0.07 | 0.27 | 2.60 | 0.982 |
| Stacking | **0.96** | **0.99** | **0.94** | **0.96** | **0.04** | **0.20** | **1.44** | 0.982 |
| Logistic Regression | 0.84 | 0.88 | 0.83 | 0.85 | 0.16 | 0.40 | 5.77 | 0.948 |
| Support Vector Machine | 0.90 | 0.93 | 0.89 | 0.91 | 0.10 | 0.32 | 3.75 | 0.975 |
| Naive Bayes | 0.89 | 0.88 | 0.93 | 0.90 | 0.11 | 0.33 | 4.04 | 0.955 |
| Multi-Layer Perceptron | 0.94 | 0.96 | 0.93 | 0.94 | 0.06 | 0.25 | 2.31 | 0.983 |
| Decision Tree | 0.93 | 0.97 | 0.90 | 0.93 | 0.07 | 0.27 | 2.60 | 0.932 |
| Stochastic Gradient Boosting | 0.95 | 0.97 | 0.94 | 0.96 | 0.05 | 0.22 | 1.73 | 0.986 |
| AdaBoost | 0.88 | 0.90 | 0.89 | 0.89 | 0.12 | 0.35 | 4.33 | 0.957 |

Fig. 20. Performance evaluation results of different ML models for property insurance fraud prediction

Accuracy is defined as the ratio of true values (TP+TN) to total sum of observations (TP+TN+FP+FN). Figure 20 shows that the stacking model has higher accuracy value, precision value, recall values, and f1 score values than the other ML methods tested. The Stacking Model has the highest accuracy score due to its preprocessing and hyper parameter tuning techniques. Precision value can be determined by taking the ratio of true positives (TP) and all positive instances (TP+FP). A high precision value indicates that the model consistently predicts positive outcomes.
A low precision value suggests that the model is likely to misclassify negative instances as positive, leading to many false positives. Figure 20 shows that the Stacking Model has the highest precision score of 0.99, with the XGBoost classifier coming in second with 0.98. The recall measures the model's ability to correctly identify positive events (TP/TP+FN). The figure shows that the stacking model has the highest recall value (.94), while the logistic regression model has the lowest (.83). The F1-score is a performance statistic that combines precision and recall to give a proper evaluation of a binary classification model. Accuracy, recall, precision, and F1 score can be computed as follows:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

$$F1\ score = 2 * \frac{(precision*recall)}{(precision+recall)} \tag{4}$$

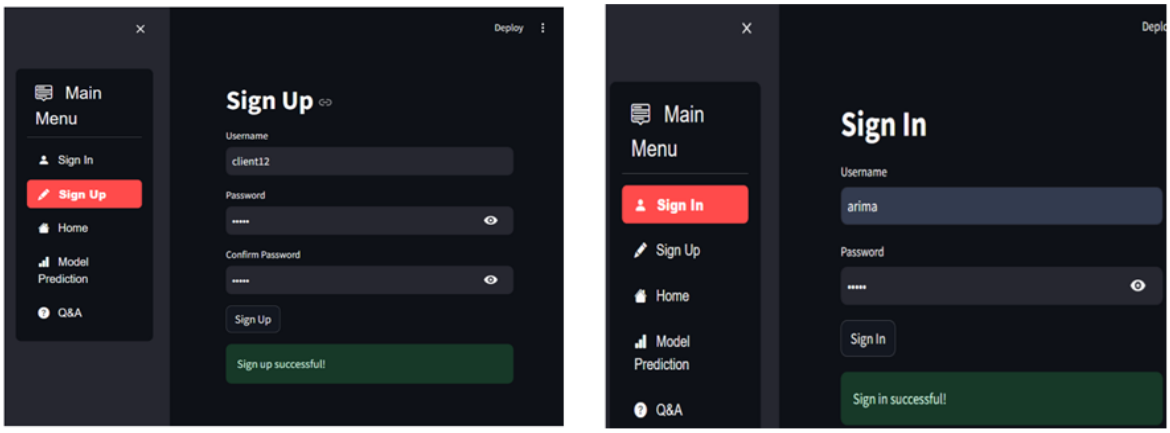
A high F1-score indicates that the model excels in both precision and recall, while a low F1-score indicates poor performance in one or both areas. Figure 20 shows that the Stacking Model has the highest F1 score (0.964), while the logistic regression model has the lowest F1 score (.85). The Random Forest Model has the highest ROC-AUC value (0.988), followed by the Stacking Model with 0.982.

Log Loss assesses the model's uncertainty by comparing projected probabilities p to actual binary results y. The Stacking Ensemble model has the lowest Log Loss Score (1.442), indicating that it is the best of the models we tested. RMSE (root mean squared error) calculates the average magnitude of errors between predicted probabilities (or scores) and actual binary outcomes (0 or 1). Figure 20 shows that the RMSE Score of the Stacking Ensemble model is 0.20, the lowest of all compared methods. Thus, the stacking ensemble model offers best RMSE value than the other ML models.

Figure 20 shows that the MAE score of the Stacking Ensemble model is the lowest (0.04), while that of logistic regression is the highest (.16). Thus, based on the evaluation results, we can conclude that the stacking ensemble model is superior for property insurance fraud prediction due to its higher accuracy and lower error values. Thus, in this work, the stacking ensemble model is used to predict property insurance fraud.

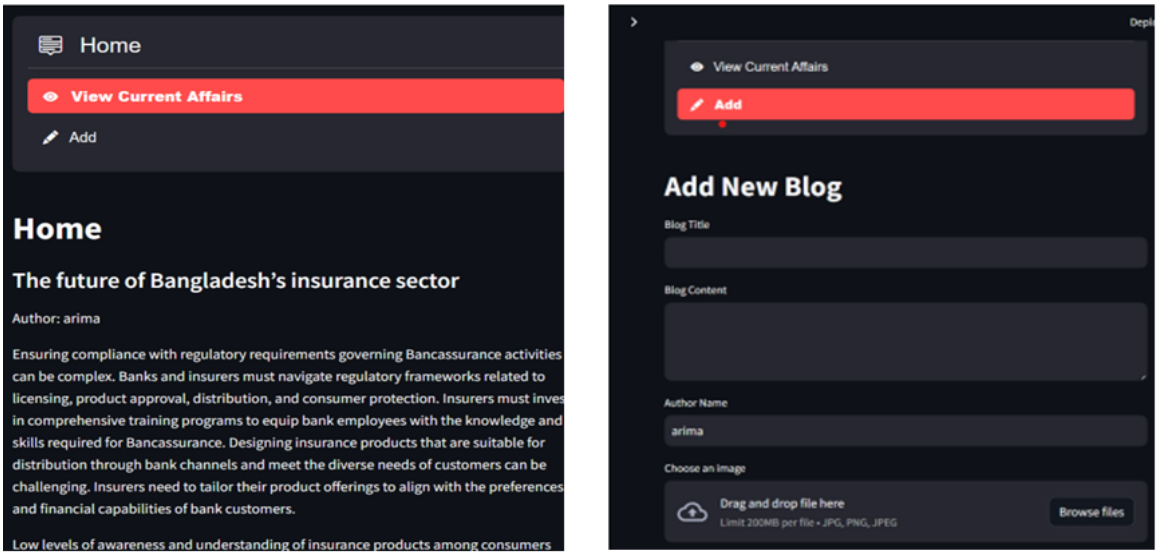
Table 1. Comparison with existing works

| Classifier | Accuracy | Recall | Method | Year |
|---|---|---|---|---|
| N. Akbar et al., [11] | 86% | 87% | Extreme gradient boosting | 2020 |
| M. Severino et al., [12] | 84.56% | 82.77% | Random forest | 2021 |
| M. Hanafy et al., [19] | 87% | 91.3% | Stochastic gradient boosting | 2021 |
| S. Pushpak et al., [25] | 92.6% | 91% | Quantum SVM | 2022 |
| Our proposed scheme | 96% | 94% | Stacking ensemble | 2024 |



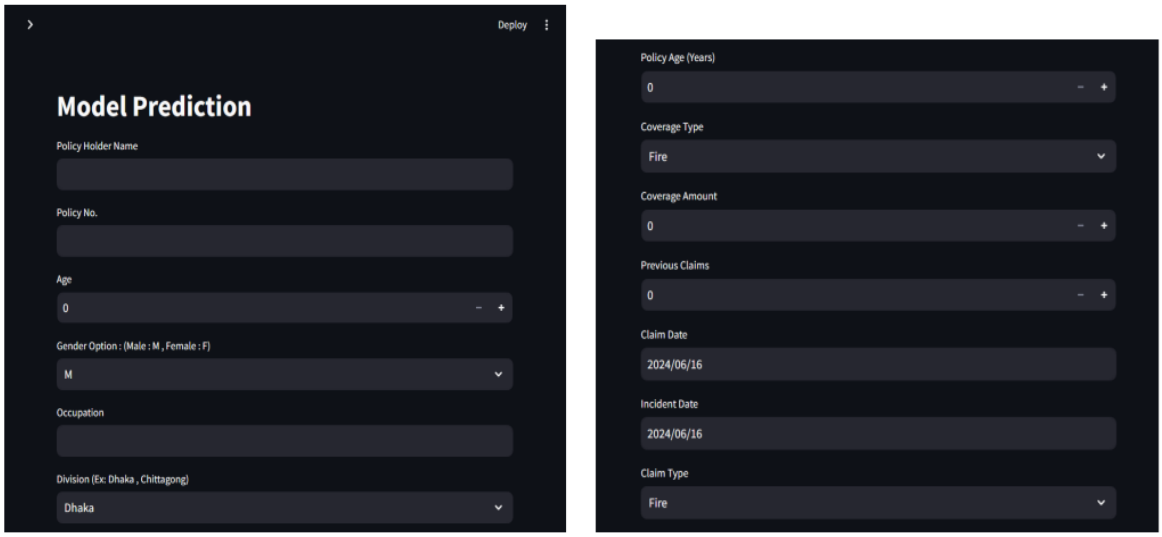

(a) (b)

Fig. 21. Sign up and sign in feature of our web app



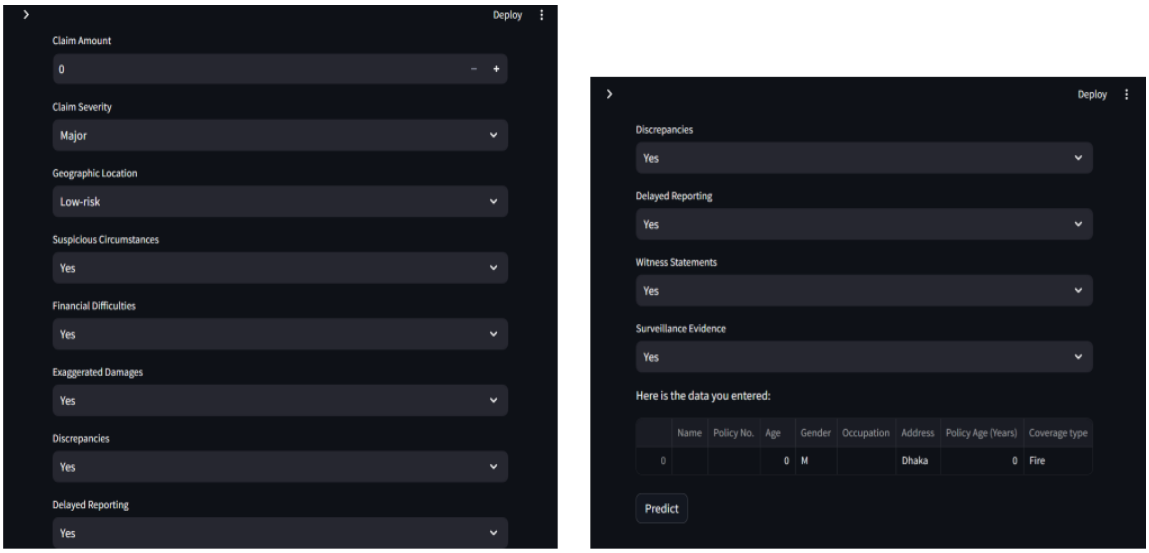

(a) (b)

Fig. 22. Home page and add new blog feature of our web app




(a) (b)

Fig. 23. Property insurance fraud prediction questions response screen




(a) (b)

Fig. 24. Property insurance fraud prediction questions response screen (cont.)

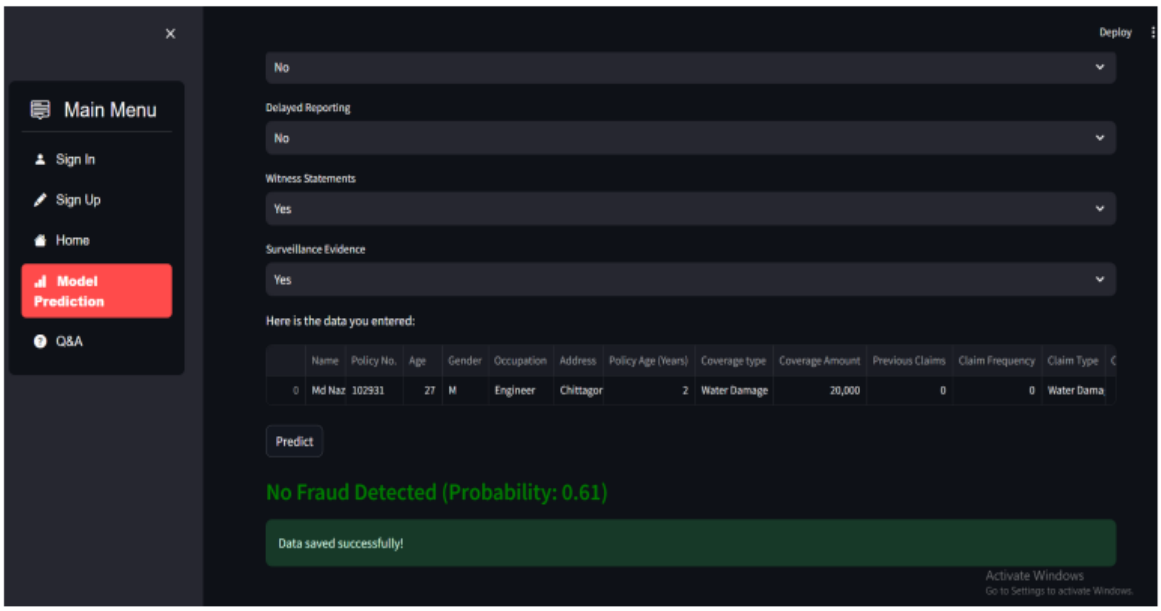
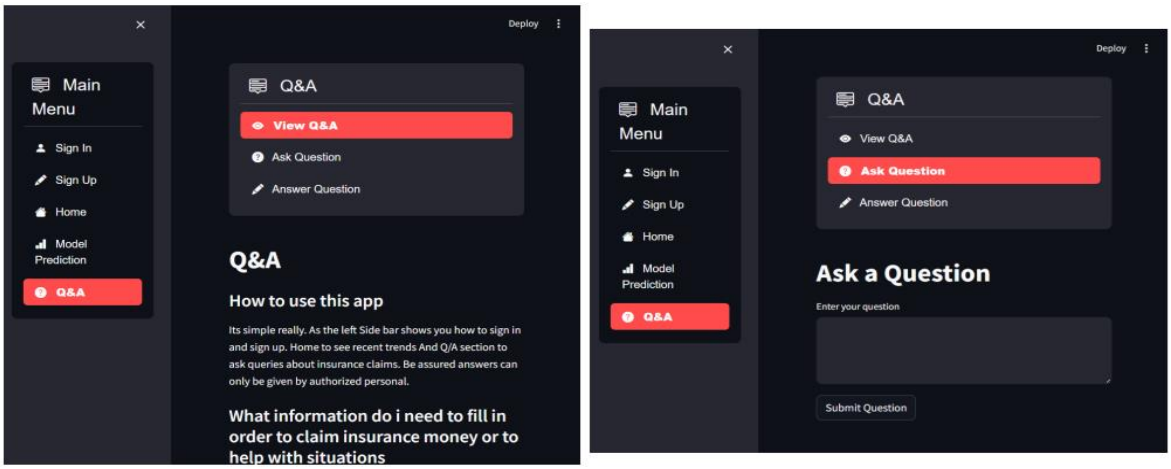



Fig. 25. ML based property insurance fraud prediction output screen using web app




(a) (b)

Fig. 26. Question and answer checking feature screen of our web app

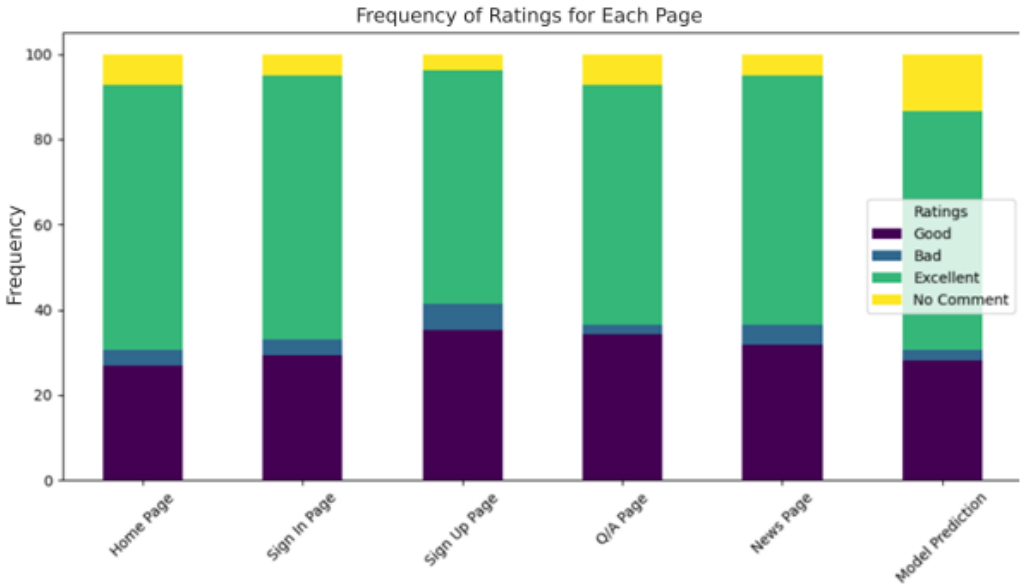



Fig. 27. Web app features evaluation results

### *3.6. Comparison with existing works*

To demonstrate the efficiency of our proposed system, we compared its accuracy and recall performance to some notable existing works (e.g., N. Akbar et al., [11], M. Severino et al., [12], and M. Hanafy et al., [19], S. Pushpak et al., [25]). Table 1 illustrates the comparison results. Table 1 hints that the proposed work has an accuracy and precision value of 96% and 94%, respectively. Table 1 depicts that the proposed stacking ensemble technique improves accuracy by 3.4% and recall by 2.7% compared to previous works. The existing works [25] secured the second best results; with 92.6 percent accuracy value and 91 percent recall value. The existing work [19] has the third best results, with 87 percent accuracy value and 91.3 percent recall value. The proposed stacking-based ensemble technique outperforms existing methods in terms of data imbalance handling, data preprocessing, hyper parameter tuning, and important feature selection techniques.

## 4. Web Application Development for Property Insurance Fraud Assistance

This paper presents a web application with features such as a machine learning-based insurance fraud prediction screen, a home page, a question submission and answer checking screen, a suggestion screen, blog post creation, and a login screen to assist customers and investors in detecting insurance fraud. The online or web application is created using Streamlit, a Python-based framework for developing interactive web apps. The front-end user interface is built with Streamlit, and the back-end functionality is written in Python. The application stores and retrieves data using MongoDB, a NoSQL database. We used the MongoDB platform's security features to address our data security concerns. This design facilitates efficient data management and seamless integration of the user interface and machine learning model for fraud detection. The MongoDB database is divided into four sections: users (which store user credentials), qa (which collects questions and answers), data (which stores claim data and predictions), and blogs. Data is saved and retrieved from the MongoDB database using Python's PyMongo package, ensuring that the application and database communicate seamlessly.

To sign up, the person must provide their username value and password value (see Figure 21(a)). After successfully creating an account, the user can sign in by following the steps outlined in Figure 21(b).

Now, verified users can navigate to the Home/News page to learn about current events in the insurance industry. Figure 22(a) depicts the home page, where users can access recent news about insurance fraud. If the person is authorized, they can use the figure 22(b) screen to add news or new blog posts. Users can also access the insurance fraud prediction page from the login page. Figures 23(a), 23(b), 24(a), 24(b), and 25(a) depict the input fields needed to predict potential fraud using client information. Figure 24 depicts the insurance fraud prediction outcome screen. The prediction result will be displayed in Figure 25 based on the answers provided by the user in figures 23 and 24.

The user can interact with the Q/A section and ask questions (refer to Figure 26(a)). Only authorized individuals or experts with access privileges can respond to the questions posed. Figure 26(b) shows how users can ask a question through the Q/A page. Authorized users can submit answers to questions using the figure 26(a) screen.

## 5. Evaluation Results

We evaluated the suitability of our property insurance fraud assistance web application based on user feedback and comments. Figure 27 depicts the recommender response results after testing various web app features. The survey asked participants to rate five key features of our web application: the Home Page, the Sign In Page, the Model Prediction Result Checker, the Sign Up Page, the Q/A Page, and the News Page. Ratings were based on a scale of Good, Bad, No Comment, and Excellent. This paper collected user responses through a Google questionnaire-based online response submission process. The figure shows that the proportion of recommenders with excellent, good, bad, or no comment for all features ranged from 54-64 percent, 25-28 percent, 1-4 percent, and 5-8 percent. According to the figures, up to 89 percent of users supported our web application features by providing good and excellent feedback.

## 6. Conclusion and Future Works

This article discussed a machine learning-based property insurance fraud prediction system using twelve machine learning classifiers and 23 features. This paper collected a real-time property insurance fraud incident dataset from an insurance company, then performed data preprocessing, cleaning, scaling, data imbalance reduction, hyper parameter tuning, and model selection using appropriate metrics. The stacking ensemble model is chosen as the best model for predicting property insurance fraud because it has a higher accuracy value of 96 percent, a higher recall value of 94 percent, a higher precision value of 99 percent, a higher f1 score value of 96 percent, a lower RMSE value of.20, and a lower MAE value of.04. Comparative results show that the proposed stacking ensemble technique improves accuracy and recall by 3.4% and 2.7%, respectively, over existing research. This article also contains a web application for users and investors that includes features such as machine learning-based property insurance fraud prediction, question submission, answer checking, login screen, suggestions and news checking, and new blog post submission. The web application evaluation results confirmed that at least 54 percent and up to 89 percent of users supported the web app's features with appropriate remarks.

In the future, some research challenges need to be addressed properly for the benefit of users and investors, such as the integration of large datasets from various sources (social media activity and financial transactions), different types of fraud activity prediction, and the use of adaptive learning methods for fraud trend and location detection. Future works also include the incorporation of AI and block chain-based systems for user data security and privacy preservation, further investigation of model over fitting issues, explainable AI-based fraud prediction reason explanation, insurance cost prediction using machine learning, and insurance churn prediction using generative AI and machine learning.

## References


[1] Peter Barrett et al., "RGA 2017 Global Claims Fraud Survey," *https://www.rgare.com/knowledge-center/article/rga-2017-global-claims-fraud-survey*, last accessed on June 2024.

[2] I. Mitic, "The Fraudster Next Door: Insurance Fraud Statistics, " *https://fortunly.com/statistics/insurance-fraud-statistics/*, last accessed on May 2024.

[3] M. Skiba, "Insurance fraud costs $309 billion a year – nearly $1,000 for every American," *https://theconversation.com/insurance-fraud-costs-309-billion-a-year-nearly-1-000-for-every-american-193087*, last accessed on July 2024.

[4] Wikipedia, "Insurance fraud," *https://en.wikipedia.org/wiki/Insurance_fraud*, last accessed on July 2024.

[5] Lugentic platform, "Fraud detection challenges insurers face," https://legentic.com/resources/5-fraud-detection-challenges-insurers-face, last accessed on June 2023.

[6] Statista, "Property Insurance-Bangladesh," *https://www.statista.com/outlook/fmo/insurances/non-life-insurances/property-insurance/bangladesh*, last accessed on may 2023.

[7] R. J. Bolton et al., "Statistical fraud detection: A review," *Statistical science*, vol. 17, no. 3, pp. 235–255, 2002.

[8] S. Viaene et al., "Strategies for detecting fraudulent claims in the automobile insurance industry," *European Journal of Operational Research*, vol. 176, no. 1, pp. 565–583, 2007.

[9] M. Artis et al., "Detection of automobile insurance fraud with discrete choice models and misclassified claims," *Journal of Risk and Insurance*, vol. 69, no. 3, pp. 325–340, 2002.

[10] S. Viaene et al., "Insurance fraud: Issues and challenges," *The Geneva Papers on Risk and Insurance-Issues and Practice*, vol. 29, no. 2, pp. 313–333, 2004.

[11] N. A. Akbar et al., 'Improvement of decision tree classifier accuracy for healthcare insurance fraud prediction by using extreme gradient boosting algorithm, " *in 2020 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, 2020, pp. 110–114.

[12] M. K. Severino et al., "Machine learning algorithms for fraud prediction in property insurance: Empirical evidence using real-world microdata, " Machine Learning with Applications, vol. 5, no. 1, pp. 1-14, 2021.

[13] T. Aziz et al., "Insurance fraud detection model: Using machine learning techniques, " *International Journal of Computational Intelligence in Control*, vol. 14, no. 1, pp. 410–422, 2022.

[14] L. Rukhsar et al., "Prediction of insurance fraud detection using machine learning algorithms", *Mehran University Research Journal of Engineering and Technology*, vol. 41, pp. 33–40, Jan. 2022.

[15] B. Itri et al., "Performance comparative study of machine learning algorithms for automobile insurance fraud detection, " *Third International Conference on Intelligent Computing in Data Sciences (ICDS)*, Marrakech, Morocco, 2019, pp. 1-4.
[16] S. Harjai et al., "Detecting fraudulent insurance claims using random forests and synthetic minority oversampling technique," in 4th International Conference on Information Systems and Computer Networks (ISCON), 2019, pp. 123–128.
[17] N. V. Chawla et al., 'Smote: Synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.
[18] S. Subudhi et al., "Use of optimized fuzzy c-means clustering and supervised classifiers for automobile insurance fraud detection," *Journal of King Saud University - Computer and Information Sciences*, vol. 32, no. 5, 2020, pp. 568-575.
[19] M. Hanafy et al., "Using machine learning models to compare various resampling methods in predicting insurance fraud," *Journal of Theoretical and Applied Information Technology*, vol. 99, pp. 2819–2833, Jul. 2021.
[20] Shanthini M. et al., "Stacking Classifier-based Automated Insurance Fraud Detection System," *IEEE Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, Gwalior, India, 2022, pp. 1-6.
[21] Snowflake Inc., "Streamlit documentation," *https://docs.streamlit.io/*, last accessed on June 2024.
[22] Wikipedia, "MongoDB," *https://en.wikipedia.org/wiki/MongoDB*, last accessed on July 2024.
[23] Pymongo, "Pymongo installation," *https://pypi.org/project/pymongo/*, last accessed on June 2024.
[24] Veena K et al., "Predicting health insurance claim frauds using supervised machine learning technique," *Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, Chennai, India, 2023, pp. 1-7.
[25] S. N. Pushpak et al., "An Implementation of Quantum Machine Learning Technique to Determine Insurance Claim Fraud," *10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Noida, India, 2022, pp. 1-5.

## Authors' Profiles

**Kazi Md. Tawsif Rahman** is a graduate student at the computer science and engineering department, Chittagong University of Engineering and Technology. His major research interests include machine learning, deep learning, and web app development. He finished his bachelor of computer science degree in 2024.

**Chowdhury Mahfuzul Hoq** obtained his PhD in 2018. He is currently a faculty member at computer science and engineering department, Chittagong University of Engineering and Technology. His major research interest include machine learning, cloud computing, communication network, and mobile app development. He has published several journal and conference articles in reputated publishers, IEEE journals, and conferences.

**How to cite this paper:** Kazi Md. Tawsif Rahman, Chowdhury Mahfuzul Hoq, "An Automated System for Detecting Property Insurance Fraud Using Machine Learning", International Journal of Mathematical Sciences and Computing(IJMSC), Vol.10, No.3, pp. 8-25, 2024. DOI: 10.5815/ijmsc.2024.03.02