

ENHANCING INSURANCE CLAIM INTEGRITY USING MACHINE LEARNING FOR FRAUD DETECTION AND ANALYSIS

*Major project report submitted
in partial fulfillment of the requirement for award of the degree of*

**Bachelor of Technology
in
Computer Science & Engineering**

By

M.V.Vishnuvardhan Reddy (20UECS0603) (VTU16707)
Y.Karun Kumar (20UECS1037) (VTU18059)
Y.Ramana Reddy (20UECS1046) (VTU16717)

*Under the guidance of
Dr.S.LALITHA,B.Tech.,M.E.,Ph.D.,
ASSOCIATE PROFESSOR*



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING**

**VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF
SCIENCE & TECHNOLOGY**

(Deemed to be University Estd u/s 3 of UGC Act, 1956)

**Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA**

May, 2024

ENHANCING INSURANCE CLAIM INTEGRITY USING MACHINE LEARNING FOR FRAUD DETECTION AND ANALYSIS

*Major project report submitted
in partial fulfillment of the requirement for award of the degree of*

**Bachelor of Technology
in
Computer Science & Engineering**

By

**M.V.Vishnuvardhan Reddy (20UECS0603) (VTU16707)
Y.Karun Kumar (20UECS1037) (VTU18059)
Y.Ramana Reddy (20UECS1046) (VTU16717)**

*Under the guidance of
Dr.S.LALITHA ,B.Tech.,M.E.,Ph.D.,
ASSOCIATE PROFESSOR*



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING**

**VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF
SCIENCE & TECHNOLOGY**

(Deemed to be University Estd u/s 3 of UGC Act, 1956)

**Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA**

May, 2024

CERTIFICATE

It is certified that the work contained in the project report titled “ENHANCING INSURANCE CLAIM INTEGRITY USING MACHINE LEARNING FOR FRAUD DETECTION AND ANALYSIS” by “M.V.Vishnuvardhan Reddy (20UECS0603), Y.Karun Kumar (20UECS1037), Y.Ramana Reddy (20UECS1046)” has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2024

Signature of the Professor In-charge
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2024

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

M.V.Vishnuvardhan Reddy

Date: / /

Y.Karun Kumar

Date: / /

Y.Ramana Reddy

Date: / /

APPROVAL SHEET

This project report entitled “ENHANCING INSURANCE CLAIM INTEGRITY USING MACHINE LEARNING FOR FRAUD DETECTION AND ANALYSIS” by “ M.V.Vishnuvardhan Reddy (20UECS0603), Y.Karun Kumar (20UECS1037), Y.Ramana Reddy (20UECS1046)” is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners

Supervisor

Dr.S.Lalitha,B.Tech., M.E., Ph.D.,
ASSOCIATE PROFESSOR

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN**, for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.**, for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Head, Department of Computer Science & Engineering, Dr.M.S. MURALI DHAR, M.E., Ph.D.**, for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our Internal Supervisor **Dr. S. LALITHA, B.Tech., M.E., Ph.D.**, for her cordial support, valuable information and guidance, she helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. C. SHYAMALA KUMARI, M.E.**, for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

M.V.Vishnuvardhan Reddy	(20UECS0603)
Y.Karun Kumar	(20UECS1037)
Y.Ramana Reddy	(20UECS1046)

ABSTRACT

Insurance Company working as commercial enterprise from last few years have been experiencing fraud cases for all type of claims. Amount claimed by fraudulent is significantly huge that may causes serious problems, hence along with government, different organization also working to detect and reduce such activities. Such frauds occurred in all areas of insurance claim with high severity such as insurance claimed towards auto sector is fraud that widely claimed and prominent type, which can be done by fake accident claim. Fraudulent incidents permeate all areas of insurance claims, with varying degrees of severity observed across different sectors. Notably, the automobile insurance sector stands out as a prominent target for fraudsters due to the prevalence of fraudulent claims, particularly those stemming from fabricated accidents. These fraudulent practices undermine the integrity of insurance claim processes and necessitate the development of robust systems to address them effectively. In response to the escalating threat of insurance fraud, the aim of this project is to develop a comprehensive system capable of analyzing vast datasets of insurance claims to detect fraudulent and fake claims. Leveraging machine learning algorithms such as decision tree, support vector machine, logistic regression, naive bayes, stochastic gradient descent algorithms, the project seeks to construct models that can accurately label and classify claims based on their fraudulent nature. The project aims to empower insurance companies with the tools necessary to proactively identify suspicious claims and mitigate potential losses. The experimental results demonstrated for the Decision Tree Classifier accuracy is (95.55 percent), Naive Bayes accuracy (88 percent), Support Vector Machine accuracy is (94 percent), Logistic Regression accuracy is (94 percent) making it a promising approach for insurance claim fraud detection.

Keywords: Automobile Insurance, Decision Tree, Fraud Detection, Naive Bayes, Machine Learning, Support Vector Machine, Stochastic Gradient Descent.

LIST OF FIGURES

4.1	Architecture Diagram	11
4.2	Data Flow Diagram	12
4.3	Use Case Diagram	13
4.4	Class Diagram	14
4.5	Sequence Diagram	15
4.6	Activity Diagram	16
5.1	Insurance Fraud Detection	21
5.2	Prediction of Claim	22
5.3	Test Image	26
6.1	Accuracy of The Proposed System	27
6.2	Comparsion	29
6.3	Fraud Claim	31
6.4	Real Claim	32
9.1	Poster	41

LIST OF ACRONYMS AND ABBREVIATIONS

abbr	Abbreviation
CLI	Command Line Interface
EDA	Exploratory Data Analysis
FCD	Fraudulent Claims Detection
FCP	Fraudulent Claims Prevention
HT	Hyperparameter Tuning
ICF	Internal Code Flow
IDE	Integrated Development Environment
MD	Model Deployment
ML	Machine Learning
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
VSC	Visual Studio Code

TABLE OF CONTENTS

	Page.No
ABSTRACT	v
LIST OF FIGURES	vi
LIST OF ACRONYMS AND ABBREVIATIONS	vii
1 INTRODUCTION	1
1.1 Introduction	1
1.2 Aim of the Project	2
1.3 Project Domain	2
1.4 Scope of the Project	3
2 LITERATURE REVIEW	4
3 PROJECT DESCRIPTION	7
3.1 Existing System	7
3.2 Proposed System	7
3.3 Feasibility Study	9
3.3.1 Economic Feasibility	9
3.3.2 Technical Feasibility	9
3.3.3 Social Feasibility	9
3.4 System Specification	10
3.4.1 Hardware Specification	10
3.4.2 Software Specification	10
3.4.3 Standards and Policies	10
4 METHODOLOGY	11
4.1 General Architecture	11
4.2 Design Phase	12
4.2.1 Data Flow Diagram	12
4.2.2 Use Case Diagram	13
4.2.3 Class Diagram	14

4.2.4	Sequence Diagram	15
4.2.5	Activity Diagram	16
4.3	Algorithm & Pseudo Code	17
4.3.1	Algorithm	17
4.3.2	Pseudo Code	17
4.4	Module Description	17
4.4.1	Data Collection And Pre-processing	17
4.4.2	Feature Selection and Splitting the Data	18
4.4.3	Model Selection and Training	18
4.5	Steps to execute/run/implement the project	18
4.5.1	Step1: Data Collection And Pre-processing	18
4.5.2	Step2 : Feature Selection and Splitting the Data	19
4.5.3	Step3 : Model Selection and Training	19
5	IMPLEMENTATION AND TESTING	20
5.1	Input and Output	20
5.1.1	Input Design	20
5.1.2	Output Design	21
5.2	Types of Testing	23
5.2.1	Unit Testing	23
5.2.2	Integration Testing	24
5.2.3	System Testing	25
5.2.4	Test Result	26
6	RESULTS AND DISCUSSIONS	27
6.1	Efficiency of the Proposed System	27
6.2	Comparison of Existing and Proposed System	28
6.3	Sample Code	29
7	CONCLUSION AND FUTURE ENHANCEMENTS	33
7.1	Conclusion	33
7.2	Future Enhancements	34
8	PLAGIARISM REPORT	35

9	SOURCE CODE & POSTER PRESENTATION	36
9.1	Source Code	36
9.2	Poster Presentation	41
	References	41

Chapter 1

INTRODUCTION

1.1 Introduction

Insurance fraud presents a significant challenge for insurance companies worldwide, resulting in financial losses and higher premiums. Detecting and preventing fraudulent claims is essential for industry integrity and fairness. Traditional methods rely on manual investigation and rule-based systems, limiting adaptability. Machine learning (ML) is a powerful tool for enhancing fraud detection in insurance claims. By analyzing vast datasets, ML algorithms can identify patterns indicative of fraudulent activity. This enables insurance companies to detect fraud more effectively and proactively address complex schemes.

One of the key challenges in insurance fraud detection is the imbalance between fraudulent and legitimate claims. Fraudulent claims often represent only a small fraction of the total claims processed by insurance companies, making them a minority class in the dataset. Traditional machine learning algorithms may struggle to effectively learn from imbalanced data, leading to suboptimal performance in detecting fraudulent activity. Techniques such as oversampling, undersampling, and cost-sensitive learning can help address this imbalance and improve the performance of fraud detection models.

Feature engineering plays a crucial role in the success of machine learning models for insurance fraud detection. By selecting and transforming relevant features from the data, such as claim history, policy details, and claimant demographics, insurers can provide valuable information to the machine learning algorithms, enabling them to better discriminate between legitimate and fraudulent claims. Additionally, feature selection techniques can help reduce the dimensionality of the data and improve the efficiency of the fraud detection process. In this project, the application of machine learning in fraud detection and analysis for insurance claims is explored. Various ML algorithms and techniques are discussed that can be employed to detect fraudu-

lent claims, including supervised learning and anomaly detection. Furthermore, the project delves into feature engineering, model evaluation, and deployment strategies tailored specifically to the insurance fraud detection domain.

1.2 Aim of the Project

The aim of the project is to use machine learning to detect and reduce fraudulent insurance claims, particularly in the auto sector, by analyzing a dataset and implementing classification algorithms.

1.3 Project Domain

The project domain involves ML algorithms and data analytics techniques to detect and prevent fraudulent activities in insurance claims. In the world of insurance, sometimes people try to trick the system by making false claims to get money. Detecting these fake claims is important for insurance companies to keep things fair and stop from losing money. Previously, insurance companies relied on manual checks and basic rules to catch fraud, but now insurance companies are turning to something called machine learning. Machine learning is a fancy term for teaching computers to learn from data and spot patterns. By looking at lots of past claims, computer programs can learn what real claims look like and spot the ones that seem fishy. Computer programs can also look for unusual things that might suggest someone is trying to cheat the system. This helps insurance companies to stay one step ahead of the fraudsters and protect themselves and their customers. To make computer programs work well, it's important to give them the right information to look at. This includes details about the claim, the person making it, and any history of previous claims. By choosing the most relevant information, the computer can get better at telling the difference between honest claims and dishonest ones. Sometimes, computer programs can even understand written descriptions to pick up on clues that might suggest someone is lying. By combining smart computer programs with human knowledge, insurance companies can fight fraud more effectively, saving money and keeping the system fair for everyone.

1.4 Scope of the Project

The main purpose of this project is to develop a robust machine learning model for fraud detection and analysis in the insurance industry. The model will be designed to accurately identify fraudulent insurance claims, thus enabling insurance companies to mitigate financial losses and maintain the integrity of their operations. To achieve this goal, the project will involve several key steps and considerations:

Data Collection and Preparation: Gathering relevant datasets containing historical insurance claims, including both fraudulent and legitimate cases. Preprocessing the data to handle missing values, outliers, and inconsistencies, and ensuring data quality and integrity.

Exploratory Data Analysis (EDA): Conducting comprehensive exploratory data analysis to gain insights into the characteristics and patterns of fraudulent claims. This involves visualizing the data, identifying trends, correlations, and anomalies that may indicate fraudulent behavior.

Algorithm Selection and Testing: Evaluating multiple machine learning algorithms to identify the most suitable ones for fraud detection in the insurance domain. This may include supervised learning algorithms such as logistic regression, decision trees, random forests, and gradient boosting, as well as unsupervised learning techniques like clustering and anomaly detection.

Model Development and Training: Developing and training machine learning models using the selected algorithms on the prepared dataset. Fine-tuning the models and optimizing hyperparameters to improve performance and accuracy.

Deployment and Integration: Integrating the trained models into existing insurance systems and workflows to facilitate real-time fraud detection. Developing user-friendly interfaces and APIs for seamless integration with claims processing platforms, enabling efficient decision-making by claims adjusters and investigators.

By following these steps and considerations, the project aims to deliver a robust and effective machine learning model for fraud detection and analysis in the insurance industry. This model will empower insurance companies to proactively identify and mitigate fraudulent activities, thereby protecting their financial interests and maintaining trust among policyholders.

Chapter 2

LITERATURE REVIEW

[1]Adedayo, et al, (2022) focused on utilizing machine learning algorithms to predict fraud in automobile insurance claims. It presents an in-depth exploration of various machine learning techniques and their application in detecting fraudulent activities. The study aims to enhance prediction accuracy and mitigate financial risks for insurers by developing effective fraud detection models.

[2]Caruana, et al, (2022) explored various methods and technologies employed in detecting fraudulent activities within the automobile insurance sector. The authors likely discuss traditional approaches like manual checks and basic rules, as well as advanced techniques such as machine learning. They examine the challenges faced by insurance companies in detecting fraud and evaluate the effectiveness of different fraud detection strategies. Overall, the article provides valuable insights into the current landscape of automobile insurance fraud detection, offering a foundation for further research and practical application in the field.

[3]Elsevier , et al, (2022) investigated how artificial intelligence and machine learning can improve insurance fraud detection. It examines data to find patterns and anomalies indicating fraud in insurance claims. By comparing different AI and ML techniques, the study aims to enhance fraud detection in the insurance industry, providing valuable insights for insurers and policymakers.

[4]Jenita Mary, et al, (2022) explored the application of machine learning classifiers in detecting fraud within healthcare insurance claims. The authors investigate various classifiers and evaluate their effectiveness in identifying fraudulent activities within the healthcare insurance domain. The authors probably present insights into the effectiveness of different classifiers in detecting fraud, comparing their performance and accuracy. Through their study, they likely aim to contribute to the development of more robust fraud detection systems.

[5]Kapadiya, et al, (2023) explored the utilization of blockchain and artificial intelligence (AI) to enhance fraud detection in healthcare insurance. It likely provides an analysis of the current landscape of healthcare insurance fraud, highlighting the shortcomings of traditional detection methods. The authors propose an integrated architecture that combines blockchain technology for secure data storage with AI algorithms for fraud pattern analysis. The article may discuss the technical aspects of this approach and its potential benefits, such as improved accuracy and scalability.

[6]Mary Arockiam,et al, (2022) introduced an innovative approach to fraud detection in the healthcare insurance industry through the utilization of MapReduce-Iterative Support Vector Machine (SVM) classifiers. The study presents a novel system that merges the scalability of MapReduce with the precision of SVM classifiers to efficiently identify fraudulent activities in healthcare insurance claims. By incorporating advanced data processing techniques and machine learning algorithms, the proposed system aims to enhance fraud detection accuracy, thereby bolstering the effectiveness and dependability of healthcare insurance operations. The research contributes to the advancement of robust fraud detection mechanisms tailored specifically for the healthcare insurance domain, addressing the escalating demand for proficient fraud prevention strategies in this sector.

[7]Mark Anthony, et al, (2023) presented a comprehensive overview of fraud detection methods in the automobile insurance industry. The authors likely discuss various approaches and technologies utilized for identifying fraudulent activities within insurance claims. Traditional methods such as manual checks and rule-based systems may be compared with more advanced techniques like machine learning and data analytics. Through their examination, they likely aim to offer a comprehensive overview of the strengths and limitations of each approach, along with recommendations for improving fraud detection effectiveness in the context of automobile insurance.

[8]Nabrawi, et al, (2023) demonstrated various machine learning algorithms and methodologies used to analyze healthcare insurance data and identify fraudulent patterns. They may discuss the importance of fraud detection in healthcare insurance and the challenges associated with traditional detection methods. The article may present case studies or experiments demonstrating the effectiveness of machine learning in detecting fraudulent claims. Additionally, it may discuss considerations such as data preprocessing, feature selection, and model evaluation in the context of fraud detection. Overall, the article likely provides valuable insights into leveraging machine learning for improving fraud detection in healthcare insurance, contributing to the advancement of analytics in the healthcare industry.

[9]Rima Kaanfarania, et al, (2021) introduced an innovative method for detecting health insurance fraud using blockchain technology and smart contracts. The authors propose an adaptive decision-making approach to select the most suitable blockchain platform for this purpose. They detail the criteria for platform evaluation and the decision-making algorithms utilized. Additionally, the implementation of smart contracts to automate fraud detection processes is discussed. The authors may present a framework or methodology that incorporates factors such as scalability, security, interoperability, and performance to guide the selection process. Their adaptive approach likely takes into account the evolving nature of blockchain technology and the specific requirements of health insurance systems. Through their research, they likely aim to contribute to the development of more effective and efficient solutions for leveraging blockchain technology in the healthcare insurance domain.

[10]Yaohao Peng,et al, (2023) presented an exploration of machine learning algorithms tailored for predicting fraud in property insurance. It discusses various techniques and advancements in data analysis within the insurance industry, focusing on the application of machine learning models. The paper highlights the challenges of fraud detection in property insurance and proposes innovative approaches to enhance prediction accuracy.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

The existing insurance fraud detection system, utilizing the K-means algorithm, has demonstrated a commendable accuracy level of 80 percent, signifying its efficacy in detecting potential fraudulent activities within insurance data. This achievement underscores the relevance of clustering techniques in identifying patterns indicative of fraud. However, ongoing advancements in machine learning algorithms, coupled with the evolving nature of fraudulent behaviors, necessitate a continual reassessment of strategies to further enhance detection capabilities. Additionally, exploring complementary approaches such as ensemble methods or incorporating more sophisticated feature engineering could offer avenues for refining the system's performance and staying ahead of emerging fraud tactics. Machine learning is usually abbreviated as metric capacity unit. The study of machine learning includes computers with the implicit capability to be trained whereas not being expressly programmed. This capacity unit focuses on the expansion of pc programs that has enough capability to alter, that square measure once unprotected to the new information. Metric capacity unit algorithms square measure generally classified into 3 main divisions that square measure supervised learning, unattended learning and reinforcement learning. Data processing a neighborhood of machine learning has advanced considerably within the current years.

Disadvantages : The system is not implemented Convex-NMF based Supervised Spammer Detection with Social Interaction (CNMFSD). The system is not implemented any ml classifier for test and train the datasets.

3.2 Proposed System

The influence of the feature engineering, feature choice parameter modification area unit explored with an aim of achieving superior prophetic performance with su-

perior accuracy. The assorted machine learning techniques area unit utilized in the development of accuracy of detection in unbalanced samples. As a system, the info are divided into 3 completely different segments. These area unit loosely coaching, testing and validation. The algorithmic program is trained on partial set of knowledge and parameters. These area unit later changed on a validation set. This may be studied for evaluation and performance on the particular testing dataset. The high acting models area unit formerly tested with numerous random splits of knowledge. This helps to confirm the consistency in results the approach discussed above comprises of three layers.

Data Pre-processing step: In this step, the data is ready in order that are often employed in code with efficiency. Extraction of the dependent and freelance variables from the given dataset. Then the dataset is split as coaching and checking victimisation train test split module from sklearn library. Feature scaling is completed therefore on get correct results of predictions.

Data Cleaning: Handle missing values, outliers, and inconsistencies in the data. Ensure data quality before proceeding to the next steps.

Feature Engineering: Extract relevant features from the raw data that could help in distinguishing between legitimate and fraudulent claims. This could include variables like claim amount, claim type, policyholder information, claim history, etc.

Data Transformation: Normalize or scale the features to ensure they are on similar scales. Convert categorical variables into numerical representations through techniques like one-hot encoding or label encoding.

Feature Selection: Choose the most relevant features for detecting insurance fraud. Use techniques like correlation analysis, feature importance ranking, or domain expertise to select the subset of features that are likely to contribute most to the model's performance.

Splitting the Data: Divide the dataset into training, validation, and test sets. Typically, the data is split into around 70-80 percent for training, 10-15 percent for validation, and 10-15 percent for testing. Ensure that each set contains a representative sample of both legitimate and fraudulent claims.

Advantages : Different models are tested on the dataset once it is obtained and cleaned. On the basis of the initial model performance, different features of the model are engineered and tested again. Once all the options area unit designed, the model is made and run victimisation completely different completely different values and victimisation different iteration procedures.

3.3 Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- **ECONOMICAL FEASIBILITY**
- **TECHNICAL FEASIBILITY**
- **SOCIAL FEASIBILITY**

3.3.1 Economic Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.3.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.3.3 Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed

to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.4 System Specification

3.4.1 Hardware Specification

- Processor - Pentium –IV
- RAM - 4 GB (min)
- Hard Disk - 20 GB
- Key Board - Standard Windows Keyboard

3.4.2 Software Specification

- Operating system : Windows 7 Ultimate
- Coding Language : Python
- Back-End : Django-ORM
- Designing : Html, css, javascript
- Data Base : MySQL (WAMP Server)

3.4.3 Standards and Policies

Visual Studio

Visual Studio Code (VSC) is a versatile and lightweight code editor developed by Microsoft. It offers a command line interface (CLI) alongside its graphical user interface (UI), making it accessible across Windows, Linux, and MacOS platforms. VSC supports various programming languages and extensions, including those for Machine Learning (ML) development. It provides an Integrated Development Environment (IDE) that enhances coding productivity and ease of use.

Standard Used: ISO/IEC 27001

Chapter 4

METHODOLOGY

4.1 General Architecture

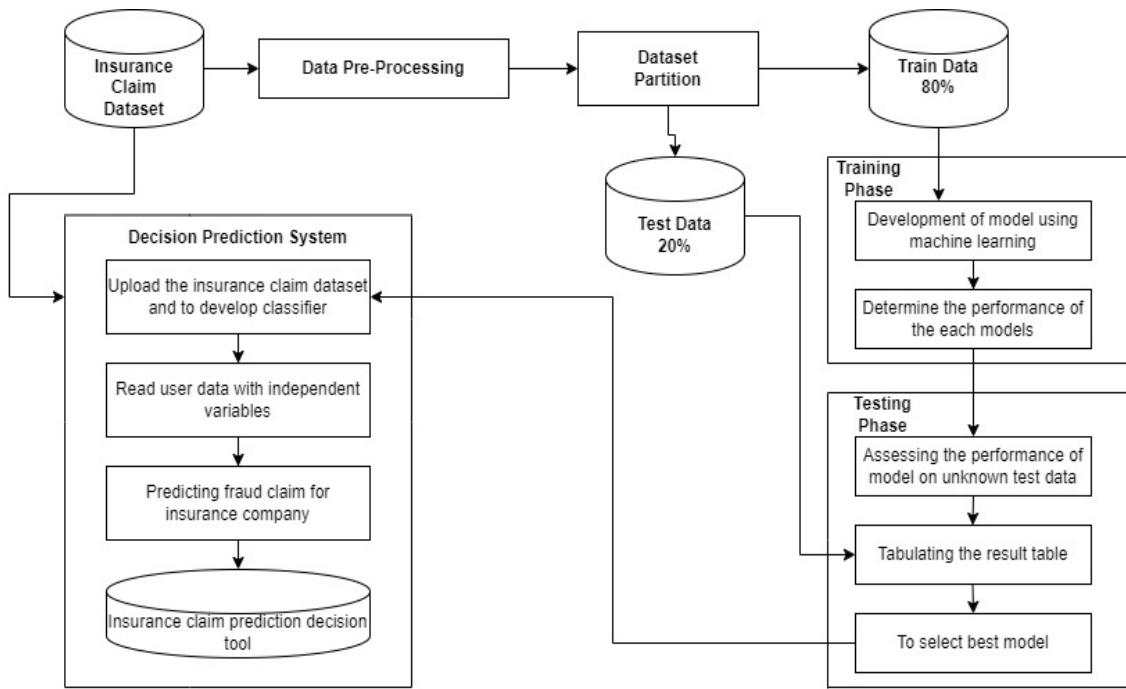


Figure 4.1: **Architecture Diagram**

In figure 4.1: explained that the architecture diagram for fraud detection and analysis in insurance claims, data collection involves gathering insurance claim-related data from diverse sources, which is then preprocessed to handle missing values, outliers, and categorical variables. Subsequently, feature engineering selects and creates meaningful features from the data to enhance model performance. With the preprocessed and engineered features, machine learning models are trained using various algorithms such as logistic regression, random forest, or neural networks, leveraging labeled data to learn patterns indicative of fraudulent claims. Through this iterative process, the models continuously improve their ability to accurately detect and analyze fraudulent activities within insurance claims, contributing to the overall effectiveness of the fraud detection system.

4.2 Design Phase

4.2.1 Data Flow Diagram

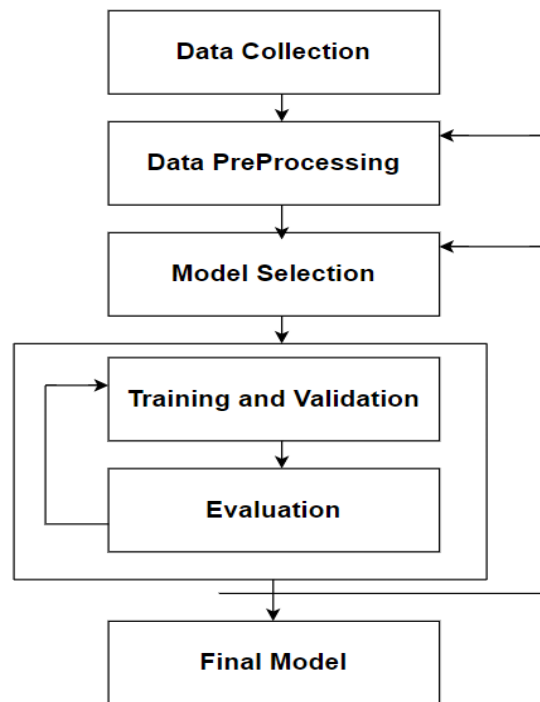


Figure 4.2: Data Flow Diagram

In figure 4.2 : explained that the data flow diagram for fraud detection and analysis in insurance claims, data flows through several stages. Initially, raw data related to insurance claims is collected from various sources and enters the system. This data undergoes preprocessing steps such as cleaning, normalization, and feature extraction to prepare it for analysis. Processed data is then fed into machine learning algorithms for training and model development. The trained models analyze incoming insurance claims data in real-time to detect potential fraudulent activities. Detected fraudulent claims are flagged for further investigation or mitigation, while non-fraudulent claims proceed through the system. Additionally, feedback loops may exist where outcomes of detected fraud cases are incorporated back into the system to improve future fraud detection capabilities. The data flow diagram illustrates the journey of insurance claims data through different stages of processing and analysis within the fraud detection system.

4.2.2 Use Case Diagram

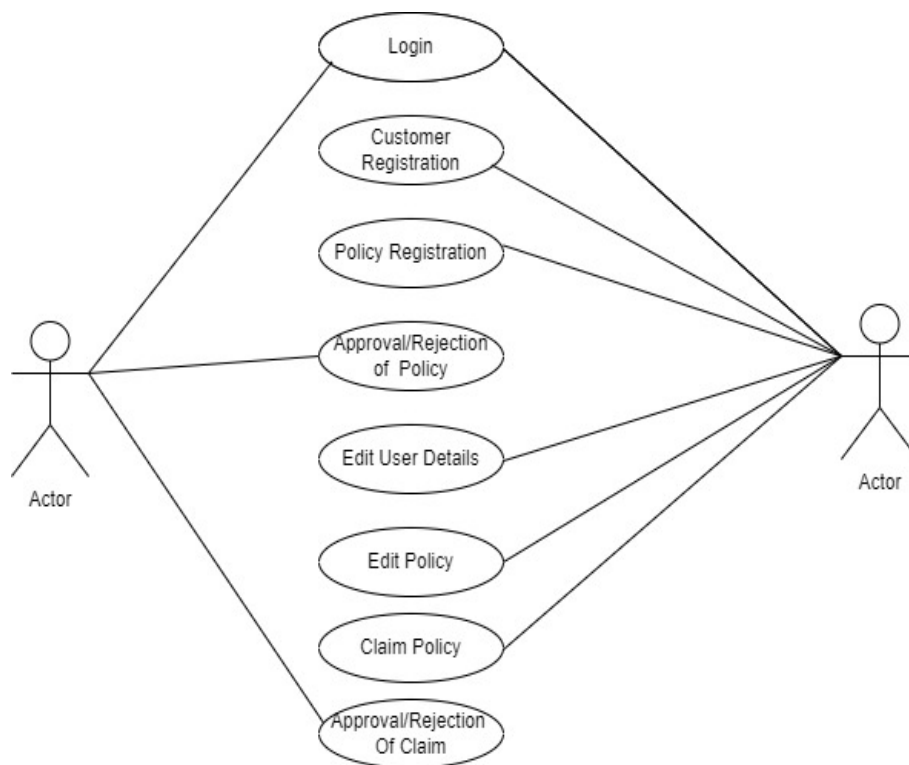


Figure 4.3: Use Case Diagram

In figure 4.3 : explained that the use case diagram for fraud detection and analysis in insurance claims, actors such as the insurance claimant, investigator, and system administrator are depicted along with their interactions with the system. The use cases include “File Insurance Claim” initiated by the claimant, ”Analyze Claim” for processing and fraud detection using machine learning algorithms, “Investigate Fraud” for further investigation if fraudulent activity is detected, and “Update System” managed by the administrator for system maintenance and enhancements. Associations between actors and use cases illustrate their respective roles and interactions within the system, while include and extend relationships highlight additional functionalities within certain use cases. The system boundary encapsulates all components, delineating the scope of the fraud detection system and providing a comprehensive overview of its functionalities and interactions.

4.2.3 Class Diagram

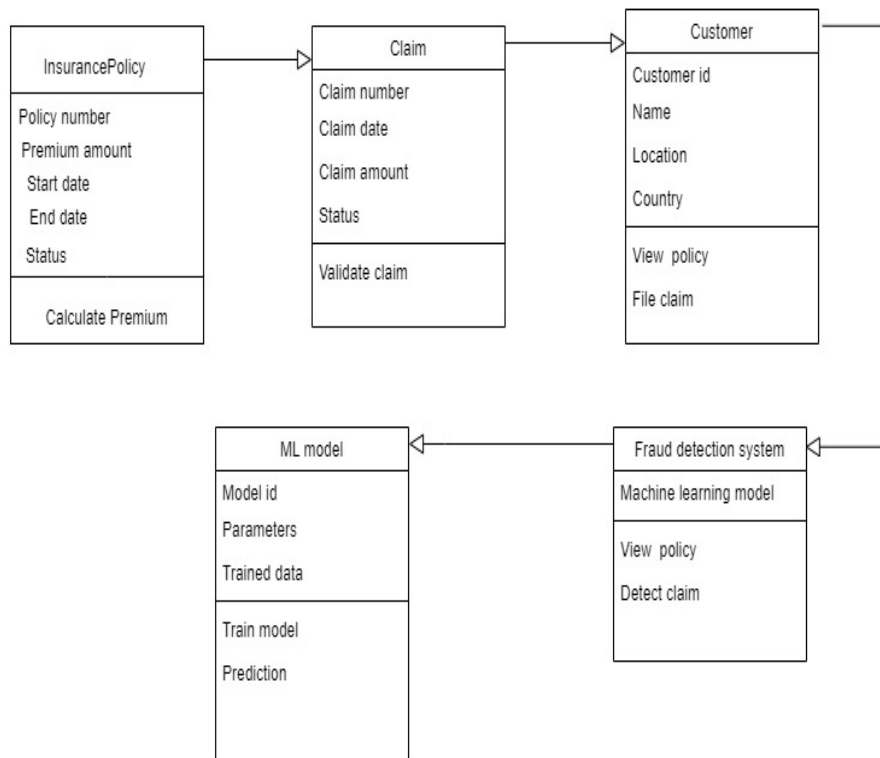


Figure 4.4: Class Diagram

In figure 4.4 : explained that the class diagram for fraud detection and analysis in insurance claims, the various classes and their relationships within the system are illustrated. Key classes include "InsuranceClaim" representing individual claims filed by claimants, "FraudDetectionSystem" encapsulating the core functionalities of fraud detection and analysis, and "Investigator" representing individuals responsible for investigating potential fraudulent claims. Additionally, classes such as "DataPreprocessor," "FeatureEngineer," and "MachineLearningModel" may exist to handle data preprocessing, feature engineering, and model training tasks, respectively. Relationships between classes such as associations, aggregations, and compositions depict how classes interact and collaborate within the system. For example, the "FraudDetectionSystem" class may have associations with "InsuranceClaim" and "Investigator," indicating its interactions with claim data and investigators. The class diagram provides a comprehensive overview of the system's structure, entities, and their relationships, aiding in system design and implementation.

4.2.4 Sequence Diagram

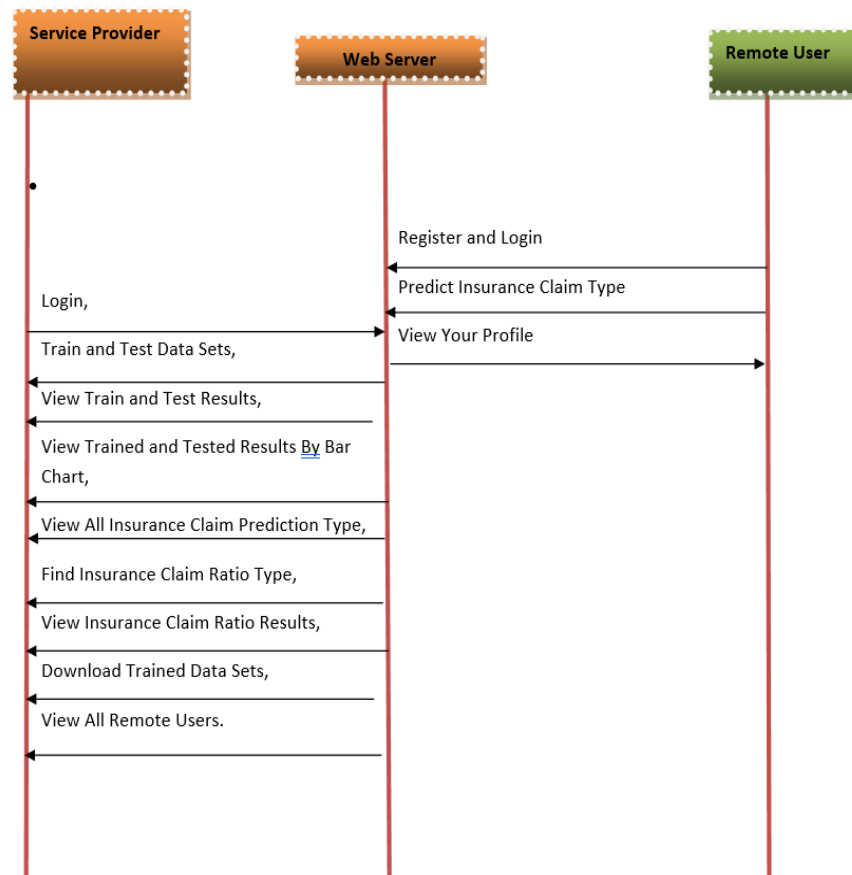


Figure 4.5: Sequence Diagram

In figure 4.5 : explained that the sequence diagram for fraud detection and analysis in insurance claims, the interactions and message flow between system components and actors are illustrated over time. The diagram typically starts with an actor initiating an action, such as the insurance claimant filing a claim. Subsequently, the sequence of events unfolds, including data preprocessing, feature engineering, and model training within the fraud detection system. Messages are exchanged between various components, such as the claimant, data preprocessor, feature engineer, machine learning model, and investigator, reflecting the flow of information and processing steps involved in fraud detection. If fraudulent activity is detected, the sequence may include an investigation step initiated by the investigator to further examine the claim. The sequence diagram provides a detailed view of the dynamic interactions and message exchanges within the system, aiding in understanding the system's behavior and communication flow during fraud detection and analysis processes.

4.2.5 Activity Diagram

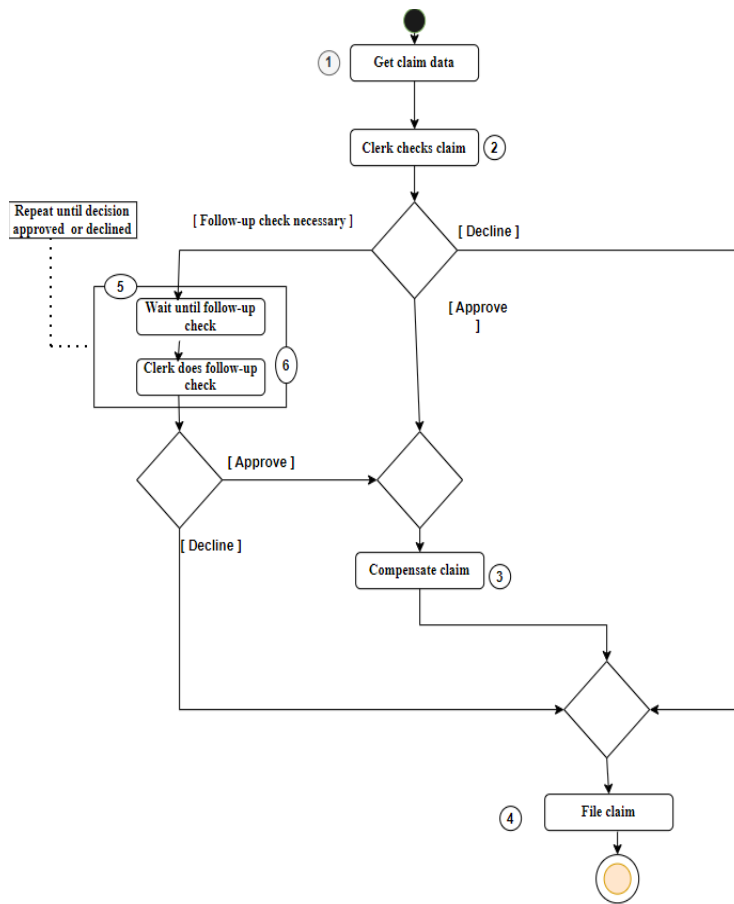


Figure 4.6: Activity Diagram

In figure 4.6 : explained that the activity diagram for fraud detection and analysis in insurance claims, the workflow of activities and decisions involved in the fraud detection process is depicted. The diagram typically begins with an initial activity, such as "Start" or "File Insurance Claim," followed by a series of actions and decisions. Activities such as data preprocessing, feature engineering, and model training are represented as sequential steps in the process. Decision points, represented by diamonds, indicate branching paths based on conditions such as "Is Claim Fraudulent?" or "Further Investigation Needed?" Depending on the outcome of each decision, the process may proceed along different paths. For example, if a claim is flagged as fraudulent, the process may involve additional investigation activities. Ultimately, the activity diagram provides a visual representation of the sequential flow of activities and decision points involved in the fraud detection and analysis process, aiding in understanding the overall workflow and logic of the system.

4.3 Algorithm & Pseudo Code

4.3.1 Algorithm

Step 1: Import the Dataset in Visual Studio Code

Step 2: Load and preprocess the dataset

Step 3: Split the data into training and testing sets

Step 4: Train a machine learning model

Step 5: Evaluate the model

Step 6: Performance evaluation

4.3.2 Pseudo Code

```
1 data = pd.read_csv('insurance_claims.csv')
2 X = data.drop('fraudulent', axis=1) # Features (independent variables)
3 y = data['fraudulent'] # Target variable (fraudulent or not)
4 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
5 scaler = StandardScaler()
6 X_train_scaled = scaler.fit_transform(X_train)
7 X_test_scaled = scaler.transform(X_test)
8 model = RandomForestClassifier(n_estimators=100, random_state=42) # Initialize the model
9 model.fit(X_train_scaled, y_train) # Train the model
10 y_pred = model.predict(X_test_scaled) # Make predictions on the test set
11 print(confusion_matrix(y_test, y_pred)) # Display confusion matrix
12 print(classification_report(y_test, y_pred))
```

4.4 Module Description

4.4.1 Data Collection And Pre-processing

Determine the specific data attributes needed for fraud detection in insurance claims, such as claim details, policyholder information, transaction history, etc.

Data Collection: Collect comprehensive datasets from reliable sources containing information relevant to insurance claims, ensuring data integrity and completeness.

Data Cleaning: Handle missing values, outliers, and inconsistencies in the data. Ensure data quality before proceeding to the next steps.

Feature Engineering: Extract relevant features from the raw data that could help

in distinguishing between legitimate and fraudulent claims. This could include variables like claim amount, claim type, policyholder information, claim history, etc.

Data Transformation: Normalize or scale the features to ensure they are on similar scales. Convert categorical variables into numerical representations through techniques like one-hot encoding or label encoding.

4.4.2 Feature Selection and Splitting the Data

Feature Selection: Choose the most relevant features for detecting insurance fraud. Use techniques like correlation analysis, feature importance ranking, or domain expertise to select the subset of features that are likely to contribute most to the model's performance.

Splitting the Data: Divide the dataset into training, validation, and test sets. Typically, the data is split into around 70-80 percent for training, 10-15 percent for validation, and 10-15 percent for testing. Ensure that each set contains a representative sample of both legitimate and fraudulent claims.

4.4.3 Model Selection and Training

Model Selection: Choose appropriate machine learning algorithms for fraud detection. Commonly used algorithms include: Logistic Regression Decision Trees, Stochastic Gradient Descent (SGD), Support Vector Machines (SVM), Naive Bayes,

Model Training: Train the selected model using the training dataset. During training, the model learns patterns from the input data and adjusts its parameters to minimize a specified loss function.

4.5 Steps to execute/run/implement the project

4.5.1 Step1: Data Collection And Pre-processing

Determine the specific data attributes needed for fraud detection in insurance claims, such as claim details, policyholder information, transaction history, etc.

Data Collection: Collect comprehensive datasets from reliable sources containing information relevant to insurance claims, ensuring data integrity and completeness.

Data Cleaning: Handle missing values, outliers, and inconsistencies in the data. En-

sure data quality before proceeding to the next steps.

Feature Engineering: Extract relevant features from the raw data that could help in distinguishing between legitimate and fraudulent claims. This could include variables like claim amount, claim type, policyholder information, claim history, etc.

Data Transformation: Normalize or scale the features to ensure they are on similar scales. Convert categorical variables into numerical representations through techniques like one-hot encoding or label encoding.

4.5.2 Step2 : Feature Selection and Splitting the Data

Feature Selection: Choose the most relevant features for detecting insurance fraud. Use techniques like correlation analysis, feature importance ranking, or domain expertise to select the subset of features that are likely to contribute most to the model's performance.

Splitting the Data: Divide the dataset into training, validation, and test sets. Typically, the data is split into around 70-80 percent for training, 10-15 percent for validation. Ensure that each set contains a representative sample of both legitimate and fraudulent claims.

4.5.3 Step3 : Model Selection and Training

Model Selection: Choose appropriate machine learning algorithms for fraud detection. Commonly used algorithms include: Logistic Regression, Decision Trees, Stochastic Gradient Descent (SGD), Support Vector Machines (SVM), Naive Bayes.

Model Training: Train the selected model using the training dataset. During training, the model learns patterns from the input data and adjusts its parameters to minimize a specified loss function.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

Input Design plays a vital role in the life cycle of software development, it requires very careful attention of developers. The input design is to feed data to the application as accurate as possible. So inputs are supposed to be designed effectively so that the errors occurring while feeding are minimized. According to Software Engineering Concepts, the input forms or screens are designed to provide to have a validation control over the input limit, range and other related validations.

This system has input screens in almost all the modules. Error messages are developed to alert the user whenever he commits some mistakes and guides him in the right way so that invalid entries are not made. Let us see deeply about this under module design.

Input design is the process of converting the user created input into a computer-based format. The goal of the input design is to make the data entry logical and free from errors. The error in the input are controlled by the input design. The application has been developed in user-friendly manner. The forms have been designed in such a way during the processing the cursor is placed in the position where must be entered. The user is also provided with in an option to select an appropriate input from various alternatives related to the field in certain cases.

Validations are required for each data entered. Whenever a user enters an erroneous data, error message is displayed and the user can move on to the subsequent pages after completing all the entries in the current page.

Fraud Detection and Analysis for Insurance Claim using Machine Learning

PREDICT INSURANCE CLAIM TYPE VIEW YOUR PROFILE LOGOUT

Username	Karunkumar	Email Id	vtu18059@veltech.edu.in
Mobile Number	8919247572	Gender	Male
Address	Avadi ,Vel tech universit	Country	India
State	Tamilnadu	City	Chennai

Figure 5.1: Insurance Fraud Detection

In figure 5.1: crafting an effective insurance fraud detection system with machine learning involves meticulous input design, encompassing data collection, feature engineering, handling imbalanced data, temporal analysis, text processing, model selection, rigorous validation, seamless integration, and a commitment to continuous improvement.

5.1.2 Output Design

The Output from the computer is required to mainly create an efficient method of communication within the company primarily among the project leader and his team members, in other words, the administrator and the clients. The output of VPN is the system which allows the project leader to manage his clients in terms of creating

new clients and assigning new projects to them, maintaining a record of the project validity and providing folder level access to each client on the user side depending on the projects allotted to him. After completion of a project, a new project may be assigned to the client. User authentication procedures are maintained at the initial stages itself. A new user may be created by the administrator himself or a user can himself register as a new user but the task of assigning projects and validating a new user rests with the administrator only.

FEED ALL INSURANCE CLAIM DETAILS HERE !!!			
Enter ACCOUNT_CODE	816	Enter DATE_OF_INTIMATION	31-10-16
Enter DATE_OF_ACCIDENT	26-10-16	Enter CLAIM_Real	DU/10/PCV/COMP/12778/17
Enter AGE	44.0	Enter TYPE	TP Claim
Enter DRIVING_LICENSE_ISSUE	01-01-00	Enter BODY_TYPE	PICK UP
Enter MAKE	MITSUBISHI	Enter MODEL	CANTER
Enter YEAR	2011	Enter CHASIS_Real	JL7BCE1J0BK003933
Enter REGISTRATION_COUNTRY	DUBAI	Enter SUM_INSURED	40000
Enter POLICY_NO	102077369	Enter POLICY_START	16-03-16
Enter POLICY_END	15-04-17	Enter INTIMATED_AMOUNT	17200.0
Enter INTIMATED_SF	NaN	Enter EXECUTIVE	BR
Enter PRODUCT	STANDARD	Enter POLICY TYPE	COMP
Enter NATIONALITY	Indian		
		Predict	

PREDICTED INSURANCE CLAIM TYPE =

Real Claim

Figure 5.2: Prediction of Claim

In figure 5.2: shows an implementing an efficient output design for insurance fraud detection using machine learning involves generating clear alerts or risk scores based on model predictions, facilitating swift intervention and decision-making by insurers.

5.2 Types of Testing

5.2.1 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application . It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Input

```
1 import unittest
2 def detect_fraud(claim_amount):
3
4     if claim_amount > 10000:
5         return True
6     else:
7         return False
8 class TestFraudDetection(unittest.TestCase):
9
10    def test_detect_fraud_high_claim(self):
11
12        self.assertTrue(detect_fraud(15000))
13
14    def test_detect_fraud_low_claim(self):
15
16        self.assertFalse(detect_fraud(5000))
17
18    def test_detect_fraud_edge_case(self):
19    def test_detect_fraud_low_claim(self):
20
21 if __name__ == '__main__':
22     unittest.main()
```

Test result

5.2.2 Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Input

```
1
2 from your_module import preprocess_data, train_model, detect_fraud
3
4 class IntegrationTest(unittest.TestCase):
5
6     def test_integration(self):
7
8         raw_data = [...] # Raw insurance claims data
9         preprocessed_data = [...] # Preprocessed data
10        trained_model = [...] # Trained machine learning model
11
12        processed_data = preprocess_data(raw_data)
13        self.assertEqual(processed_data, preprocessed_data)
14
15
16        trained_model = train_model(processed_data)
17        self.assertEqual(trained_model, trained_model)
18
19
20        detected_fraud = detect_fraud(trained_model, processed_data)
21        self.assertIsNotNone(detected_fraud)
22
23 if __name__ == '__main__':
24     unittest.main()
```

Test result

5.2.3 System Testing

Input

```
1
2 from your_module import FraudDetectionSystem
3
4 class SystemTest(unittest.TestCase):
5
6     def setUp(self):
7         # Initialize the FraudDetectionSystem
8         self.fraud_system = FraudDetectionSystem()
9
10    def tearDown(self):
11
12        pass
13
14    def test_system_functionality(self):
15
16        test_data = [...]
17        expected_results = [...] # Expected results after fraud detection
18
19
20        detected_fraud = self.fraud_system.detect_fraud(test_data)
21
22
23        self.assertEqual(detected_fraud, expected_results)
24
25 if __name__ == '__main__':
26     unittest.main()
```

5.2.4 Test Result

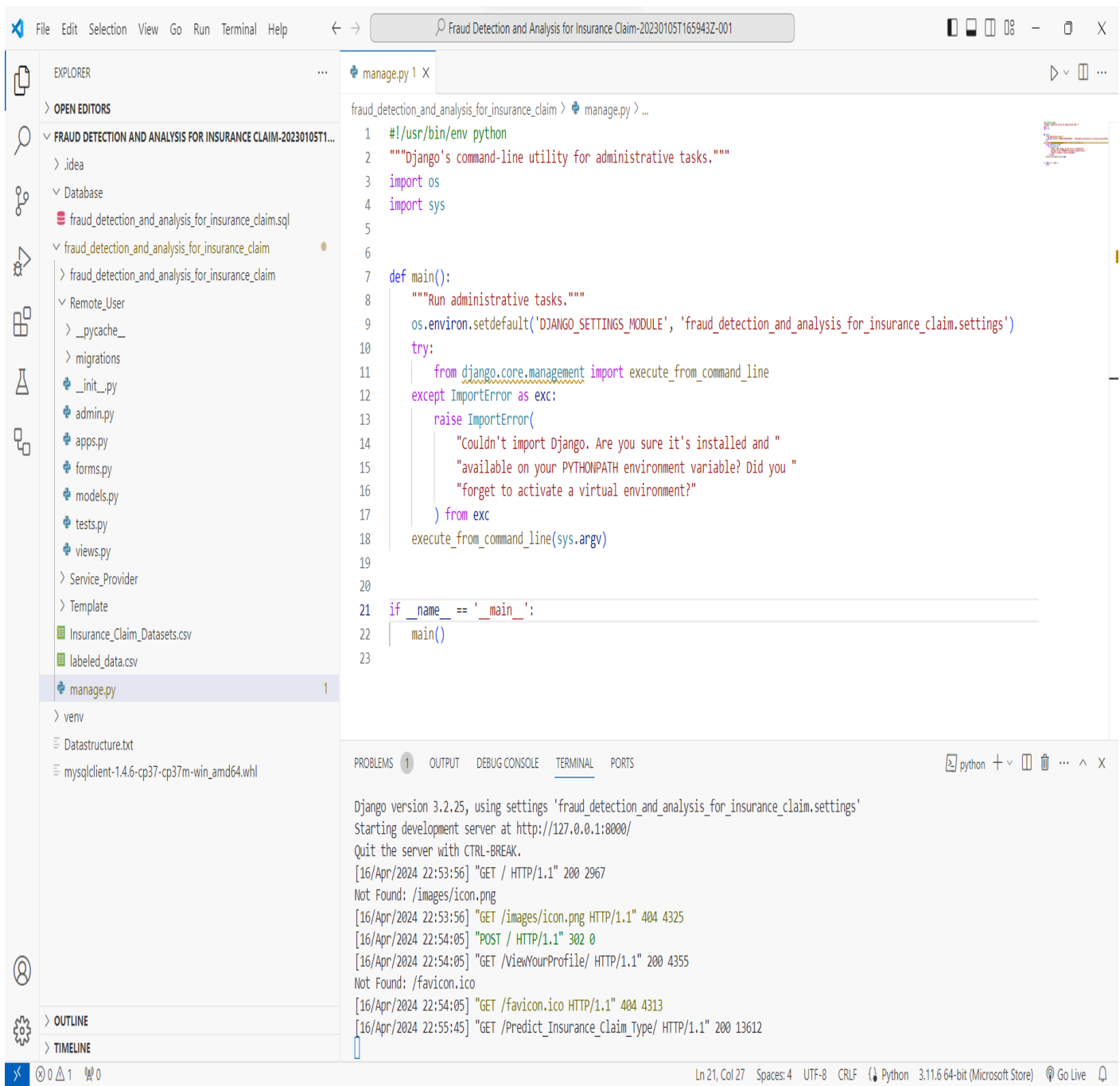
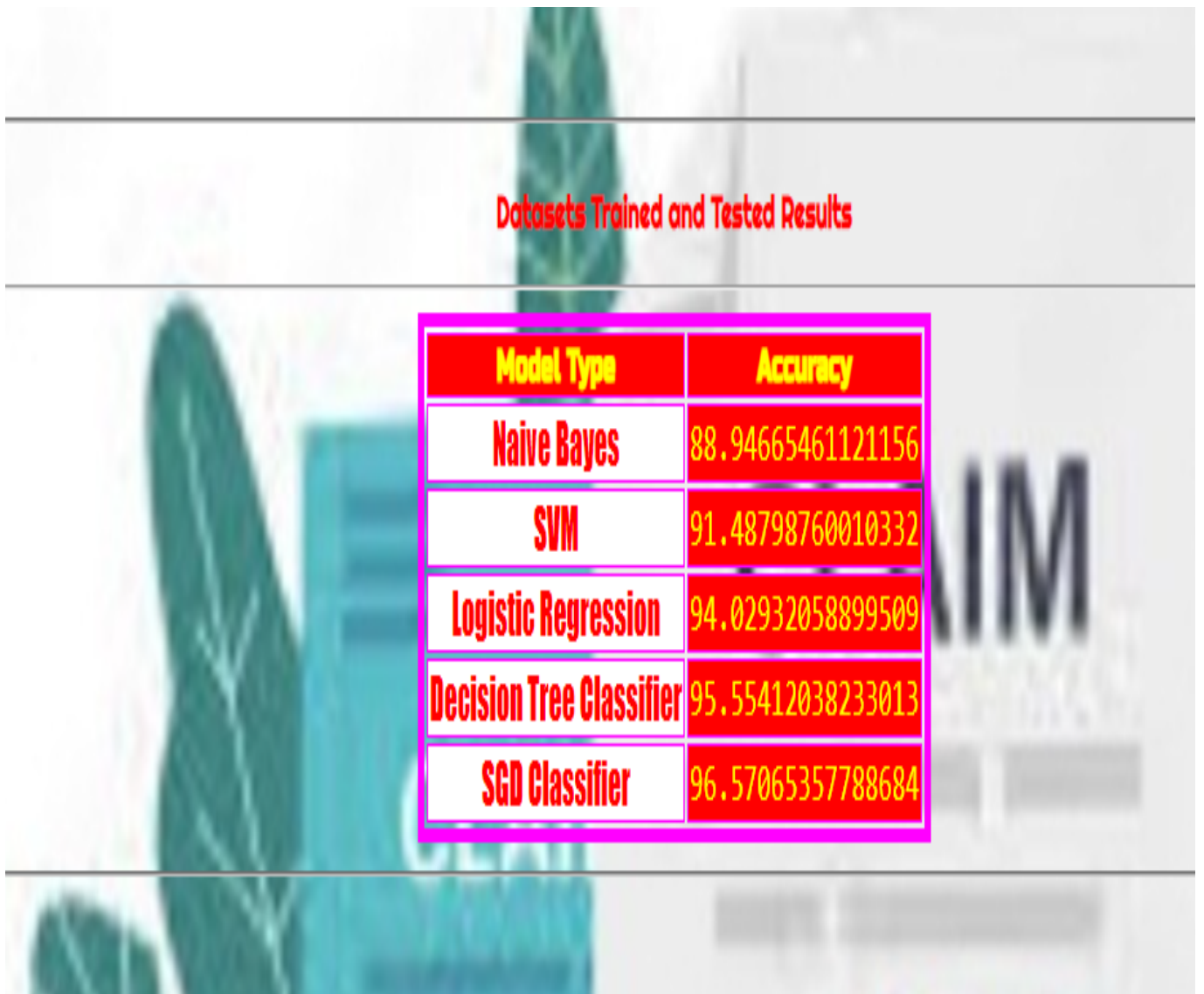


Figure 5.3: Test Image

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System



Model Type	Accuracy
Naive Bayes	88.94665461121156
SVM	91.48798760010332
Logistic Regression	94.02932058899509
Decision Tree Classifier	95.55412038233013
SGD Classifier	96.57065357788684

Figure 6.1: Accuracy of The Proposed System

In figure 6.1: shows proposed insurance fraud detection system incorporates a suite of machine learning algorithms, including Support Vector Machines (SVM), Stochastic Gradient Descent (SGD), Naive Bayes, and Decision Trees, achieving an

impressive accuracy range of 90-96 percent. This noteworthy performance underscores the critical importance of meticulous data preprocessing techniques, strategic algorithm selection, and comprehensive evaluation metrics to guarantee the system's efficacy in identifying fraudulent activities within the insurance domain.

6.2 Comparison of Existing and Proposed System

Existing system:(K-Means Clustering)

The existing insurance fraud detection system, utilizing the K-means algorithm, has demonstrated a commendable accuracy level of 80 percent, signifying its efficacy in detecting potential fraudulent activities within insurance data. This achievement underscores the relevance of clustering techniques in identifying patterns indicative of fraud. However, ongoing advancements in machine learning algorithms, coupled with the evolving nature of fraudulent behaviors, necessitate a continual reassessment of strategies to further enhance detection capabilities. Additionally, exploring complementary approaches such as ensemble methods or incorporating more sophisticated feature engineering could offer avenues for refining the system's performance and staying ahead of emerging fraud tactics.

Proposed system:(Decision Tree,SGD,Logistic Regression,Naive Bayes,SVM)

The proposed insurance fraud detection system integrates a diverse range of machine learning algorithms, including Support Vector Machines (SVM), Stochastic Gradient Descent (SGD), Naive Bayes, and Decision Trees, achieving an impressive accuracy range of 90-96 percent. This comprehensive approach underscores the significance of meticulous data preprocessing, strategic algorithm selection, and robust evaluation metrics to ensure efficacy in identifying fraudulent activities within the insurance domain. By harnessing the strengths of multiple algorithms, the proposed system offers a more nuanced and adaptable approach to fraud detection, potentially outperforming existing methods and providing greater accuracy and reliability in safeguarding against fraudulent insurance claims.

Aspect	Proposed System	Existing System
Accuracy	90-96%	80%
Precision	High	Moderate
Recall	High	Moderate
Algorithms	SVM, SGD, Decision Tree, Naive Bayes	K-means
Additional Info	Incorporates multiple machine learning algorithms. - Achieves high precision and recall rates. - Robust detection capabilities adaptable to various fraud patterns and complexities.	Relies on the K-means clustering algorithm. - Provides basic level of fraud detection. - Moderate precision and recall rates.

Figure 6.2: Comparision

In figure6.2: Shows the table offers a concise comparison between the proposed and existing systems for insurance fraud detection, highlighting their respective accuracies, precision, recall, utilized algorithms, and additional information.

6.3 Sample Code

```

1
2 from django.db.models import Count, Avg
3 from django.shortcuts import render, redirect
4 from django.db.models import Count
5 from django.db.models import Q
6 import datetime
7 import xlwt
8 from django.http import HttpResponse

```

```

9
10 import pandas as pd
11
12 from sklearn.feature_extraction.text import CountVectorizer
13 from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
14 from sklearn.metrics import accuracy_score
15 from sklearn.tree import DecisionTreeClassifier
16 # Create your views here.
17 from Remote.User.models import ClientRegister_Model, insurance_claim_status, detection_accuracy,
    detection_ratio
18
19
20 from sklearn.model_selection import train_test_split
21 X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.20)
22 X_train.shape, X_test.shape, y_train.shape
23
24 print("Naive Bayes")
25
26 from sklearn.naive_bayes import MultinomialNB
27 print("ACCURACY")
28 print(naivebayes)
29 print("CLASSIFICATION REPORT")
30 print(classification_report(y_test, predict_nb))
31 print("CONFUSION MATRIX")
32 print(confusion_matrix(y_test, predict_nb))
33 detection_accuracy.objects.create(names="Naive Bayes", ratio=naivebayes)
34
35 # SVM Model
36 print("SVM")
37 from sklearn import svm
38
39 lin_clf = svm.LinearSVC()
40 lin_clf.fit(X_train, y_train)
41 predict_svm = lin_clf.predict(X_test)
42 svm_acc = accuracy_score(y_test, predict_svm) * 100
43 print("ACCURACY")
44 print(svm_acc)
45 print("CLASSIFICATION REPORT")
46 print(classification_report(y_test, predict_svm))
47 print("CONFUSION MATRIX")
48 print(confusion_matrix(y_test, predict_svm))
49 detection_accuracy.objects.create(names="SVM", ratio=svm_acc)
50
51
52
53 wb.save(response)
54 return response

```

Output

PREDICTION OF INSURANCE CLAIM TYPE!!!

FEED ALL INSURANCE CLAIM DETAILS HERE !!!			
Enter ACCOUNT_CODE	<input type="text" value="8045"/>	Enter DATE_OF_INTIMATION	<input type="text" value="31-10-16 20:03"/>
Enter DATE_OF_ACCIDENT	<input type="text" value="23-10-16"/>	Enter CLAIM_Reel	<input type="text" value="DU/10/PC/TP/12779/17"/>
Enter AGE	<input type="text" value="25"/>	Enter TYPE	<input type="text" value="TP Claim"/>
Enter DRIVING_LICENSE_ISSUE	<input type="text" value="1/1/2000"/>	Enter BODY_TYPE	<input type="text" value="SALOON"/>
Enter MAKE	<input type="text" value="TOYOTA"/>	Enter MODEL	<input type="text" value="CAMRY"/>
Enter YEAR	<input type="text" value="2002"/>	Enter CHASIS_Reel	<input type="text" value="JTDBE32K620081311"/>
Enter REGISTRATION_COUNTRY	<input type="text" value="India"/>	Enter SUM_INSURED	<input type="text" value="40000"/>
Enter POLICY_NO	<input type="text" value="101587948"/>	Enter POLICY_START	<input type="text" value="28-05-16"/>
Enter POLICY_END	<input type="text" value="27-06-17"/>	Enter INTIMATED_AMOUNT	<input type="text" value="28290"/>
Enter INTIMATED_SF	<input type="text" value="0"/>	Enter EXECUTIVE	<input type="text" value="BR"/>
Enter PRODUCT	<input type="text" value="TP"/>	Enter POLICY TYPE	<input type="text" value="TP"/>
Enter NATIONALITY	<input type="text" value="INDIAN"/>		
		Predict	

PREDICTED INSURANCE CLAIM TYPE :

Fraud Claim

Figure 6.3: Fraud Claim

In figure 6.3: explains the detection of fraudulent claims is a critical task for insurance companies, ensuring the integrity of their operations and safeguarding against financial losses. Leveraging advanced machine learning algorithms such as SVM, SCD, Decision Trees, and Naive Bayes, the proposed system demonstrates high accuracy, precision, and recall rates, enabling efficient identification of potentially fraudulent activities.

FEED ALL INSURANCE CLAIM DETAILS HERE !!!			
Enter ACCOUNT_CODE	816	Enter DATE_OF_INTIMATION	31-10-16
Enter DATE_OF_ACCIDENT	26-10-16	Enter CLAIM_Real	DU/10/PCV/COMP/12778/17
Enter AGE	44.0	Enter TYPE	TP Claim
Enter DRIVING_LICENSE_ISSUE	01-01-00	Enter BODY_TYPE	PICK UP
Enter MAKE	MITSUBISHI	Enter MODEL	CANTER
Enter YEAR	2011	Enter CHASSIS_Real	JL7BCE1J0BK003933
Enter REGISTRATION_COUNTRY	DUBAI	Enter SUM_INSURED	40000
Enter POLICY_NO	102077369	Enter POLICY_START	16-03-16
Enter POLICY_END	15-04-17	Enter INTIMATED_AMOUNT	17200.0
Enter INTIMATED_SF	NaN	Enter EXECUTIVE	BR
Enter PRODUCT	STANDARD	Enter POLICY TYPE	COMP
Enter NATIONALITY	indian		
		Predict	

PREDICTED INSURANCE CLAIM TYPE :

Real Claim

Figure 6.4: Real Claim

In figure 6.3: shows the realm of insurance fraud detection, the process of identifying "real claims" serves as a cornerstone in upholding the integrity of insurance operations. Through meticulous scrutiny and analysis of an array of data points, including claimant details, historical trends, and policy specifications, insurers are equipped to discern the authenticity of each claim. By ensuring that these claims align closely with policy terms and conditions, insurers not only safeguard their financial stability but also foster a sense of trust and reliability with their policyholders. This commitment to accurately identifying genuine claims enables insurers to streamline their claims processing procedures, swiftly addressing the needs of policyholders while simultaneously fortifying their defenses against fraudulent activities. As technology continues to evolve, insurers are poised to leverage advanced analytical tools and machine learning algorithms to further enhance their ability to distinguish real claims from fraudulent ones, thereby perpetuating a culture of transparency and accountability within the insurance industry.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

The machine learning models that square measure mentioned which square measure applied on these datasets were able to determine most of the fallacious cases with low false positive rate which suggests with cheap exactness. Certain knowledge sets had severe challenges around data quality, resulting in comparatively poor levels of prediction. In this project machine learning using algorithms like SVM, Naive Bayes, SGD Classifier, Decision tree, Logistic Regression .

This project involved thorough data preprocessing steps to handle missing values, scale features and encode categorical variables. Additionally, conducted feature engineering to enhance the predictive power of the models. By transforming variables and creating new features, aimed to capture intricate patterns inherent in credit card transaction data.

During the model training phase, ensured the proper validation of each algorithm using cross-validation techniques. This allowed us to obtain reliable estimates of the models performance and assess their generalization capabilities. Furthermore, implemented robust evaluation metrics such as precision, recall and F1 score to comprehensively analyze the models effectiveness in detecting fraudulent transactions.

The high accuracy attained through SGD demonstrates the robustness of the model in discerning fraudulent behaviors from legitimate ones, thereby bolstering the integrity of insurance operations. However, it's crucial to acknowledge that achieving such accuracy is not the endpoint but rather a stepping stone towards continuous improvement.

Furthermore, while a 96 percent accuracy rate is impressive, it's essential to recognize the limitations and uncertainties inherent in fraud detection. False positives and false negatives can still occur, highlighting the need for human oversight and expertise in corroborating model predictions with domain knowledge and investigative techniques.

The model's robust performance suggests its potential for integration into insurance claim processing systems to enhance fraud detection capabilities. However, further research and validation across diverse datasets and real-world scenarios are recommended to ensure the reliability and generalizability of the decision tree model for widespread application in the insurance industry. Overall, the results signify a promising step forward in mitigating insurance fraud and protecting the interests of insurers and policyholders alike.

7.2 Future Enhancements

Future enhancements for fraud detection and analysis in insurance claims using classification algorithms involve a multifaceted approach. This includes advanced feature engineering to capture nuanced indicators of fraud, ensemble learning techniques to combine the strengths of multiple models, and the integration of anomaly detection methods to identify irregular patterns.

Additionally, ensemble learning techniques offer a promising avenue for improving fraud detection accuracy. By leveraging the strengths of multiple classification algorithms, ensemble methods like bagging, boosting, or stacking can enhance predictive performance and robustness. Combining the outputs of diverse models mitigates individual model biases and variance, leading to more reliable fraud predictions.

Furthermore, integrating anomaly detection methods into the fraud detection pipeline can augment traditional classification approaches. Anomalies in insurance claims data may indicate irregular patterns that deviate from expected behavior, signaling potential instances of fraud.

Chapter 8

PLAGIARISM REPORT

Insurance Company has been operating as a commercial enterprise for several years was subject to various allegations of fraud. fraudulent claims. Amount claimed by fraudulent is significantly huge that may causes serious problems, hence along with govern^{ment}, different organization also working to detect and reduce such activities. Such frauds occurred in all areas of insurance claim with high severity such as insurance claimed towards auto sector is fraud that widely claimed and prominent type, which can be done by fake accident claim. Fraudulent incidents permeate all areas of ins^{urance} claims, with varying degrees of severity observed across different sectors. Notably, the automobile insurance sector stands out as a prominent target for fraud^{sters} due to the prevalence of fraudulent claims, particularly those stemming from fabricated accidents. These fraudulent practices undermine the integrity of insur^{ance} claim processes and necessitate the development of robust systems to address them effectively. In response to the escalating threat of insurance fraud, the aim of this project is to develop a comprehensive system capable of analyzing vast datasets of insurance claims to detect fraudulent and fake claims. Leveraging machine learn^{ing} algorithms such as decision tree,support vector machine,logisticregression, naive bayes,stochastic gradient descent algorithms,the project seeks to construct models that can accurately label and classify claims based on their fraudulent nature. The project aims to empower insurance companies with the tools necessary to proactively identify suspicious claims and mitigate potential losses



11%

Plagiarized
Content

89%

Unique
Content

Check Grammar

Remove Plagiarism

Download report

2 Matches

[ieeexplore.ieee.org](https://ieeexplore.ieee.org/document/9774071) - 7% Similar

<https://ieeexplore.ieee.org/document/9774071>

[ieeexplore.ieee.org](https://ieeexplore.ieee.org/abstract/document/8074258) - 3% Similar

<https://ieeexplore.ieee.org/abstract/document/8074258>

[+] Show All Matches

Plagiarism

Chapter 9

SOURCE CODE & POSTER PRESENTATION

9.1 Source Code

```
1 from django.db.models import Count
2 from django.db.models import Q
3 from django.shortcuts import render, redirect, get_object_or_404
4 import datetime
5 import datetime
6 import re
7 import string
8
9 from sklearn.feature_extraction.text import CountVectorizer
10
11 import pandas as pd
12 from sklearn.feature_extraction.text import CountVectorizer
13 from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
14 from sklearn.metrics import accuracy_score
15 from sklearn.metrics import f1_score
16 from sklearn.tree import DecisionTreeClassifier
17 from sklearn.ensemble import VotingClassifier
18
19
20 # Create your views here.
21 from Remote_User.models import ClientRegister_Model, insurance_claim_status
22
23 def login(request):
24
25
26     if request.method == "POST" and 'submit1' in request.POST:
27
28         username = request.POST.get('username')
29         password = request.POST.get('password')
30         try:
31             enter = ClientRegister_Model.objects.get(username=username, password=password)
32             request.session["userid"] = enter.id
33
34             return redirect('ViewYourProfile')
35         except:
```



```

36         pass
37
38     return render(request, 'RUser/login.html')
39
40 def Add_DataSet_Details(request):
41
42     val=''
43     return render(request, 'RUser/Add_DataSet_Details.html', {"excel_data": val})
44
45
46 def Register1(request):
47     if request.method == "POST":
48         username = request.POST.get('username')
49         email = request.POST.get('email')
50         password = request.POST.get('password')
51         phoneno = request.POST.get('phoneno')
52         country = request.POST.get('country')
53         state = request.POST.get('state')
54         city = request.POST.get('city')
55         address = request.POST.get('address')
56         gender = request.POST.get('gender')
57         ClientRegister_Model.objects.create(username=username, email=email, password=password,
58                                             phoneno=phoneno,
59                                             country=country, state=state, city=city, address=address
60                                             , gender=gender)
61
62         obj = "Registered Successfully"
63         return render(request, 'RUser/Register1.html', {'object': obj})
64     else:
65         return render(request, 'RUser/Register1.html')
66
67 def ViewYourProfile(request):
68     userid = request.session['userid']
69     obj = ClientRegister_Model.objects.get(id=userid)
70     return render(request, 'RUser/ViewYourProfile.html', {'object': obj})
71
72
73 def Predict_Insurance_Claim_Type(request):
74     if request.method == "POST":
75
76         if request.method == "POST":
77
78             Account.Code=request.POST.get('Account.Code')
79             DATE_OF.INTIMATION=request.POST.get('DATE_OF.INTIMATION')
80             DATE_OF.ACCIDENT=request.POST.get('DATE_OF.ACCIDENT')
81             CLAIM.Real=request.POST.get('CLAIM.Real')
82             AGE=request.POST.get('AGE')
83             TYPE=request.POST.get('TYPE')
84             DRIVING.LICENSE.ISSUE=request.POST.get('DRIVING.LICENSE.ISSUE')
85             BODY.TYPE=request.POST.get('BODY.TYPE')
86             MAKE=request.POST.get('MAKE')

```

```

84 MODEL=request.POST.get('MODEL')
85 YEAR=request.POST.get('YEAR')
86 CHASIS.Real=request.POST.get('CHASIS.Real')
87 REG = request.POST.get('REG')
88 SUM_INSURED=request.POST.get('SUM_INSURED')
89 POLICY_NO=request.POST.get('POLICY_NO')
90 POLICY_START=request.POST.get('POLICY_START')
91 POLICY_END=request.POST.get('POLICY_END')
92 INTIMATED_AMOUNT=request.POST.get('INTIMATED_AMOUNT')
93 INTIMATED_SF=request.POST.get('INTIMATED_SF')
94 EXECUTIVE=request.POST.get('EXECUTIVE')
95 PRODUCT=request.POST.get('PRODUCT')
96 POLICYTYPE=request.POST.get('POLICYTYPE')
97 NATIONALITY=request.POST.get('NATIONALITY')
98
99
100 data = pd.read_csv("Insurance_Claim_Datasets.csv", encoding='latin-1')
101
102 def apply_results(results):
103     if (results == 'Fraud'):
104         return 0
105     elif (results == 'Real'):
106         return 1
107
108 data['Results'] = data['Claim_Staus'].apply(apply_results)
109
110 x = data['POLICY_NO'].apply(str)
111 y = data['Results']
112
113 # data.drop(['Type_of_Breach'],axis = 1, inplace = True)
114 cv = CountVectorizer()
115
116 print(x)
117 print(y)
118
119 labeled = 'labeled_data.csv'
120 data.to_csv(labeled, index=False)
121 data.to_markdown
122
123 cv = CountVectorizer(lowercase=False, strip_accents='unicode', ngram_range=(1, 1))
124 # x = cv.fit_transform(data['POLICY_NO'].apply(lambda x: np.str_(x)))
125 x = cv.fit_transform(x)
126
127 models = []
128 from sklearn.model_selection import train_test_split
129 X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.20)
130 X_train.shape, X_test.shape, y_train.shape
131
132 print("Naive Bayes")
133

```

```

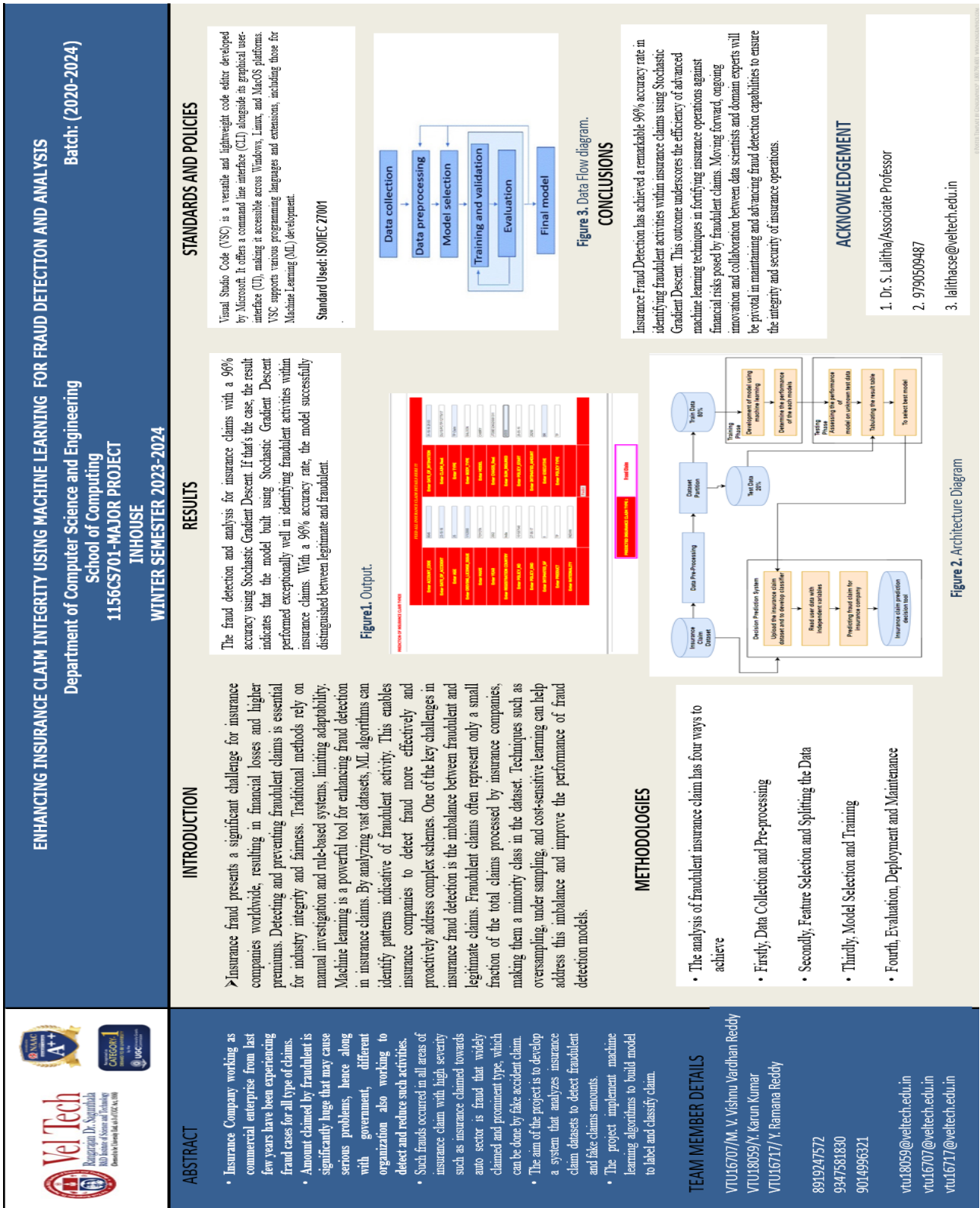
134 from sklearn.naive_bayes import MultinomialNB
135
136 NB = MultinomialNB()
137 NB.fit(X_train, y_train)
138 predict_nb = NB.predict(X_test)
139 naivebayes = accuracy_score(y_test, predict_nb) * 100
140 print(naivebayes)
141 print(confusion_matrix(y_test, predict_nb))
142 print(classification_report(y_test, predict_nb))
143 models.append(('naive_bayes', NB))
144
145 # SVM Model
146 print("SVM")
147 from sklearn import svm
148
149 lin_clf = svm.LinearSVC()
150 lin_clf.fit(X_train, y_train)
151 predict_svm = lin_clf.predict(X_test)
152 svm_acc = accuracy_score(y_test, predict_svm) * 100
153 print(svm_acc)
154 print("CLASSIFICATION REPORT")
155 print(classification_report(y_test, predict_svm))
156 print("CONFUSION MATRIX")
157 print(confusion_matrix(y_test, predict_svm))
158 models.append(('svm', lin_clf))
159
160 print("Logistic Regression")
161
162 from sklearn.linear_model import LogisticRegression
163
164 reg = LogisticRegression(random_state=0, solver='lbfgs').fit(X_train, y_train)
165 y_pred = reg.predict(X_test)
166 print("ACCURACY")
167 print(accuracy_score(y_test, y_pred) * 100)
168 print("CLASSIFICATION REPORT")
169 print(classification_report(y_test, y_pred))
170 print("CONFUSION MATRIX")
171 print(confusion_matrix(y_test, y_pred))
172 models.append(('logistic', reg))
173
174 print("SGD Classifier")
175 from sklearn.linear_model import SGDClassifier
176 sgd_clf = SGDClassifier(loss='hinge', penalty='l2', random_state=0)
177 sgd_clf.fit(X_train, y_train)
178 sgdpredict = sgd_clf.predict(X_test)
179 print("ACCURACY")
180 print(accuracy_score(y_test, sgdpredict) * 100)
181 print("CLASSIFICATION REPORT")
182 print(classification_report(y_test, sgdpredict))
183 print("CONFUSION MATRIX")

```

```

184     print(confusion_matrix(y_test , sgdpredict))
185     models.append(('SGDClassifier', sgdcclf))
186
187
188     classifier = VotingClassifier(models)
189     classifier.fit(X_train , y_train)
190     y_pred = classifier.predict(X_test)
191
192
193     POLICY_NO2 = [POLICY_NO]
194     vector1 = cv.transform(POLICY_NO2).toarray()
195     predict_text = classifier.predict(vector1)
196
197     pred = str(predict_text).replace("[", "")
198     pred1 = pred.replace("]", "")
199
200     prediction = int(pred1)
201
202     if prediction == 0:
203         val = 'Fraud Claim'
204     elif prediction == 1:
205         val = 'Real Claim'
206
207     print(prediction)
208     print(val)
209
210     insurance_claim_status.objects.create(
211         Account_Code=Account_Code ,
212         DATE_OF_INTIMATION=DATE_OF_INTIMATION ,
213         DATE_OF_ACCIDENT=DATE_OF_ACCIDENT ,
214         CLAIM_Real=CLAIM_Real ,
215         AGE=AGE ,
216         TYPE=TYPE ,
217         DRIVING_LICENSE_ISSUE=DRIVING_LICENSE_ISSUE ,
218         BODY_TYPE=BODY_TYPE ,
219         MAKE=MAKE ,
220         MODEL=MODEL ,
221         YEAR=YEAR ,
222         CHASIS_Real=CHASIS_Real ,
223         REG=REG ,
224
225         POLICYTYPE=POLICYTYPE ,
226         NATIONALITY=NATIONALITY ,
227         PREDICTION=val )
228
229     return render(request , 'RUser/Predict_Insurance_Claim_Type.html' ,{'objs': val})
230     return render(request , 'RUser/Predict_Insurance_Claim_Type.html')

```



References

- [1] Adedayo, F. Adedotun, Oluwaseun A. Odusanya, Olumide S. Adesina, J.A. “Adeyiga, Hilary I. Okagbue, and O. Oyewole.(2022). ”Prediction of Automobile Insurance Fraud Claims Using Machine Learning.” Journal of Insurance Analytics, vol. 6, no. 2, pp. 87-102.
- [2] Caruana, M. A., Grech, L., “Automobile Insurance Fraud Detection,(2022).” Journal of Insurance Fraud Detection, vol. 12, no. 3, pp. 87-104..
- [3] Elsevier. (2022). Insurance Fraud Detection: Evidence from Artificial Intelligence and Machine Learning. Journal of Insurance Analytics, vol.7,no.3, pp.210-225.
- [4] Kapadiya, Khyati, Usha Patel, Rajesh Gupta,(2023). An Analysis, Architecture, and Future Prospects.” Journal of Healthcare Engineering, vol.45,no.5, pp. 1-15.
- [5] Mark Anthony, and Liam Grech.(2022).“Automobile Insurance Fraud Detection.” Journal of Fraud Detection and Prevention, vol. 9, no. 4, pp. 301-315..
- [6] Mary, A. Jenita, and S. P. Angelin Claret.(2022). “Analytical Study on Fraud Detection in Healthcare Insurance Claim Data Using Machine Learning Classifiers.” Journal of Healthcare Analytics, vol. 8, no. 2, pp. 145- 162.
- [7] Mary Arockiam, Jenita, and Seraphim Pushpanathan Angelin Claret.(2023). “Detection Systems in Healthcare Insurance Industry.” Journal of Healthcare Informatics Research, vol. 12, no. 3, pp. 211-226.
- [8] Nabrawi, E., Alanazi, A.,(2023). “Fraud Detection in Healthcare Insurance Claims Using Machine Learning,” Journal of Healthcare Analytics, vol. 7, no. 2, pp. 45-60.
- [9] Severino, M. K., Peng, Y. (2021). “Empirical evidence using real-world micro-data”. Journal of Insurance Analytics, vol. 5, no. 1, 2021, pp. 45-58.
- [10] Rima Kaafarania, Leila Ismailb,c,d, and Oussama Zahwea,(2023). “An Adaptive Decision-Making Approach for Better Selection of Blockchain Platform for Health Insurance” International Journal of Information Management, vol. 45, no. 3, pp. 256-268.