INTERNET FINANCIAL FRAUD DETECTION BASED ON A DISTRIBUTED BIG DATA APPROACH WITH NODE2VEC

A Project Stage - II Report submitted to
JNTU Hyderabad in partial fulfilment
of the requirements for the award of the degree

BACHELOR OF TECHNOLOGY
In
ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

BATTULA SHILPA	**20RG1A0468**
BOBIDE VAISHNAVI	**20RG1A0470**
KOMPALLY TANUJA	**20RG1A0487**
LINGALA NIKITHA GOUD	**20RG1A0490**
NUKALA SHREEYA REDDY	**20RG1A04A1**

Under the Guidance of

Mrs. NIRMALA

B.Tech., M.Tech.

Assistant Professor

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING
MALLA REDDY COLLEGE OF ENGINEERING FOR WOMEN
UGC Autonomous Institution
Approved by AICTE New Delhi and Affiliated to JNTU
An ISO 9001: 2015 Certified Institution
B.Tech. Programs CSE, ECE Accredited by NBA
IIC 5.0 with 4-star Rating; INDIAN RANKINGS 2023 NIRF Innovation Band 151-300
Maisammaguda, Medchal (Dist.), Hyderabad -500100, Telangana.

MAY-2024

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

MALLA REDDY COLLEGE OF ENGINEERING FOR WOMEN

UGC Autonomous Institution

Approved by AICTE New Delhi and Affiliated to JNTU
An ISO 9001: 2015 Certified Institution
B.Tech. Programs CSE, ECE Accredited by NBA
IIC 5.0 with 4-star Rating; INDIAN RANKINGS 2023 NIRF Innovation Band 151-300
Maisammaguda, Medchal (Dist.), Hyderabad -500100, Telangana.

MAY-2024

CERTIFICATE

This is to certify that the Project stage -II entitled **"INTERNET FINANCIAL FRAUD DETECTION BASED ON A DISTRIBUTED BIG DATA APPROACH WITH NODE2VEC"** has been submitted by **Battula Shilpa (20RG1A0468), Bobide Vaishnavi (20RG1A0470),Kompally Tanuja (20RG1A0487), Lingala Nikitha Goud (20RG1A0490), Nukala Shreeya Reddy (20RG1A04A1)** in partial fulfilment of the requirements for the award of **BACHELOR OF TECHNOLOGY** in **ELECTRONICS & COMMUNICATION ENGINEERING**. This record of bonafide work was carried out by them under my guidance and supervision. ***The result embodied in this Project stage -II report has not been submitted to any other University or Institute for the award of any degree.***

Mrs. Nirmala
Assistant Professor
Project Guide

Dr. Selvamani Indrajith
Head of the Department

External Examiner

ACKNOWLEDGEMENT

The Project Stage - II work carried out by our team in the Department of Electronics and Communication Engineering, Malla Reddy College of Engineering for Women, Hyderabad. ***This work is original and has not been submitted in part or full for any degree or diploma of any other university.***

We wish to acknowledge our sincere thanks to our project guide ***Mrs. Nirmala,*** Assistant Professor of Electronics & Communication Engineering Department for formulating the problem, analysis, guidance and her continuous supervision during the project work.

We acknowledge our sincere thanks to ***Dr. Kanaka Durga Returi,*** Principal and Professor of Computer Science & Engineering Department. ***Dr. Selvamani Indrajith***, Professor and Head of the Electronics & Communication Engineering Department, for their kind cooperation in making this Project Stage - II work a success.

We extend our gratitude to ***Sri. Ch. Malla Reddy***, Founder-Chairman and ***Sri. Ch. Mahender Reddy***, Secretary and ***Dr. Vaka Murali Mohan,*** Director for their kind cooperation in providing the infrastructure for successful completion of our Project stage -II work.

We convey our special thanks to the entire teaching faculty and non-teaching staff members of the Electronics & Communication Engineering Department for their support in making this project work a success.

BATTULA SHILPA	**(20RG1A0468)**	_____
BOBIDE VAISHNAVI	**(20RG1A0470)**	_____
KOMPALLY TANUJA	**(20RG1A0487)**	_____
LINGALA NIKITHA GOUD	**(20RG1A0490)**	_____
NUKALA SHREEYA REDDY	**(20RG1A04A1)**	_____

INDEX

ABSTRACT

The rapid development of information technologies like Internet of Things, Big Data, Artificial Intelligence, Blockchain, etc., has profoundly affected people's consumption behaviours and changed the development model of the financial industry. The financial services on Internet and IoT with new technologies has provided convenience and efficiency for consumers, but new hidden fraud risks are generated also. Fraud, arbitrage, vicious collection, etc., have caused bad effects and huge losses to the development of finance on Internet and IoT. However, as the scale of financial data continues to increase dramatically, it is more and more difficult for existing rule-based expert systems and traditional machine learning model systems to detect financial frauds from large-scale historical data. In the meantime, as the degree of specialization of financial fraud continues to increase, fraudsters can evade fraud detection by frequently changing their fraud methods.

In this project, an intelligent and distributed Big Data approach for Internet financial fraud detections is proposed to implement graph embedding algorithm Node2Vec to learn and represent the topological features in the financial network graph into low-dimensional dense vectors, so as to intelligently and efficiently classify and predict the data samples of the large-scale dataset with the deep neural network. The approach is distributed Ly performed on the clusters of Apache Spark Graph X and Hadoop to process the large dataset in parallel. The groups of experimental results demonstrate that the proposed approach can improve the efficiency of Internet financial fraud detections with better precision rate, recall rate, F1-Score and F2-Score.

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

With the rapid development of the information technologies like Internet of Things, Big Data, Artificial Intelligence, Blockchain, etc., the digital life led by financial technology has profoundly affected people's consumption behaviours and changed the development model of the traditional financial industry to a certain extent. In particular, technical products such as mobile payment, IOT financial services and Internet financial wealth management have penetrated into lots of aspects of economic and social activities.

From 2014 to the present, the development momentum of China's Internet consumer finance industry has been good, and various mobile e-commerce companies have entered the consumer finance field through installment payments and small loans, which has promoted the development of related industries.

Internet financial services based on consumer credits in China, such as Huabei launched by Ant Financial and Alipay of Alibaba Group, Jing dong Baitiao operated by JD.com, WeiLiDai launched by We Bank of Tencent, etc., have enabled consumers to enjoy the online shopping experience of "consumption first, pay later", and covered the e-commerce installment shopping, cash borrowing and other businesses.

Especially in 2020, the COVID-19 pandemic has caused a surge in online transaction volume and brought a large number of online customers to online service providers. It has cultivated the habit of more groups of users to make online purchases and payments through mobile phones and IOT devices, which brings continuous impetus to the development of the Internet financial industry.

The rapid development of mobile and IOT financial payment services has not only provided convenience and efficiency for consumers, but also brought more hidden fraud risks. Due to the concealment of the complex network, there could be a breeding ground for fraudulent activities by criminals. The control of fraud risks is becoming more and more difficult and

fraud cases occur frequently, which causes the fraud losses to commercial banks and financial institutions are also increasing.

The continuous happening of Internet financial fraudulent problems, such as the agreement cash-out incident of Huabei and Taobao merchants, and "Baitiao" multiple fraud incidents, have not only damaged the legitimate rights and interests of the service platform, but also caused consumers to question the company's account security and risk identification capabilities.

A large number of violations are beyond the scope of the industry's existing laws and regulations, and industry regulation has always lagged behind the innovative development of Internet consumer finance, which makes the regulatory laws and regulations are often in a state of absence so that it impossible to deal with industry violations in a timely manner.

Fraud, arbitrage, vicious collection and other phenomena are becoming more and more rampant in online financial service platforms, which has caused bad effects and huge losses to the development of consumer finance on Internet and IOT. Fraud is an illegal or criminal deception aimed at obtaining financial or personal benefits.

Fraud generally has the attributes of abnormal or unfair transactions. Due to the inconsistency with previous fund operation rules or other normal behaviours, fraudulent behaviour presents various abnormal characteristics, including abnormal transaction amount, abnormal transaction time, abnormal transaction account, abnormal transaction IP, or abnormal personal credit rating.

Currently, fraud detection schemes in the industry mainly include rule-based expert systems and machine learning- based model systems. The rule-based expert system requires anti-fraud experts to manually analyse a large amount of normal and abnormal transaction data, accurately identify the behavior of fraudsters, find important features that can effectively distinguish fraud, and write expert rules for fraud detection.

Therefore, the rule-based expert system strongly relies on the professional knowledge and business knowledge of the anti-fraud experts. If

the experts cannot detect increasingly complex fraud patterns in a timely and keen manner, it will cause huge losses. With the continuous increase of machine computing power, model systems based on machine learning have emerged. The machine learning-based model system is generally divided into four modules: data preprocessing, feature engineering, model training and model prediction.

Data preprocessing includes missing value processing, sampling and other steps. After the processing is completed, cumulative calculations are usually performed based on historical transaction data to convert the original data into characteristic data. After that, models such as machine learning regression or classification are used for training and evaluation on the data set. Finally, the model goes online for fraud detection.

However, as the scale of financial transaction data continues to increase dramatically, it is more and more difficult for rule-based expert systems and traditional machine learning model systems to detect transaction frauds or fraudulent behaviour patterns from large-scale historical data when faced with massive data levels. In the meantime, as the degree of specialization of financial fraud continues to increase, fraudsters can evade fraud detection by frequently changing their own fraud methods.

Nevertheless, it is difficult for fraudsters to change all their associated relationships. When the associated network graph can cover a large area, even if a fraudster or fraudulent behaviour is careful, it may unwittingly reveal clues. Therefore, in the context of large-scale financial data, how to effectively mine the topological structure characteristics of the association network graph in real time and improve the effect of models for financial fraud detection is a new direction for researchers to explore.

In this article, an intelligent and distributed Big Data approach for Internet financial fraud detection is proposed to implement graph embedding algorithm Node2Vec to learn and represent the topological features in the financial network graph into low-dimensional dense vectors, so as to intelligently and efficiently classify and predict the data samples of the large-scale dataset with the deep neural network.

The approach is distribute performed on the clusters of Apache Spark Graph X and Hadoop to process the large dataset in parallel. The groups of experimental results demonstrate that the proposed approach can improve the efficiency of Internet financial fraud detections with better precision rate, recall rate, F1-Score and F2-Score.

The rest of the article is organized as follows. Literature of related works is described in Section 2. Section 3 demonstrates the graph embedding algorithm of Node2Vec representation learning. An intelligent and distributed Big Data approach for Internet financial fraud detection is proposed in Section 4. In Section 5, groups of experiments are implemented to evaluate the efficiency of the proposed approach. Conclusions and future works are summarized in Section 6.

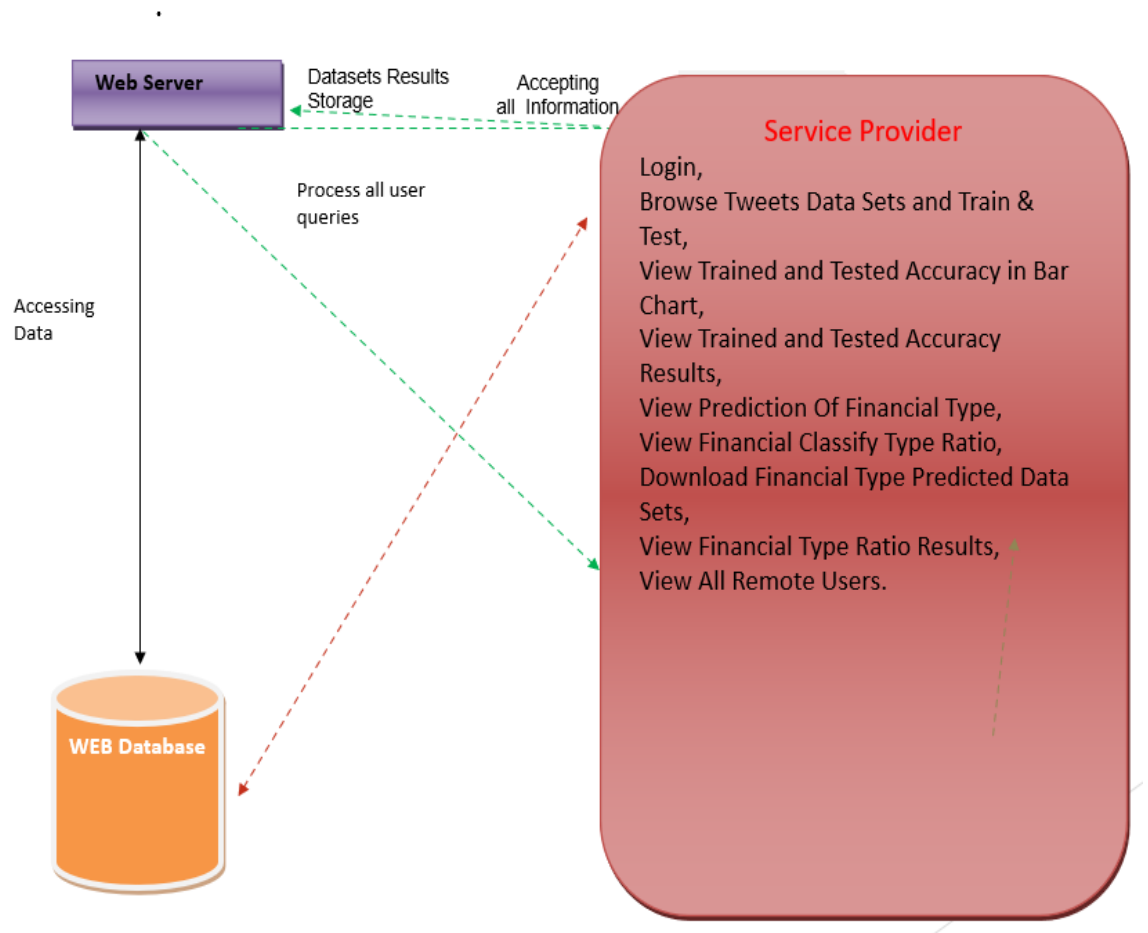

Fig: 1.1 Architecture Diagram

CHAPTER 2: LITERATURE SURVEY

U. Paschen, C. Pitt, and J. Kietzmann [1] The range of topics and the opinions expressed on artificial intelligence (AI) are so broad that clarity is needed on the field's central tenets, the opportunities AI presents, and the challenges it poses. To that end, we provide an overview of the six building blocks of artificial intelligence: structured data, unstructured data, preprocesses, main processes, a knowledge base, and value-added information outputs. We then develop a typology to serve as an analytic tool for managers grappling with AI's influence on their industries. The typology considers the effects of AI-enabled innovations on two dimensions: the innovations' boundaries and their effects on organizational competencies. The typology's first dimension distinguishes between product-facing innovations, which influence a firm's offerings, and process-facing innovations, which influence a firm's operations. The typology's second dimension describes innovations as either competence-enhancing or competence-destroying; the former enhances current knowledge and skills, whereas the latter renders existing skills and knowledge obsolete. This framework lets managers evaluate their markets, the opportunities within them, and the threats arising from them, providing valuable background and structure to important strategic decisions.

P. Yu, Z. Xia, J. Fei, and S. K. Jha [2] "An application review of artificial intelligence in prevention and cure of COVID-19 pandemic," *CMC-Computers Materials & Continua*, vol. 65, no. 1, pp. 743-760, 2020. Coronaviruses are a well-known family of viruses that can infect humans or animals. Recently, the new coronavirus (COVID-19) has spread worldwide. All countries in the world are working hard to control the coronavirus disease. However, many countries are faced with a lack of medical equipment and an insufficient number of medical personnel because of the limitations of the medical system, which leads to the mass spread of diseases. As a powerful tool, artificial intelligence (AI) has been successfully applied to solve various complex problems ranging from big data analysis to computer vision. In the process of epidemic control, many algorithms are proposed to solve problems in various fields of medical

treatment, which is able to reduce the workload of the medical system. Due to excellent learning ability, AI has played an important role in drug development, epidemic forecast, and clinical diagnosis. This research provides a comprehensive overview of relevant research on AI during the outbreak and helps to develop new and more powerful methods to deal with the current pandemic.

L. Shen, X. Chen, Z. Pan, K. Fan, F. Li, and J. Lei [3] No-reference stereoscopic images quality assessment (NR-SIQA) via deep learning has gained increasing attention. In this paper, we propose a no-reference stereoscopic image quality assessment method based on global and local content characteristics. The proposed method simulates the perception route of human visual system, and derives features from the fused view and single view through the global feature fusion sub-network and local feature enhancement sub-network. As for the fused view, a cross-fusion strategy is applied to model the process in the V1 visual cortex, and the multi-scales pooling (MSP) is utilized to integrate context information under different sub-regions for effective global feature extraction. As for the single view, the asymmetric convolution block (ACB) is introduced to strengthen the local information description. By jointly considering the fused view and single view, the proposed network can efficiently extract the features for quality assessment. Finally, a weighted average strategy is applied to estimate the visual quality of stereoscopic image.

H. Beck [4] This paper examines the question how the future of financial intermediaries and banks as special financial intermediaries may look like in the age of the Internet. The reduction of transaction costs caused by the Internet will reduce the barriers to enter the market for financial products, because there may be no longer a need to run a large system of cost-intensive branches. But as closer examination of the functions of financial intermediaries shows, not everybody can sell and distribute financial products. This is true because of asymmetric information problems in financial business which require an intermediary with a good reputation and because of the need to keep large funds of capital to transform the risk of

assets. Both requirements represent an important barrier to enter the market for financial intermediation.

G. N. Weiss, K. Pelger, and A. Horsch [5] This paper presents novel empirical evidence on the success of efforts by P2P lending platforms to limit adverse selection using a unique sample of 5,385 credit transactions on the internet platform Prosper.com. The results of our regressions on the probability of a credit bid's successful funding show that all variables for which a significant influence on the probability of funding success could be found describe information which is verified by Prosper.com. Conversely, all non-verified variables do not possess any significant influence on the dependent variable thus confirming our hypothesis that the screening of potential borrowers is a major instrument in mitigating adverse selection in P2P lending and preventing the online market to collapse. Moreover, we find evidence confirming the proposition that the screening of potential borrowers by groups can help mitigate adverse selection.

Y. Houston, C. Jongrong, J. H. Cliff, and H. Y. Chih [6] This study investigates the impact of e-commerce and R&D on productivity, using a unique panel dataset obtained from Taiwanese manufacturing firms for the period from 1999 to 2002. We specifically consider the network externalities of e-commerce and employ the system generalized method of moment (GMM) technique to deal with the endogenous problem of e-commerce adoption. The empirical results show that both e-commerce and R&D capital have a positive influence on productivity, while R&D exhibits a larger productivity-enhancing effect. We also find a complementary relationship between e-commerce and R&D on enhancing productivity.

F. Allen, J. Mcandrews, and P. Strahan [7] E-finance is defined as "The provision of financial services and markets using electronic communication and computation". In this paper we outline research issues related to e-finance that we believe set the stage for further work in this field. Three areas are focused on. These are the use of electronic payments systems, the operations of financial services firms and the operation of financial markets. A number of research issues are raised. For example, is the widespread use

of paper-based checks efficient? Will the financial services industry be fundamentally changed by the advent of the Internet? Why have there been such large differences in changes to market microstructure across different financial markets

A. Momparler, C. Lassala, and D. Ribeiro [9] This paper investigates the provision of financial services by banks as a two-stage production process involving three different basic activities. The first stage includes service activities, while the second stage comprises both investment-related and risk management activities. Financial services performance is assessed in terms of service efficiency and investment and risk management efficiency for years 2002–2010. The major empirical findings are that the Internet-primary bank is more efficient than most branching banks in deposit-raising activities, but with regard to investment and risk management activities, there are many brick-and-mortar banks that match the online bank performance. Copyright Springer-Verlag Berlin Heidelberg 2013

V. Jambulapati and J. Stavins [10] The Credit CARD Act of 2009 was intended to prevent practices in the credit card industry that lawmakers viewed as deceptive and abusive. Among other changes, the Act restricted issuers' account closure policies, eliminated certain fees, and made it more difficult for issuers to change terms on credit card plans. Critics of the Act argued that because of the long lag between approval and implementation of the law, issuing banks would be able to take preemptive actions that might disadvantage cardholders before the law could take effect. Using credit bureau data as well as individual data from a survey of U.S. consumers, we test whether banks closed consumers' credit card accounts or otherwise restricted access to credit just before the enactment of the CARD Act. Because the period prior to the enactment of the CARD Act coincided with the financial crisis and recession, causality in this case is particularly difficult to establish. We find evidence that a higher fraction of credit card accounts were closed following the Federal Reserve Board's adoption of its credit card rules, but not between May 2009, when the CARD Act was signed, and when most of its provisions became law in February 2010

CHAPTER 3: EXISTING SYSTEM

Allen et al. find that there are many credit channels in the United States and based on the research of American household credit models, and that household consumption, household income, credit banks and credit scale are obviously related. Kregel studies the development trend of consumer finance and finds that the development of Internet consumer finance companies must fully consider the current market legal environment, financial market and consumer behavior factors, etc. Internet consumer finance is directly related to the current development of the national financial system. Momparler et al. take the American Internet consumer finance company as the research object, study the risks and advantages of the Internet consumer finance platform, and design a related risk management model.

Ficawoyi et al. analyze the positive relationship between Internet exposure levels and credit card default through surveys on consumer finance and income nodes. The research points out that Internet access, low income, and male families are more likely to cause credit card defaults. Giudici et al. propose how to improve credit risk accuracy of P2P Internet financial platforms and of those who lend to small and medium enterprises. The augment traditional credit scoring methods are put forward with “alternative data” that consist of centrality measures derived from similarity networks among borrowers and deduced from their financial ratios. The experimental findings suggest that the proposed approach improves predictive accuracy as well as model explainability.

CHAPTER 4: PROPOSED SYSTEM

Through studying a large number of Internet financial fraud cases, two important characteristics are found. The pattern of Internet financial fraud continues to evolve and develop over time, not just repeating the existing individual behavior patterns appeared in historical cases. With the advancement of anti-fraud technology, it is getting harder for individuals to commit Internet financial fraud. It needs to be organized and conducted through related and connected groups. A graph is an abstract graph formed by a number of nodes and the edges connecting each node. It is usually used to describe a specific relationship between things.

A relational network graph refers to a graph-based data structure composed of nodes and edges. Each node represents an entity, and each edge is the relationship between an entity and the other connected entity. The relationship network graph connects different entities together according to their relationships; thus it could provide the ability to analyze problems from the perspective of "relationship".

In anti-fraud applications, entities in the network graph, such as people, equipment, mailboxes, card numbers, etc., can be represented by nodes, and the relationships between these nodes in the business can be represented by edges. Through continuous construction and reproduction of the associated relationships hidden covertly in Internet financial frauds, fraud characteristics can be detected and corresponding risk control strategies can be designed. The graph algorithms can characterize various high-risk features in the Internet finance, such as batch attacks, intermediary participation, etc., which is more effective to identify abnormal group frauds from normal behaviors.

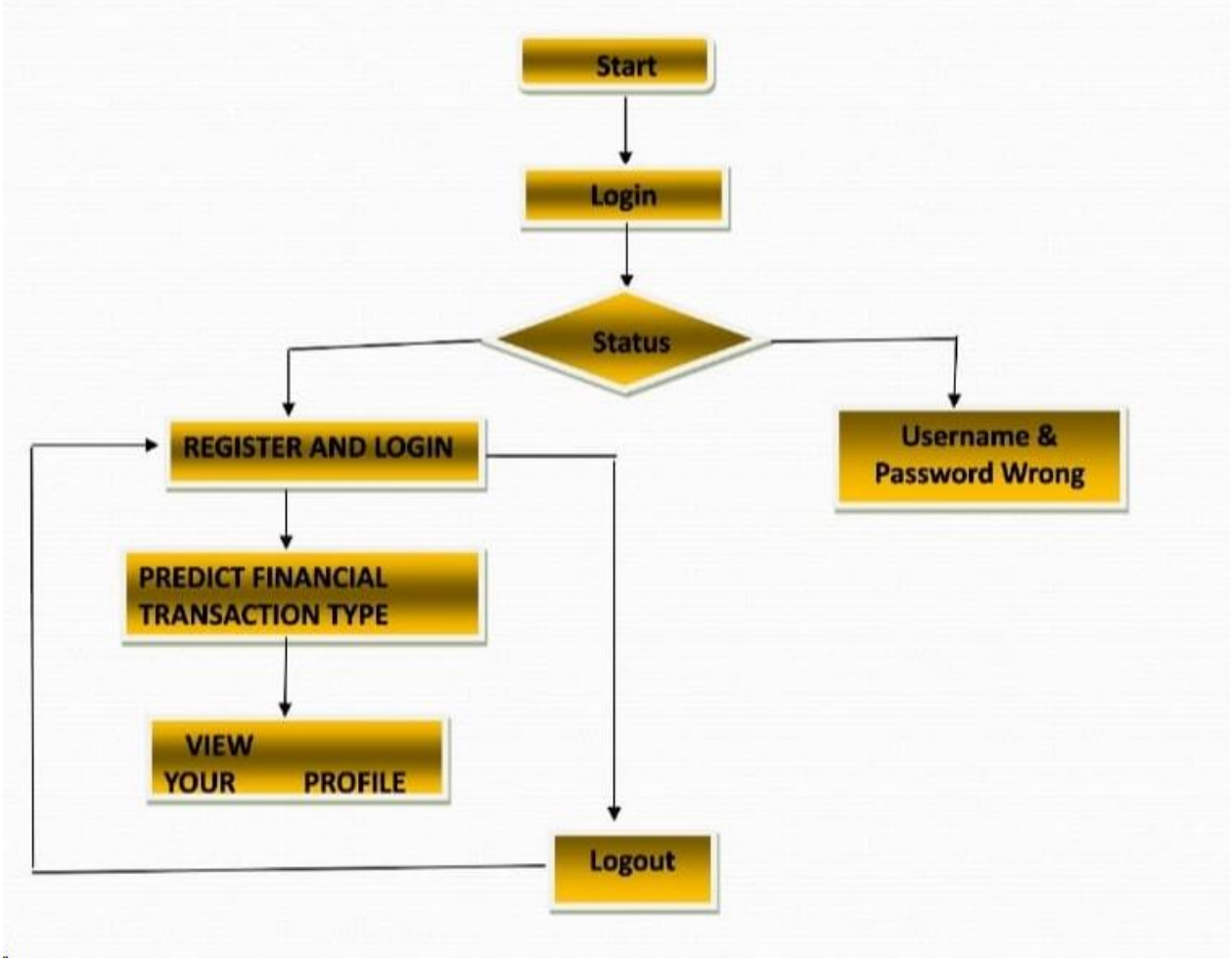

Fig: 4.1 Flow chart of the proposed system

In this chapter describe about the to get the result. At first we start the process and login to the server then after we get the status there we need to register or login with our registered Id and password. If there's no registered Id and password we get a user name and password wrong. By entering registered Id and password we can predict financial transaction type and after we can see our profile. This process repeats after completion of process we can log out.

CHAPTER 5: MODULE DESCRIPTION

Service Provider:

In this module, the Service Provider has to login by using valid user name and password. After login successful he can do some operations such as Login, Browse Tweets Data Sets and Train & Test, View Trained and Tested Accuracy in Bar Chart, View Trained and Tested Accuracy Results, View Prediction Of Financial Type, View Financial Classify Type Ratio, Download Financial Type Predicted Data Sets, View Financial Type Ratio Results, View All Remote Users..

View and Authorize Users

In this module, the admin can view the list of users who all registered. In this, the admin can view the user's details such as, user name, email, address and admin authorizes the users.

Remote User

In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like REGISTER AND LOGIN, PREDICT FINANCIAL TRANSACTION TYPE, VIEW YOUR PROFILE.

CHAPTER 6: SYSTEM REQUIREMENTS

SYSTEM REQUIREMENTS

- Processor - Pentium –IV
- RAM - 4 GB (min)
- Hard Disk - 20 GB
- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse
- Monitor - SVGA

SOFTWARE REQUIREMENTS:

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

CHAPTER 7: SYSTEM STUDY

7.1 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

ECONOMICAL 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.

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.

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. 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

CHAPTER 8: SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

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.

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.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

8.1 Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

8.2 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

8.3 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

SYSTEM_TESTING

TESTING METHODOLOGIES

The following are the Testing Methodologies:

- **Unit Testing.**
- **Integration Testing.**
- **User Acceptance Testing.**
- **Output Testing.**
- **Validation Testing.**

Unit Testing

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for the expected results. All error handling paths are also tested.

Integration Testing

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

The following are the types of Integration Testing:

1.Top -Down Integration

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner. In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

2. Bottom-up Integration

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps:

- The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.
- A driver (i.e.) the control program for testing is written to coordinate test case input and output.
- The cluster is tested.
- Drivers are removed and clusters are combined moving upward in the program structure

The bottom up approaches tests each module individually and then each module is module is integrated with a main module and tested for functionality.

OTHER TESTING METHODOLOGIES

User Acceptance Testing

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

Validation Checking

Validation checks are performed on the following fields

Text Field:

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

Numeric Field:

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error messages. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested.

A successful test is one that gives out the defects for the inappropriate data and produces and output revealing the errors in the system.

Preparation of Test Data:

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

Using Live Test Data:

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves.

It is difficult to obtain live data in sufficient amounts to conduct extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, assuming that the live data entered are in fact typical, such data generally will not test all combinations

or formats that can enter the system. This bias toward typical values then does not provide a true system test and in fact ignores the cases most likely to cause system failure.

Using Artificial Test Data:

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department , make possible the testing of all login and control paths through the program.

The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications.

The package "Virtual Private Network" has satisfied all the requirements specified as per software requirement specification and was accepted.

USER TRAINING

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed. For this purpose the normal working of the project was demonstrated to the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

MAINTAINENCE

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user's requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent. With development in

technology, it may be possible to add many more features based on the requirements in future. The coding and designing is simple and easy to understand which will make maintenance easier.

TESTING STRATEGY:

A strategy for system testing integrates system test cases and design techniques into a well-planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation. A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

SYSTEM TESTING:

Software once validated must be combined with other system elements (e.g. Hardware, people, database). System testing verifies that all the elements are proper and that overall system function performance is achieved. It also tests to find discrepancies between the system and its original objective, current specifications and system documentation.

UNIT TESTING:

In unit testing different are modules are tested against the specifications produced during the design for the modules. Unit testing is essential for verification of the code produced during the coding phase, and hence the goals to test the internal logic of the modules. Using the detailed design description

as a guide, important Conrail paths are tested to uncover errors within the boundary of the modules. This testing is carried out during the programming stage itself. In this type of testing step, each module was found to be working satisfactorily as regards to the expected output from the module.

INPUT SAMPLE :

open XAMPP control panel

Start Apache and then start My SQL

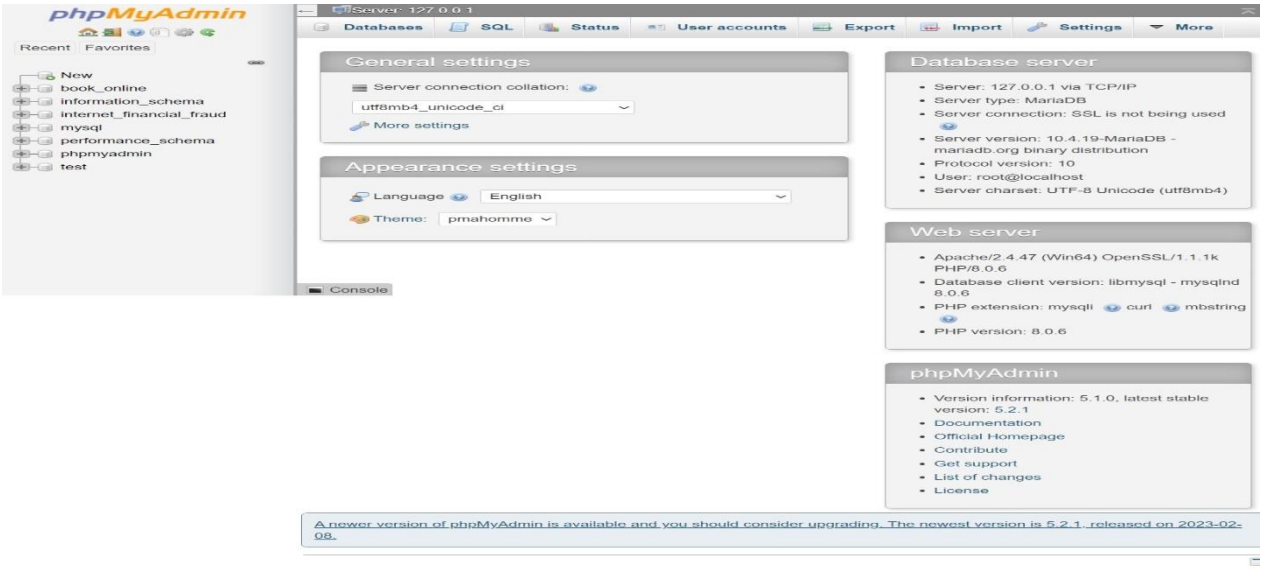

Fig:8.1. Click on import

Fig:8.2. Importing the file into the current server

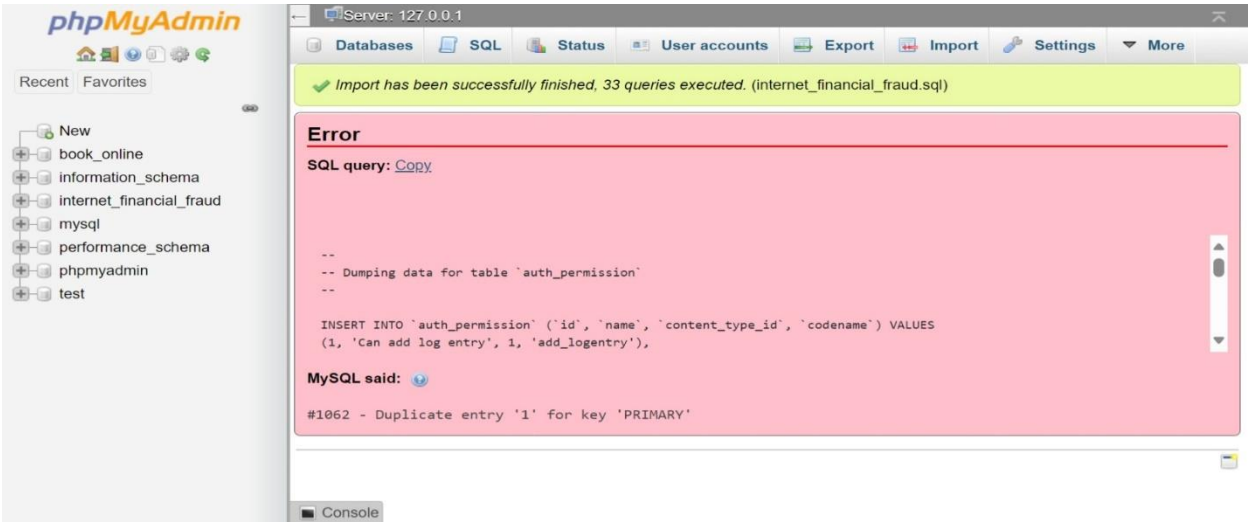

Fig:8.3. File has been successfully Imported

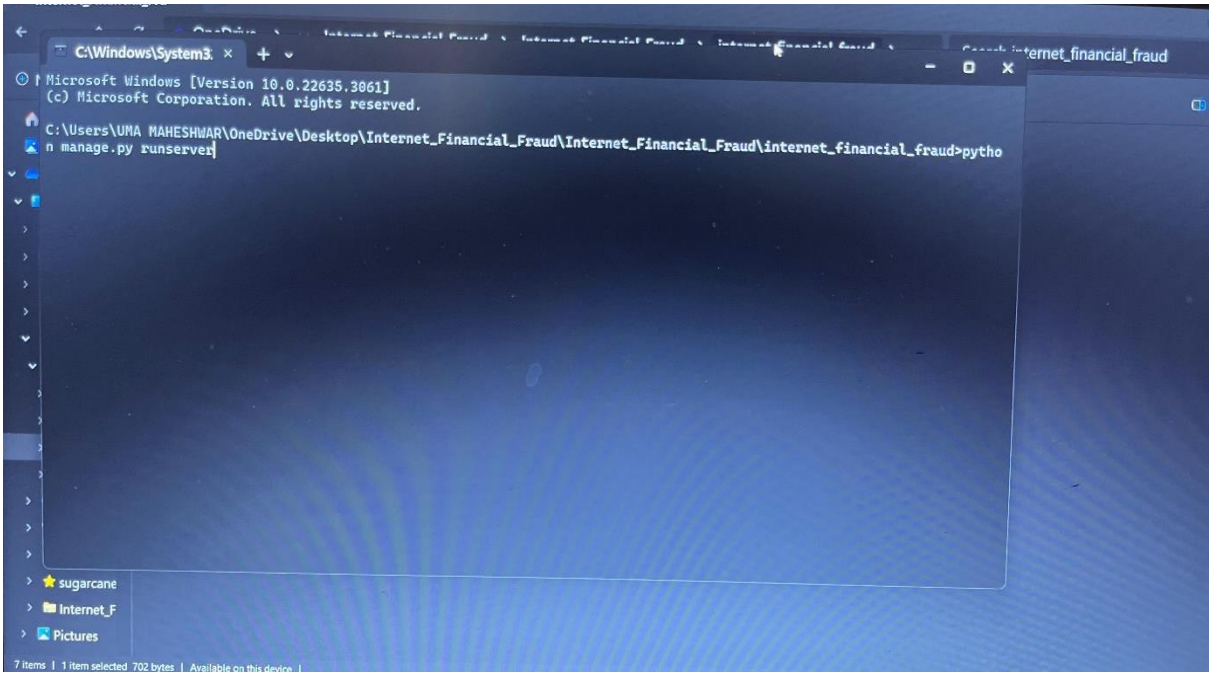

Fig:8.4. Performing system check

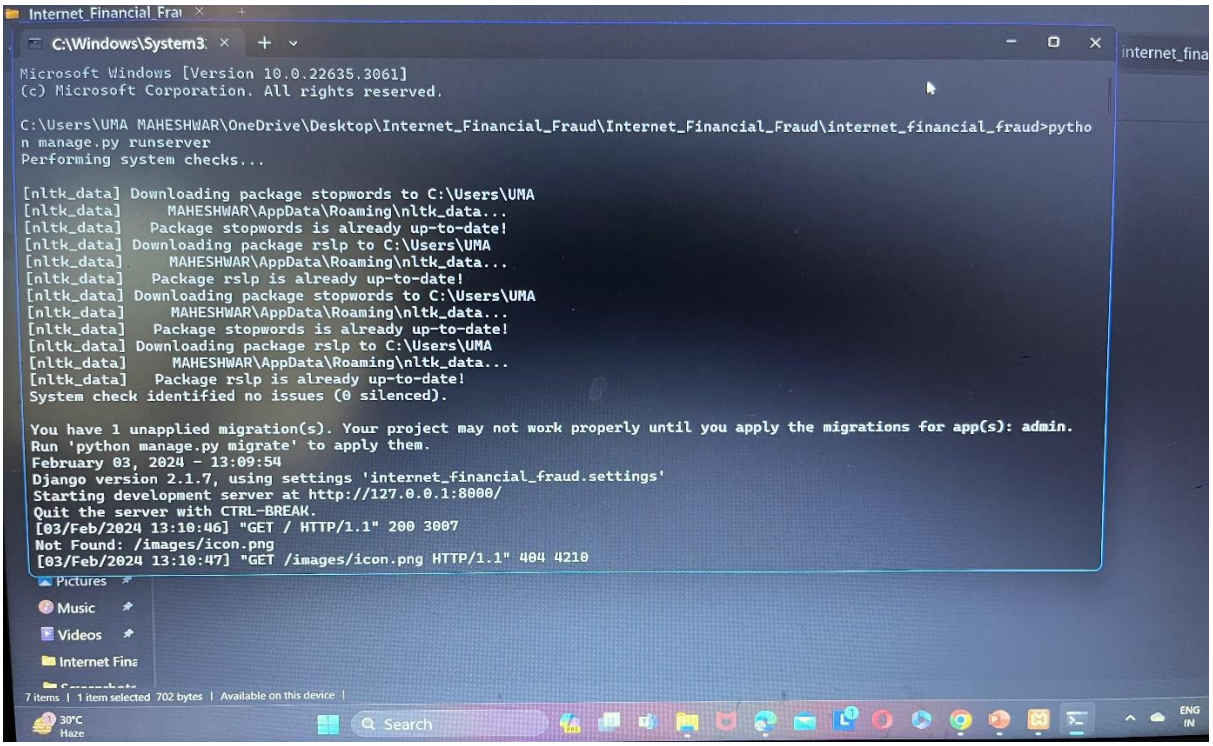

Fig:8.5. Copying the server link

Fig:8.6. Login service provider

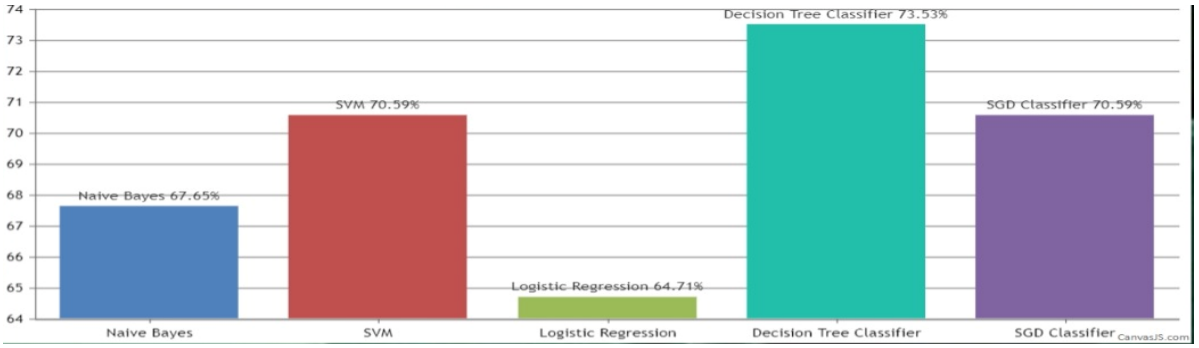

Fig: 9.1 Bar chart of all Machine learning algorithms

From above graph we can see Logistic regression and Decision Tree classifier is giving higher accuracy.

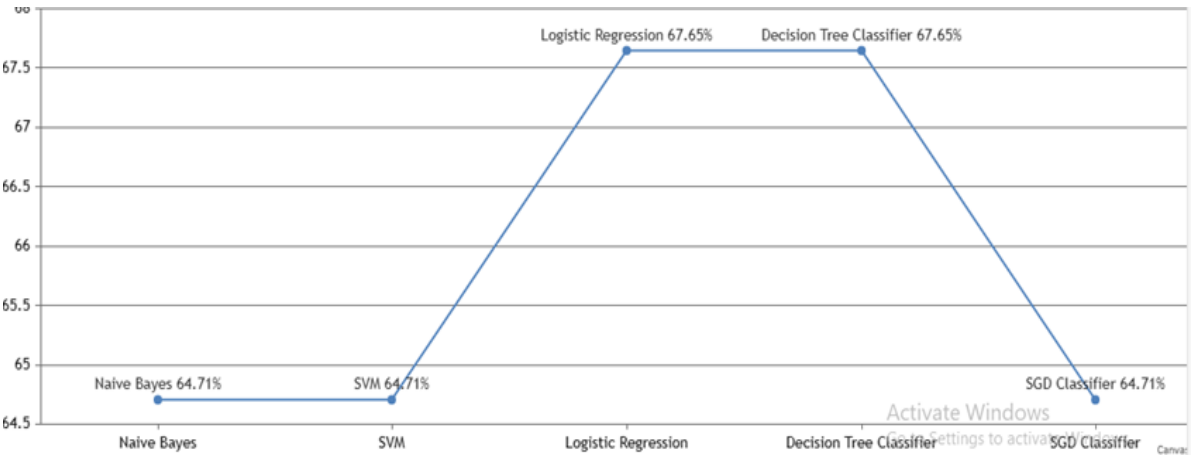

Fig: 9.2 Line chart of all Machine Learning algorithms

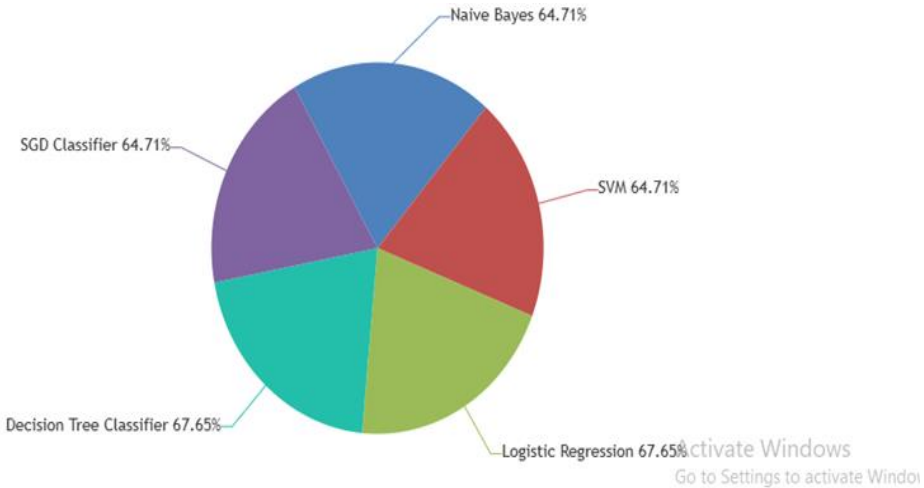

Fig:9.3 Pie chart of all Machine Learning algorithms

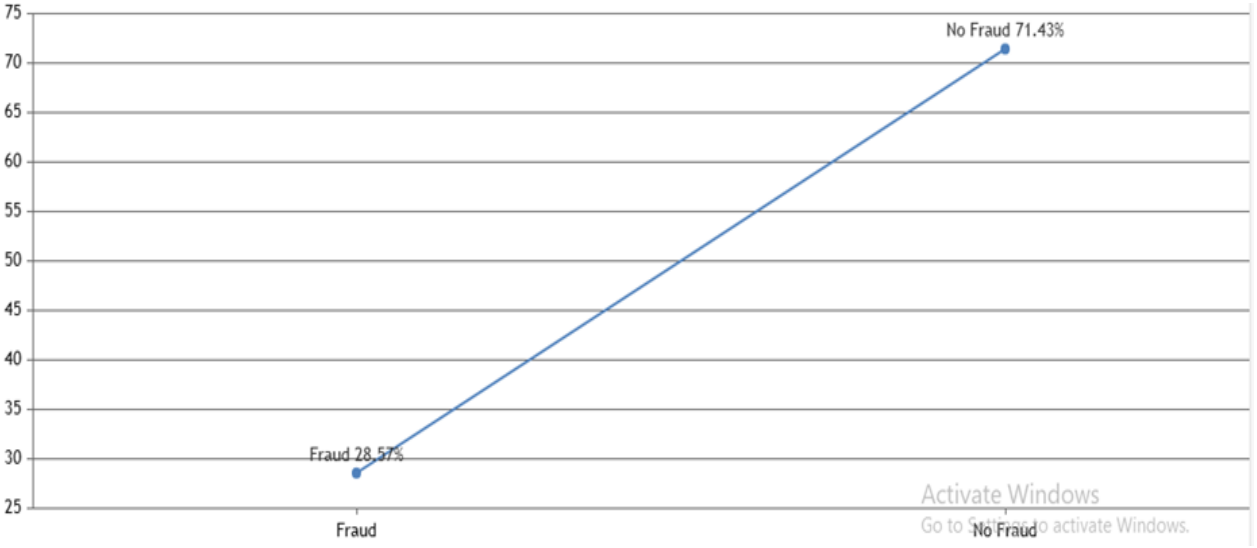

Fig:9.4 Line chart showing percentage of fraud and not fraud

Here we can see 28.57% fraud are detected and 71.43% are genuine.

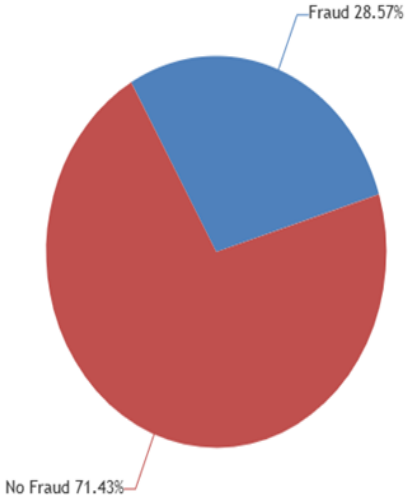

Fig:9.5 Pie chart

CHAPTER 10 : CONCLUSION AND FUTURE SCOPE

10.1 CONCLUSION:

The occurrences of Internet financial fraud cases have caused huge losses to commercial banks or financial institutions. In order to enhance the efficiency of financial fraud detections, an intelligent and distributed Big Data approach is proposed in this article. The approach mainly includes four modules: data preprocessing module, normal data feature module, graph embedding module, prediction module. The graph embedding algorithm Node2Vec is implemented on Spark Graph X and Hadoop to learn and represent the topological features of each vertex in the network graph into a low- dimensional dense vector, so as to improve the classification effectiveness of deep neural network and predict the fraudulent samples of the dataset. The experiments evaluate the indicators of precision rate, recall rate, F1-Score and F2- Score, and the results show that due to the Node2Vec properties of structural equivalence and homophily, the features of samples can be better learned and represented and the proposed approach is better than the comparative methods. In future work, the inductive graph embedding network algorithms, such as Graph Sage, Pin Sage, etc., would be improved and implemented to effectively learn the features of newly generated vertices in a dynamic network graph, so as to achieve the better effect of financial fraud detection.

10.2 FUTURE SCOPE:

Artificial Intelligence and Machine Learning can help analyse data and detect fraud patterns. Behaviour Biometrics can analyse user behaviour to identify suspicious activities. Advanced Analytics techniques can uncover complex fraud schemes. Real-time Monitoring and Automation can quickly flag and respond to suspicious activities.

CHAPTER 11: APPENDIX

Introduction

*One of the most popular languages is Python. Guido van Rossum released this language in 1991. Python is available on the Mac, Windows, and Raspberry Pi operating systems. The syntax of Python is simple and identical to that of English. When compared to Python, it was seen that the other language requires a few extra lines.

*It is an interpreter-based language because code may be run line by line after it has been written. This implies that rapid prototyping is possible across all platforms. Python is a big language with a free, binary-distributed interpreter standard library.

* It is inferior to maintenance that is conducted and is straightforward to learn. It is an object-oriented, interpreted programming language. It supports several different programming paradigms in addition to object-oriented programming, including functional and procedural programming..

Here are some key features and characteristics of Python:

- Readability: Python emphasizes code readability with its clean and intuitive syntax. It uses indentation and whitespace to structure code blocks, making it easy to understand and maintain.
- Easy to Learn: Python's simplicity and readability make it an excellent choice for beginners. Its straightforward syntax and extensive documentation make it accessible for newcomers to programming.
- Interpreted Language: Python is an interpreted language, meaning that it doesn't need to be compiled before running. The Python interpreter reads and executes the code directly, making the development process faster and more interactive.
- Cross-platform Compatibility: Python is available for major operating systems like Windows, macOS, and Linux. This cross-platform compatibility allows developers to write code once and run it on different platforms without modifications.

Installation

To install Python on your computer, follow these basic steps:

- Step 1: Visit the Python website Go to the official Python website at https://www.python.org/.
- Step 2: Select the operating system Choose the appropriate installer for your operating system. Python supports Windows, macOS, and various Linux distributions. Make sure to select the correct version that matches your operating system.
- Step 3: Check which version of Python is installed; if the 3.7.0 version is not there, uninstall it through the control panel and
- Step 4: Install Python 3.7.0 using Cmd.
- Step 5: Install the all libraries that required to run the project
- Step 6: Run

Python Features:

1) **Easy:** Because Python is a more accessible and straightforward language, Python programming is easier to learn.
2) **Interpreted language:** Python is an interpreted language, therefore it can be used to examine the code line by line and provide results.
3) **Open Source:** Python is a free online programming language since it is open-source.
4) **libraries:** Python offers a sizable library that we may utilize to create applications quickly.
5) **GUI:** It stands for GUI (Graphical User Interface)

Python GUI (Tkinter)

- Python provides a wide range of options for GUI development (Graphical User Interfaces).
- Tkinter, the most widely used GUI technique, is used for all of them.

- The Tk GUI toolkit offered by Python is used with the conventional Python interface.
- A part of Python's built-in library is Tkinter. The GUI programs were created.
- Making a GUI application is easy using Tkinter. Following are the steps:
- 1) Install the Tkinter module in place.
- 2) The GUI applicatioMakeske the primary window
- 3) Include one or more of the widgets mentioned above in the GUI application.
- 4) Set up the main event loop such that it reacts to each user-initiated event.
- Although Tkinter is the only GUI framework included in the Python standard library, Python includes a GUI framework. The default library for Python is called Tkinter. Tk is a scripting language often used in designing, testing, and developing GUIs. Tk is a free, open-source widget toolkit that may be used to build GUI applications in a wide range of computer languages.

Python IDLE

- Python IDLE offers a full-fledged file editor, which gives you the ability to write and execute Python programs from within this program. The built-in file editor also includes several features, like code completion and automatic indentation, that will speed up your coding workflow.
- Guido Van Rossum named Python after the British comedy group Monty Python while the name IDLE was chosen to pay tribute to Eric Idle, who was one of the Monty Python's founding members. IDLE comes bundled with the default implementation of the Python language since the 01.5. 2b1 release
- IDLE has two modes: interactive and script. We wrote our first program, “Hello, World!” in interactive mode. Interactive mode immediately returns the results of commands you enter into the shell. In script mode, you will write a script and then run it.

❖ Python IDLE (Integrated Development and Learning Environment) is an interactive development environment included with the Python programming language. It provides a convenient way to write, execute, and debug Python code.

When you install Python, IDLE is typically installed along with it. To open IDLE, you can follow these steps:

- Open the command prompt (Windows) or terminal (macOS/Linux).
- Type "idle" and press Enter. Alternatively, you can specify the version with "idle3" or "idle2" for Python 3 or Python 2, respectively.
- Once IDLE is launched, you will see the Python shell, which is an interactive environment where you can type and execute Python code directly.

Here are some features and functionalities provided by Python IDLE:

- Editor: IDLE includes a text editor where you can write your Python code. It offers syntax highlighting, automatic indentation, and code completion to enhance your coding experience.
- Interactive Shell: The Python shell in IDLE allows you to execute Python code interactively. You can type commands, statements, or function calls directly in the shell, and Python will execute them immediately.
- Debugging: IDLE provides basic debugging capabilities to help you find and fix errors in your code. You can set breakpoints, step through code, inspect variables, and track the program's execution.
- Python Help: IDLE provides access to the Python documentation and built-in help. You can access the help menu to find information about Python modules, functions, classes, and more.
- Customization: IDLE can be customized to suit your preferences. You can modify settings related to syntax highlighting, indentation, fonts, and more.

SOURCE CODE

```
from django.db.models import  Count, Avg
from django.shortcuts import render, redirect
from django.db.models import Count
from django.db.models import Q
import datetime
import xlwt
from django.http import HttpResponse
import numpy as np


import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from sklearn.pipeline import Pipeline

#to data preprocessing
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder

#NLP tools
import re
import nltk
nltk.download('stopwords')
nltk.download('rslp')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import CountVectorizer

#train split and fit models
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from nltk.tokenize import TweetTokenizer
from sklearn.ensemble import VotingClassifier
#model selection
from sklearn.metrics import confusion_matrix, accuracy_score,
plot_confusion_matrix, classification_report

# Create your views here.
from Remote_User.models import
ClientRegister_Model,Financial_Fraud_Prediction,detection_ratio,detecti
on_accuracy


def serviceproviderlogin(request):
    if request.method  == "POST":
        admin = request.POST.get('username')
        password = request.POST.get('password')
        if admin == "Admin" and password =="Admin":
            detection_accuracy.objects.all().delete()
            return redirect('View_Remote_Users')
```

```
    return render(request,'SProvider/serviceproviderlogin.html')

def View_Financial_Type_Ratio(request):
    detection_ratio.objects.all().delete()
    rratio = ""
    kword = 'Fraud'
    print(kword)
    obj =
Financial_Fraud_Prediction.objects.all().filter(Q(Prediction=kword))
    obj1 = Financial_Fraud_Prediction.objects.all()
    count = obj.count();
    count1 = obj1.count();
    ratio = (count / count1) * 100
    if ratio != 0:
        detection_ratio.objects.create(names=kword, ratio=ratio)

    ratio1 = ""
    kword1 = 'No Fraud'
    print(kword1)
    obj1 =
Financial_Fraud_Prediction.objects.all().filter(Q(Prediction=kword1))
    obj11 = Financial_Fraud_Prediction.objects.all()
    count1 = obj1.count();
    count11 = obj11.count();
    ratio1 = (count1 / count11) * 100
    if ratio1 != 0:
        detection_ratio.objects.create(names=kword1, ratio=ratio1)


    obj = detection_ratio.objects.all()
    return render(request, 'SProvider/View_Financial_Type_Ratio.html',
{'objs': obj})

def View_Remote_Users(request):
    obj=ClientRegister_Model.objects.all()
    return
render(request,'SProvider/View_Remote_Users.html',{'objects':obj})

def ViewTrendings(request):
    topic =
Financial_Fraud_Prediction.objects.values('topics').annotate(dcount=Cou
nt('topics')).order_by('-dcount')
    return
render(request,'SProvider/ViewTrendings.html',{'objects':topic})

def charts(request,chart_type):
    chart1 =
detection_ratio.objects.values('names').annotate(dcount=Avg('ratio'))
    return render(request,"SProvider/charts.html", {'form':chart1,
'chart_type':chart_type})

def charts1(request,chart_type):
    chart1 =
detection_accuracy.objects.values('names').annotate(dcount=Avg('ratio')
)
```

```
    return render(request,"SProvider/charts1.html", {'form':chart1,
'chart_type':chart_type})

def View_Prediction_Of_Financial_Type(request):
    obj =Financial_Fraud_Prediction.objects.all()
    return render(request,
'SProvider/View_Prediction_Of_Financial_Type.html', {'list_objects':
obj})

def likeschart(request,like_chart):
    charts
=detection_accuracy.objects.values('names').annotate(dcount=Avg('ratio'
))
    return render(request,"SProvider/likeschart.html", {'form':charts,
'like_chart':like_chart})


def Download_Trained_DataSets(request):

    response = HttpResponse(content_type='application/ms-excel')
    # decide file name
    response['Content-Disposition'] = 'attachment;
filename="Predicted_Data.xls"'
    # creating workbook
    wb = xlwt.Workbook(encoding='utf-8')
    # adding sheet
    ws = wb.add_sheet("sheet1")
    # Sheet header, first row
    row_num = 0
    font_style = xlwt.XFStyle()
    # headers are bold
    font_style.font.bold = True
    # writer = csv.writer(response)
    obj = Financial_Fraud_Prediction.objects.all()
    data = obj  # dummy method to fetch data.
    for my_row in data:
        row_num = row_num + 1
        ws.write(row_num, 0, my_row.Customer_Email, font_style)
        ws.write(row_num, 1, my_row.customerPhone, font_style)
        ws.write(row_num, 2, my_row.customerDevice, font_style)
        ws.write(row_num, 3, my_row.customerIPAddress, font_style)
        ws.write(row_num, 4, my_row.customerBillingAddress, font_style)
        ws.write(row_num, 5, my_row.No_Transactions, font_style)
        ws.write(row_num, 6, my_row.No_Orders, font_style)
        ws.write(row_num, 7, my_row.No_Payments, font_style)
        ws.write(row_num, 8, my_row.Prediction, font_style)


    wb.save(response)
    return response

def train_model(request):
    detection_accuracy.objects.all().delete()
    dataset = pd.read_csv('Transaction_Datasets.csv', encoding='latin-
1')
```

```
    dataset.rename(columns={'Fraud': 'label', 'customerBillingAddress':
'caddress'}, inplace=True)

    def apply_results(label):
        if (label == 0):
            return 0  # False
        elif (label == 1):
            return 1  # True

    dataset['results'] = dataset['label'].apply(apply_results)
    dataset.drop(['label'], axis=1, inplace=True)
    results = dataset['results'].value_counts()

    cv = CountVectorizer()

    x = dataset["caddress"]
    y = dataset["results"]

    # x = cv.fit_transform(x)

    x = cv.fit_transform(dataset['caddress'].apply(lambda x:
np.str_(x)))

    models = []
    from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(x, y,
test_size=0.20)
    X_train.shape, X_test.shape, y_train.shape

    print("Naive Bayes")

    from sklearn.naive_bayes import MultinomialNB

    NB = MultinomialNB()
    NB.fit(X_train, y_train)
    predict_nb = NB.predict(X_test)
    naivebayes = accuracy_score(y_test, predict_nb) * 100
    print("ACCURACY")
    print(naivebayes)
    print("CLASSIFICATION REPORT")
    print(classification_report(y_test, predict_nb))
    print("CONFUSION MATRIX")
    print(confusion_matrix(y_test, predict_nb))
    detection_accuracy.objects.create(names="Naive Bayes",
ratio=naivebayes)

    # SVM Model
    print("SVM")
    from sklearn import svm

    lin_clf = svm.LinearSVC()
    lin_clf.fit(X_train, y_train)
    predict_svm = lin_clf.predict(X_test)
    svm_acc = accuracy_score(y_test, predict_svm) * 100
    print("ACCURACY")
    print(svm_acc)
    print("CLASSIFICATION REPORT")
```

```
    print(classification_report(y_test, predict_svm))
    print("CONFUSION MATRIX")
    print(confusion_matrix(y_test, predict_svm))
    detection_accuracy.objects.create(names="SVM", ratio=svm_acc)

    print("Logistic Regression")

    from sklearn.linear_model import LogisticRegression

    reg = LogisticRegression(random_state=0,
solver='lbfgs').fit(X_train, y_train)
    y_pred = reg.predict(X_test)
    print("ACCURACY")
    print(accuracy_score(y_test, y_pred) * 100)
    print("CLASSIFICATION REPORT")
    print(classification_report(y_test, y_pred))
    print("CONFUSION MATRIX")
    print(confusion_matrix(y_test, y_pred))
    detection_accuracy.objects.create(names="Logistic Regression",
ratio=accuracy_score(y_test, y_pred) * 100)

    print("Decision Tree Classifier")
    dtc = DecisionTreeClassifier()
    dtc.fit(X_train, y_train)
    dtcpredict = dtc.predict(X_test)
    print("ACCURACY")
    print(accuracy_score(y_test, dtcpredict) * 100)
    print("CLASSIFICATION REPORT")
    print(classification_report(y_test, dtcpredict))
    print("CONFUSION MATRIX")
    print(confusion_matrix(y_test, dtcpredict))
    detection_accuracy.objects.create(names="Decision Tree Classifier",
ratio=accuracy_score(y_test, dtcpredict) * 100)

    print("SGD Classifier")
    from sklearn.linear_model import SGDClassifier
    sgd_clf = SGDClassifier(loss='hinge', penalty='l2', random_state=0)
    sgd_clf.fit(X_train, y_train)
    sgdpredict = sgd_clf.predict(X_test)
    print("ACCURACY")
    print(accuracy_score(y_test, sgdpredict) * 100)
    print("CLASSIFICATION REPORT")
    print(classification_report(y_test, sgdpredict))
    print("CONFUSION MATRIX")
    print(confusion_matrix(y_test, sgdpredict))
    detection_accuracy.objects.create(names="SGD Classifier",
ratio=accuracy_score(y_test, sgdpredict) * 100)

    labeled = 'Processed_data.csv'
    dataset.to_csv(labeled, index=False)
    dataset.to_markdown

    obj = detection_accuracy.objects.all()
    return render(request,'SProvider/train_model.html', {'objs': obj})
```

CHAPTER 12: BIBLIOGRAPHY

1. U. Paschen, C. Pitt, and J. Kietzmann, "Artificial intelligence: building blocks and an innovation typology," *Business Horizons*, vol. 63, no. 2, pp. 147-155, 2020.
2. P. Yu, Z. Xia, J. Fei, and S. K. Jha, "An application review of artificial inteligence in prevention and cure of COVID-19 pandemic," *CMC-Computers Materials & Continua*, vol. 65, no. 1, pp. 743-760, 2020.
3. L. Shen, X. Chen, Z. Pan, K. Fan, F. Li, and J. Lei, "No-reference stereoscopic image quality assessment based on global and local content characteristics," *Neurocomputing*, vol. 424, no. 2, pp. 132- 142, 2021.
4. H. Beck, "Banking is essential, banks are not, the future of financial intermediation in the age of the Internet," *Netnomics*, vol. 3, no. 1, pp. 7-22, 2001.
5. G. N. Weiss, K. Pelger, and A. Horsch, "Mitigating adverse selection in p2p lending–Empirical evidence from prosper.com," *SSRN Electronic Journal*, vol. 19, no. 7, pp. 65-93, 2010.
6. Y. Houston, C. Jongrong , J. H. Cliff, and H. Y. Chih, "E-commerce, R&D, and productivity: firm-level evidence from Taiwan," *Information Economics and Policy*, vol. 18, no. 5, pp. 561-569, 2013.
7. F. Allen, J. Mcandrews, and P. Strahan, "E-finance: an introduction," *Center for Financial Institutions Working Papers*, vol. 22, no. 1, pp.
8. J. A. Kregel, "Margins of safety and weight of the argument in generating financial fragility," *Journal of Economics Issues*, vol. 6, no. 31, pp. 543-548, 2016.
9. A. Momparler, C. Lassala, and D. Ribeiro, "Efficiency in banking services: a comparative analysis of Internet-primary and branching banks in the US," *Service Business*, vol. 7, no. 4, pp. 641-663, 2013.
10. V.Jambulapati and J. Stavins, "Credit card act of 2009: what did banks do?," *Banking & Finance*, vol. 46, no. 9, pp. 21-30, 2014.
11. H. Shefrin and C. M. Nicols, "Credit card behavior, financial styles and heuristics," *Business Research*, vol. 67, no. 8, pp. 1679-1687, 2014.

12. C. B. Hem and D. A. Ficawoyi, “Internet consumer spending and credit card balance: evidence from US consumers,” *Review of Financial Economics*, vol. 30, no. 9, pp. 11-22, 2016.
13. D. Andrew and K. Jiseob, “Explaining changes in the US credit card market: lenders are using more information,” *Economic Modelling*, vol. 61, no. 2, pp. 76-92, 2017.
14. D. A. Ficawoyi and C. B. Hem, “Credit card delinquency: how much is the Internet to blame?,” *North American Journal of Economics and Finance*, vol. 48, no. 4, pp. 481-497, 2019.
15. P. Giudici, M. B. Hadji, and A. Spelta, “Network based credit risk models,” *Quality Engineering*, vol. 32, no. 2, pp. 199-211, 2020.