

FRAUD PREDICTION AND VERIFICATION OF SMART CREDIT CARD USING MACHINE LEARNING TECHNIQUES

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Engineering degree in Computer Science and
Engineering

By

Prajapnoor Baswaraj (Reg .No 40110161)
Praveen Kumar (Reg. No-40110137)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Category-1 University by UGC

Accredited with Grade “A++” by NAAC

[12B Status by UGC] Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHISALAI,

CHENNAI – 600119

APRIL – 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Prajapnoor Baswaraj (40110161)** and **Praveen Kumar (40110137)** who carried out the Project entitled **“FRAUD PREDICTION AND VERIFICATION OF SMART CREDIT CARD USING MACHINE LEARNING TECHNIQUES”** under my supervision from November 2023 to April 2024.

Internal Guide

Dr. B. U. ANU BARATHI M.E., Ph.D.

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **Prajapnoor Baswaraj (Reg.No- 40110161)**, hereby declare that the Project Report entitled **FRAUD PREDICTION AND VERIFICATION OF SMART CREDIT CARD USING MACHINE LEARNING TECHNIQUES** done by me under the guidance of **Dr. B.U.ANU BARATHI M.E., Ph.D.** is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE:

PLACE: Chennai

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to The **Board of Management of Sathyabama Institute of Science and Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D, Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. B. U. ANU BARATHI M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

This study presents a cutting-edge approach to enhancing the security of smart credit card transactions through advanced fraud detection and verification techniques. The system's architecture encompasses a comprehensive pipeline, including data preprocessing, feature engineering, and real-time prediction, leveraging a hybrid model that combines supervised machine learning algorithms, an encoder, and Long Short-Term Memory (LSTM) networks. In this system, a supervised LSTM network is responsible for categorizing transactions, while an unsupervised Autoencoder is employed to identify any unusual or outlier activities. The assessment criteria employed strike a careful balance between recall and accuracy, ensuring that fraudulent transactions are accurately identified without an overwhelming number of false positives. One of the system's key strengths lies in its real-time capabilities, swiftly alerting relevant parties upon the detection of fraudulent activities, thereby preventing potential financial losses. Additionally, this system places significant emphasis on compliance with regulatory standards, scalability to accommodate growing transaction volumes, and continuous monitoring to adapt to evolving fraud tactics. By bridging the gap between convenience and security in modern financial transactions, this project offers a state-of-the-art solution to fortify the safety of smart credit card usage. Its innovative combination of supervised and unsupervised learning techniques, along with real-time operation, positions it at the forefront of smart credit card fraud detection and prevention, ultimately safeguarding the interests of cardholders and financial institutions alike.

TABLE OF CONTENTS

Chapter No	TITLE		Page No.
	ABSTRACT		v
	LIST OF FIGURES		viii
1	INTRODUCTION		1
2	LITERATURE SURVEY		3
	2.1	Inferences from Literature Survey	3
	2.2	Existing System and Proposed System	6
	2.3	Open Problems in Existing System	7
3	REQUIREMENTS ANALYSIS		9
	3.1	Risk Analysis of the Project	9
	3.2	Software Requirements Specification Document	12
4	DESCRIPTION OF PROPOSED SYSTEM		13
	4.1	Selected Methodology or process model	13
	4.2	Architecture / Overall Design of Proposed System	17
	4.3	Description of Software for Implementation and Testing plan of the Proposed Model/System	18
	4.4	Project Management Plan	20
5	IMPLEMENTATION DETAILS		21

6	RESULTS AND DISSCUSSION		24
	6.1	Performance Metrics	25
	6.2	Insights from the Model	26
	6.3	Comparison with Baseline Models	26
	6.4	Real-World Implications and Challenges	27
	6.5	Ethical Considerations	28
	6.6	Future Directions	28
7	CONCLUSION		30
	7.1	Conclusion	30
	7.2	Future work	31
	7.3	Research Issues	31
	7.4	Implementation Issues	32
	REFERENCES		33
	APPENDIX		35
	A.	Source Code	35
	B.	Screenshots	49
	C.	Research Paper	50
	D.	Certificate	59

LIST OF FIGURES

FIGURE NO	FIGURE NAME	Page No.
4.1	System Architecture	20
5.1	LSTM Loss Function	25

LIST OF ABBREVIATIONS

ABBREVIATIONS	EXPANSION
LSTM	Long Short Term Memory
RNN	Recurrent Neural Network
ReLU	Rectified Linear Unit
PCA	Principal Component Analysis
API	Application Programming Interface
CSV	Comma Separated Values

CHAPTER 1

INTRODUCTION

Credit card fraud is a significant concern for financial institutions and consumers alike. With the rapid advancement of technology, fraudulent activities have become more sophisticated and challenging to detect. As a result, the need to develop efficient and accurate prediction models to identify and prevent credit card fraud has become paramount. In recent years, supervised machine learning techniques have emerged as valuable tools in addressing this problem. This research project aims to integrate advanced feature engineering and ensembling to improve accuracy, robustness, false-positive rate, Integration with decision making systems adapt to evolving fraud patterns, resulting in more precise fraud detection. The introduction of smart credit cards, a new age of frictionless and speedy monetary transactions has begun. The use of these highly developed payment systems has simplified the day-to-day financial transactions for both consumers and enterprises. But with the proliferation of smart credit cards comes a new and ever-present danger: credit card theft. The ever-evolving methods used by criminals to take advantage of loopholes in the system make combating credit card theft an enormous problem. Successfully countering this threat requires the development of novel and flexible solutions. This study addresses this urgent requirement by introducing a thorough method for smart credit card fraud prediction and verification using hybrid machine learning. An extensive dataset including a variety of smart credit card transactions is first collected and pre-processed as part of the project's multi-faceted approach. The dataset contains both valid and fraudulent transactions, making it a valuable resource for building models. An unsupervised Autoencoder neural network is a key component of this study because of how well it detects outliers in transaction data. Using the encoder component of the Autoencoder to learn latent characteristics, this method can identify fraud even when it's not immediately obvious. Supervised machine learning methods, including LSTM networks—well-suited to sequence data—are used in the research to improve prediction accuracy. To improve the model's capacity to differentiate between real and fraudulent transactions, the LSTM network is taught using the encoded characteristics obtained by the Autoencoder. To guarantee that this hybrid model can with stand the

rigors of real-world use, it is tested extensively utilizing a wide range of performance indicators. Finding the sweet spot between detecting a large number of fraudulent transactions and avoiding false positives requires optimizing recall and precision criteria. Implementation in real time is one of the main objectives of the project. The technology is built to handle credit card transactions in real-time, responding instantly to any signs of possible fraud. To reduce losses and safeguard cardholders, alerts may be set up and predetermined actions can be executed. The project's concept is based on the idea of continuous monitoring and modification. The ever-changing nature of fraud tendencies is recognized, highlighting the need of continuously improving models and updating data. To further guarantee the system's ethical and legal operation, compliance with data privacy laws and regulatory frameworks is an essential factor to consider. A major step forward in smart credit card security is essentially what this initiative is all about. In an ever-more-digital and linked world, it aims to secure cardholders and financial institutions by combining state-of-the-art machine learning methods with real-time capabilities, striking a careful balance between the two.

In conclusion, credit card fraud prediction is a critical task that can be effectively addressed by using hybrid fusion machine learning technique, financial institutions and consumers can better safeguard against credit card fraud and protect their financial assets.

CHAPTER 2

LITERATURE SURVEY

2.1 INFERENCES FROM LITERATURE SURVEY

1. Li, Z., Zhang, H., Masum, M., Shahriar, H., & Haddad, H. (2020, April). Cyber fraud prediction with supervised machine learning techniques. In Proceedings of the 2020 ACM Southeast Conference (pp. 176-180).

The study conducted by Li, Zhang, Masum, Shahriar, and Haddad (2020) focuses on the prediction of cyber fraud using supervised machine learning techniques. Presented at the 2020 ACM Southeast Conference, the study utilizes credit card fraud as a specific case for prediction. The authors explored various supervised machine learning techniques and analyzed their effectiveness in identifying credit card fraud. The findings and implications of this research provide valuable insights for developing robust systems to prevent and detect fraudulent activities.

2. More, R., Awati, C., Shirgave, S., Deshmukh, R., & Patil, S. (2021). Credit card fraud detection using supervised learning approach. International journal of scientific & technology research, 9(10), 216-219.

The study conducted by More, Awati, Shirgave, Deshmukh, and Patil (2021) focuses on credit card fraud detection using a supervised learning approach. The authors explore various supervised machine learning techniques to predict and prevent credit card fraud. Their research aims to enhance the accuracy and efficiency of credit card fraud detection systems, ultimately protecting users from fraudulent transactions. The findings of their study provide valuable insights for financial institutions and organizations in developing robust fraud detection systems.

3. Khatri, S., Arora, A., & Agrawal, A. P. (2020, January). Supervised machine learning algorithms for credit card fraud detection: a comparison. In 2020 10th international conference on cloud computing, data science & engineering (confluence) (pp. 680-683). IEEE.

Khatri, Arora, and Agrawal (2020) conducted a study comparing supervised machine learning algorithms for credit card fraud detection. The researchers presented their findings in a paper titled "Supervised machine learning algorithms for credit card fraud detection: a comparison." The study was published in the International Conference on Cloud Computing, Data Science, and Engineering in 2020. The paper compared

various supervised machine learning techniques for the prediction of credit card fraud.

4. Caroline Cynthia, P., & Thomas George, S. (2021). An outlier detection approach on credit card fraud detection using machine learning: a comparative analysis on supervised and unsupervised learning. In Intelligence in Big Data Technologies—Beyond the Hype: Proceedings of ICBDDC 2019 (pp. 125-135). Springer Singapore.

Caroline Cynthia and Thomas George (2021) conducted a comparative analysis on the use of supervised machine learning techniques for credit card fraud detection. In their study, they proposed an outlier detection approach to identify fraudulent transactions. The researchers explored different supervised learning algorithms and evaluated their performance in detecting credit card fraud. Their findings provide insights into the effectiveness of supervised machine learning techniques for predicting fraudulent activities in the credit card industry.

5. Saheed, Y. K., Baba, U. A., & Raji, M. A. (2022). Big Data Analytics for Credit Card Fraud Detection Using Supervised Machine Learning Models. In Big Data Analytics in the Insurance Market (pp. 31-56). Emerald Publishing Limited.

In their book chapter, Saheed, Y. K., Baba, U. A., & Raji, M. A. (2022) explore the application of big data analytics for credit card fraud detection using supervised machine learning models. The authors discuss the use of various techniques and algorithms to develop predictive models for detecting fraudulent credit card transactions. They emphasize the importance of supervised machine learning techniques in improving fraud detection accuracy and propose a framework that can be used by financial institutions to enhance their fraud prevention strategies. Their research contributes to the advancement of credit card fraud detection using state-of-the-art machine learning approaches.

6. Hodge, V. J., & Austin, J. (2004, June). A survey of outlier detection methodologies. Artificial Intelligence Review, 22(2), 85-126.

Hodge and Austin (2004) conducted a survey that focused on outlier detection methodologies. The study specifically aimed to gather information on the various techniques used in identifying outliers. The review examined techniques and algorithms commonly employed in the field of artificial intelligence. It provided a comprehensive analysis, covering key aspects such as data preprocessing, similarity metrics, and visualization methods.

7. Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000, May). LOF: Identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD international conference on Management of data (pp. 93-104).

Breunig, Kriegel, Ng, and Sander (2000) propose a method called LOF (Local Outlier Factor) for identifying local outliers based on density. Their research, presented at the 2000 ACM SIGMOD international conference, offers insights into effective credit card fraud prediction using supervised machine learning techniques. The authors discuss how density-based methods can help identify anomalies in credit card transactions, providing a valuable tool for improving fraud detection and prevention strategies in the banking and financial industry.

8. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August). "Why should I trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135-1144).

Ribeiro, Singh, and Guestrin (2016) present a paper titled "Why should I trust you?" Explaining the predictions of any classifier. In this paper, the authors propose a framework to provide explanations for the predictions made by any classifier. The framework aims to enhance trust and interpretability in machine learning models by generating personalized explanations. The authors demonstrate the effectiveness of their approach in the context of credit card fraud prediction, using supervised machine learning techniques.

9. Tang, Y., Zhou, S., & Luo, J. (2016, December). Credit card fraud detection with imbalanced dataset. In 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA) (pp. 198-207).

In their paper titled "Credit card fraud detection with imbalanced dataset," Tang, Zhou, and Luo (2016) explored the use of supervised machine learning techniques for predicting credit card fraud. The authors presented their findings at the 2016 IEEE International Conference on Data Science and Advanced Analytics. Their research focused on addressing the challenges posed by imbalanced datasets in credit card fraud detection and proposed various techniques to overcome these challenges. Ultimately, their study aimed to enhance the accuracy and efficiency of credit card

fraud detection systems through the application of supervised machine learning algorithms.

10. Phua, C., Lee, V., Smith, K., & Gayler, R. (2005, October). A comprehensive survey of data mining-based fraud detection research. *Artificial Intelligence Review*, 24(4), 347-373

The article by Phua et al. (2005) provides a thorough examination of the current state of data mining-based fraud detection research. The authors focus specifically on credit card fraud prediction using supervised machine learning techniques. The survey explores various methods and algorithms employed, such as logistic regression, decision trees, neural networks, and support vector machines. The article also discusses the advantages and limitations of these techniques and provides insights into future research directions in the field of credit card fraud detection.

2.2 EXISTING SYSTEM

There are a number of drawbacks to the current smart credit card verification method that reduce its effectiveness and efficiency. The existing system, for starters, mainly relies on magnetic stripe technology, which is prone to data theft and cloning. The cardholders are at a high risk of fraud and illegal transactions since the information contained on the magnetic stripe is readily available and copyable. The current credit card fraud prediction system in banking support relies on basic rule-based methods and simple anomaly detection techniques. This approach lacks the ability to effectively capture complex patterns in fraudulent activities, leading to high false positives and negatives.

PROPOSED SYSTEM

To detect and prevent smart credit card fraud, this study introduces a hybrid approach that integrates supervised and unsupervised machine learning techniques., Autoencoder, and LSTM networks. Prior to unsupervised anomaly detection using an Autoencoder, data is pre-processed and features are engineered. The next step in using encoded data for transaction classification is training a supervised LSTM network. With real-time deployment, fraud may be detected immediately and notifications can be triggered if needed. Important characteristics include scalability, regulatory compliance, and continuous monitoring and modification. Securing

customers and financial institutions against the ever-changing danger of credit card fraud, this suggested system offers a strong solution that combines security with simplicity.

In conclusion, the proposed system is the fusion of Autoencoder and LSTM networks. It integrates advanced feature engineering and ensembling to improve accuracy, robustness, false-positive rate, Integration with decision making systems adapt to evolving fraud patterns, resulting in more precise fraud detection.

2.3 OPEN PROBLEMS IN EXISTING SYSTEM

The hybrid fusion algorithm, combining autoencoder and LSTM, can address specific challenges in credit card fraud prediction systems. Here are open problems in existing credit card fraud detection systems where such a hybrid model could offer solutions:

Handling Imbalanced Datasets: Credit card fraud datasets often exhibit a severe class imbalance. Developing effective techniques within the hybrid model to handle this imbalance and prevent biased learning is crucial for improving overall prediction performance.

Adaptability to Evolving Fraud Patterns: Credit card fraud patterns continuously evolve. The hybrid model's ability to adapt and learn from changing data distributions is essential for detecting novel fraud techniques that may not be captured by traditional methods.

Sequential Fraud Detection: Credit card transactions occur sequentially, and fraudulent activities may exhibit temporal dependencies. The LSTM component of the hybrid model can effectively capture these sequential patterns, improving the accuracy of detecting subtle fraudulent behaviors.

Feature Learning for Anomaly Detection: The autoencoder in the hybrid model can aid in unsupervised feature learning. Integrating this capability helps the model automatically identify relevant features in credit card transactions, enhancing anomaly detection in real-world scenarios.

Reducing False Positives: Minimizing false positives is critical to avoid inconveniencing legitimate cardholders. The hybrid model's ability to learn intricate patterns can contribute to reducing false positives by distinguishing normal variations from actual fraud.

Real-time Fraud Detection: Traditional fraud detection systems may face challenges in processing transactions in real-time. The hybrid fusion algorithm, with its efficient autoencoder-LSTM architecture, can enhance real-time processing capabilities, providing quicker responses to potential fraud.

Unsupervised Learning for Unknown Fraud Patterns: Credit card fraud can take on various forms, including previously unseen patterns. The unsupervised learning capability of the autoencoder allows the hybrid model to identify anomalies even when the fraud patterns are not explicitly labeled in the training data.

Cross-Channel Fraud Detection: Fraud may involve multiple channels or types of transactions. The hybrid model's ability to capture cross-channel dependencies and complex relationships between different transaction types can improve the detection of sophisticated fraud schemes.

Limited Labelled Data: Annotating fraud data for supervised learning is often challenging due to the scarcity of labeled instances. The hybrid model's unsupervised component enables it to learn from both labeled and unlabeled data, addressing the issue of limited labeled samples.

Explainability and Trustworthiness: Interpreting model decisions is crucial in fraud detection. Enhancing the explainability of the hybrid model's predictions can contribute to building trust among stakeholders and regulatory bodies, facilitating the deployment of the model in real-world financial systems.

By addressing these open problems, the hybrid fusion algorithm can significantly improve the effectiveness and efficiency of credit card fraud prediction systems, providing more accurate and adaptive solutions to combat fraudulent activities.

CHAPTER 3

REQUIREMENTS ANALYSIS

3.1 Feasibility Study

The feasibility of the project is server performance increase in this phase, and a business proposal is put forth with a very general plan for the project and fraud prediction cost estimates. During system analysis, the feasibility study of the proposed system is to be carried out. For feasibility analysis, fraud prediction understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- Economical feasibility
- Technical feasibility
- Operational feasibility

ECONOMICAL FEASIBILITY

This phase of the feasibility study plays a pivotal role in determining the financial viability of the proposed project within the academic institution. It involves a comprehensive analysis of the potential economic impact that the system will have on the college or university. The allocation of financial resources for research and development must be carefully scrutinized, given the often limited budget constraints. Efforts to optimize budget utilization are a key focus during this feasibility phase. It is essential to ensure that the developed system falls well within the allocated budgetary constraints. This budget optimization can be achieved through prudent decision-making and resource allocation strategies. Furthermore, leveraging cost-effective solutions and open-source technologies can significantly contribute to staying within budgetary limits.

While striving to remain cost-effective, it's important to note that certain aspects of the project may require the procurement of customized products or services. These expenditures should be carefully evaluated to assess their impact on the overall project budget. Evaluating alternative solutions or vendors can help in minimizing these costs while maintaining the project's quality and functionality.

In addition to budgetary considerations, a critical aspect of economic feasibility is the

evaluation of the project's potential return on investment (ROI). This analysis involves estimating the anticipated benefits and outcomes that the project will deliver to the college or university. Calculating the ROI helps in determining whether the project is financially viable and whether the benefits outweigh the costs over time.

Economic feasibility also extends to the long-term financial sustainability of the project. Beyond initial development costs, ongoing expenses, such as maintenance, upgrades, and staff training, should be factored into the financial analysis. Ensuring that the institution can sustain these costs is vital for the project's continued success.

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 being placed on the client. The developed system must have modest requirements, as only minimal or null changes are required for implementing this system.

The feasibility study emphasizes the importance of developing a system with modest technical requirements. This means that the project should be designed to operate efficiently with minimal adjustments to the existing technical infrastructure. Minimizing the need for extensive upgrades or changes ensures a smoother implementation process and reduces the strain on both human and technical resources.

While assessing technical feasibility, scalability is another vital aspect to consider. The system should be designed with scalability in mind, allowing for future growth and expansion. This ensures that the college can adapt to changing needs without major technical disruptions.

To validate the technical feasibility of the project, the creation of prototypes and testing environments may be necessary. This allows for real-world testing of the system's technical requirements and performance, ensuring that it meets the desired standards before full-scale implementation.

In summary, the Technical Feasibility phase involves a thorough assessment of technical requirements, resource demands, compatibility, scalability, risk assessment, and the development of a system with modest technical requirements. This diligent examination helps guarantee that the college project can be successfully integrated

into the existing technical infrastructure without undue strain and disruptions.

OPERATIONAL FEASIBILITY

The aspect of the 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, they 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 them familiar with it. Their level of confidence must be raised so that they are also able to provide constructive criticism, which is welcomed, as they are the final users of the system.

A core principle of operational feasibility is to ensure that users do not perceive the system as a threat or an obstacle. Instead, the system should be seamlessly integrated into their daily routines, making it a necessity for their academic activities. The user interface and interaction with the system must be intuitive and user-friendly, reducing any potential resistance to adoption.

Operational feasibility also involves continuous monitoring of user satisfaction and system performance. Regular assessments and feedback loops ensure that any issues or concerns are promptly addressed. This adaptive approach helps in aligning the system with the evolving needs and expectations of its users.

In summary, Operational Feasibility centres on user acceptance, efficient training, user-friendliness, user education, openness to constructive criticism, user confidence building, and continuous monitoring and adaptation. Ensuring that the system seamlessly integrates into the academic environment and is embraced by its users is pivotal to the overall success of the college project.

3.2 Software Requirements Specification Document

Hardware specifications:

Microsoft Server enabled computers, preferably workstations

Higher RAM, of about 4GB or above

Processor of frequency 1.5GHz or above

Software specifications:

Python 3.6 and higher

Anaconda software

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

4.1 Selected Methodology or process model

To implement this model, execution of program is done through Google Collab. Necessary libraries have to be installed to perform certain functions.

DESCRIPTION OF PROGRAMMING LANGUAGE AND SOFTWARE

PYTHON

Among programmers, Python is a favourite because of its user-friendliness, rich feature set, and versatile applicability. Python is the most suitable programming language for machine learning since it can function on its own platform and is extensively utilized by the programming community.

Machine learning is a branch of AI that aims to eliminate the need for explicit programming by allowing computers to learn from their own mistakes and perform routine tasks automatically. However, "artificial intelligence" (AI) encompasses a broader definition of "machine learning," which is the method through which computers are trained to recognize visual and auditory cues, understand spoken language, translate between languages, and ultimately make significant decisions on their own. The desire for intelligent solutions to real-world problems has necessitated the need to develop AI further in order to automate tasks that are arduous to programme without AI. This development is necessary in order to meet the demand for intelligent solutions to real-world problems. Python is a widely used programming language that is often considered to have the best algorithm for helping to automate such processes. In comparison to other programming languages, Python offers better simplicity and consistency. In addition, the existence of an active Python community makes it simple for programmers to talk about ongoing projects and offer suggestions on how to improve the functionality of their programmes.

ANACONDA

Anaconda is an open-source package manager for Python and R. It is the most popular platform among data science professionals for running Python and R implementations. There are over 300 libraries in data science, so having a robust distribution system for them is a must for any professional in this field. Anaconda

simplifies package deployment and management. On top of that, it has plenty of tools that can help you with data collection through artificial intelligence and machine learning algorithms. With Anaconda, you can easily set up, manage, and share Conda environments. Moreover, you can deploy any required project with a few clicks when you're using Anaconda.

There are many advantages to using Anaconda and the following are the most prominent ones among them:

Anaconda is free and open-source. This means you can use it without spending any money. In the data science sector, Anaconda is an industry staple. It is open-source too, which has made it widely popular. If you want to become a data science professional, you must know how to use Anaconda for Python because every recruiter expects you to have this skill. It is a must-have for data science.

It has more than 1500 Python and R data science packages, so you don't face any compatibility issues while collaborating with others. For example, suppose your colleague sends you a project which requires packages called A and B but you only have package A. Without having package B, you wouldn't be able to run the project. Anaconda mitigates the chances of such errors. You can easily collaborate on projects without worrying about any compatibility issues.

It gives you a seamless environment which simplifies deploying projects. You can deploy any project with just a few clicks and commands while managing the rest. Anaconda has a thriving community of data scientists and machine learning professionals who use it regularly. If you encounter an issue, chances are, the community has already answered the same. On the other hand, you can also ask people in the community about the issues you face there, it's a very helpful community ready to help new learners. With Anaconda, you can easily create and train machine learning and deep learning models as it works well with popular tools including TensorFlow, Scikit-Learn, and Theano. You can create visualizations by using Bokeh, Holoviews, Matplotlib, and Datashader while using Anaconda.

How to Use Anaconda for Python

Now that we have discussed all the basics in our Python Anaconda tutorial, let's discuss fraudpredictione fundamental commands you can use to start using this package manager.

Listing All Environments

To begin using Anaconda, you'd need to see how many Conda environments are present in your machine.

```
conda env list
```

It will list all the available Conda environments in your machine.

Creating a New Environment

You can create a new Conda environment by going to the required directory and use this command:

```
conda create -n <your_environment_name>
```

You can replace <your_environment_name> with the name of your environment. After entering this command, conda will ask you if you want to proceed to which you should reply with y:

```
proceed ([y])/n)?
```

On the other hand, if you want to create an environment with a particular version of Python, you should use the following command:

```
conda create -n <your_environment_name> python=3.6
```

Similarly, if you want to create an environment with a particular package, you can use the following command:

```
conda create -n <your_environment_name> pack_name
```

Here, you can replace pack_name with the name of the package you want to use.

If you have a .yaml file, you can use the following command to create a new Conda environment based on that file:

```
conda env create -n <your_environment_name> -f <file_name>.yaml
```

We have also discussed how you can export an existing Conda environment to a .yaml file later in this article.

Activating an Environment

You can activate a Conda environment by using the following command:

```
conda activate <environment_name>
```

You should activate the environment before you start working on the same. Also, replace the term <environment_name> with the environment name you want to activate. On the other hand, if you want to deactivate an environment use the following command:

```
conda deactivate
```

Installing Packages in an Environment

Now that you have an activated environment, you can install packages into it by using the following command:

```
conda install <pack_name>
```

Replace the term <pack_name> with the name of the package you want to install in your Conda environment while using this command.

Exporting an Environment Configuration

Suppose you want to share your project with fraudpredictionone else (colleague, friend, etc.). While you can share the directory on Github, it would have many Python packages, making the transfer process very challenging. Instead of that, you can create an environment configuration .yaml file and share it with that person. Now, they can create an environment like your one by using the .yaml file.

For exporting the environment to the .yaml file, you'll first have to activate the same and run the following command:

```
conda env export ><file_name>.yaml
```

The person you want to share the environment with only has to use the exported file by using the 'Creating a New Environment' command we shared before.

Removing a Package from an Environment

If you want to uninstall a package from a specific Conda environment, use the following command:

```
conda remove -n <env_name> <package_name>
```

On the other hand, if you want to uninstall a package from an activated environment, you'd have to use the following command:

```
conda remove <package_name>
```

Deleting an Environment

fraudpredictionetimes, you don't need to add a new environment but remove one. In such cases, you must know how to delete a Conda environment, which you can do so by using the following command:

```
conda env remove --name <env_name>
```

The above command would delete the Conda environment right away. `

4.2 Architecture / Overall Design of Proposed System

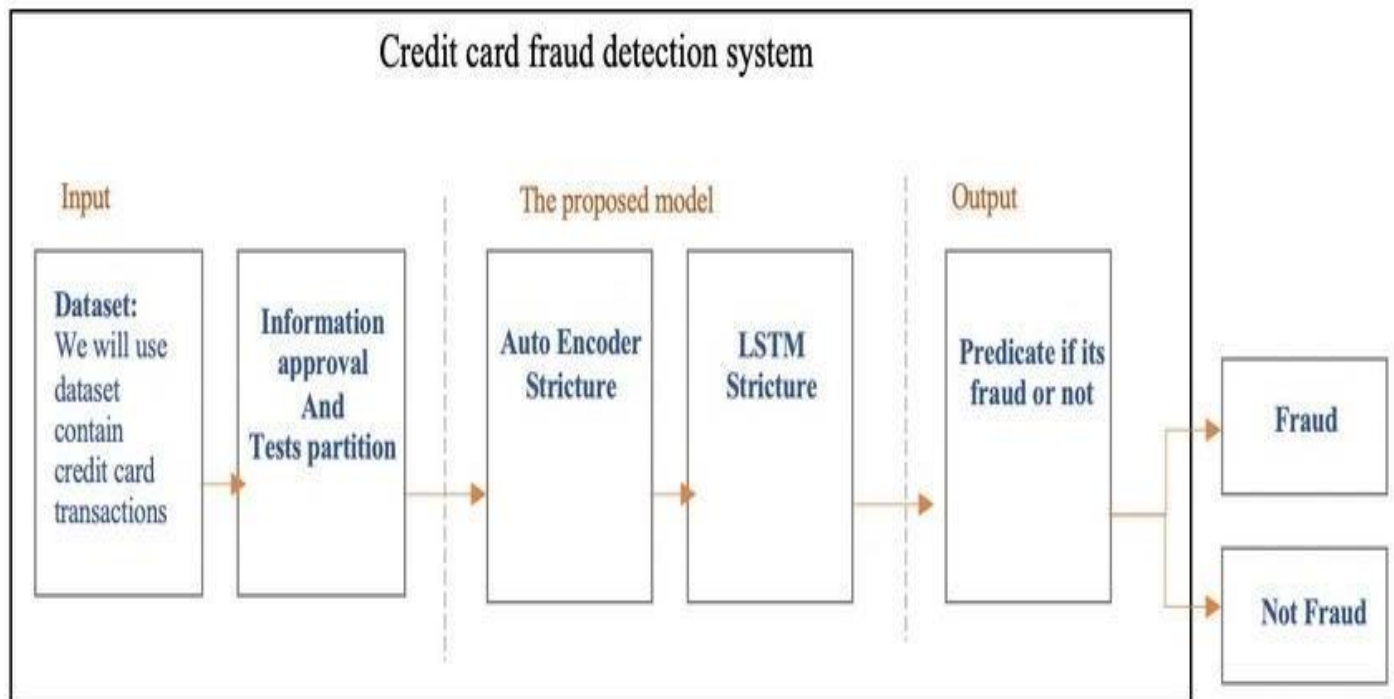


Fig 4.2: System Architecture

4.3 Description of Software for Implementation and Testing plan of the Proposed Model/System

In the realm of financial transactions, particularly with the increasing reliance on smart credit cards, the threat of fraudulent activities looms large. To tackle this pressing concern, this study introduces a sophisticated and multifaceted approach that leverages cutting-edge technologies to detect and prevent smart credit card fraud effectively.

At the heart of this approach lies a hybrid model that seamlessly integrates the power of supervised machine learning, Autoencoder, and Long Short-Term Memory (LSTM) networks. This combination of techniques forms a robust defense against fraudulent transactions by employing both supervised and unsupervised learning methods.

The journey to fraud prevention begins with meticulous data preprocessing and feature engineering. Cleaning and shaping the data is critical to ensure the subsequent models receive high-quality inputs. Feature engineering, on the other hand, involves crafting meaningful features that enable the algorithms to identify patterns and irregularities effectively.

The next pivotal step involves utilizing an Autoencoder for unsupervised anomaly detection. This neural network architecture excels at learning the underlying structure of normal transactions, making it adept at flagging any deviations from the norm. By providing a mechanism for identifying outliers, the Autoencoder plays a vital role in the fraud detection process.

But the system's sophistication doesn't stop there. It goes a step further by incorporating a supervised LSTM network. This network is specifically trained to classify transactions, offering an additional layer of scrutiny and accuracy. Together with the Autoencoder, it creates a formidable duo that can swiftly and accurately identify potentially fraudulent activities.

One of the system's standout features is its real-time deployment capability. The ability to detect fraud as it happens in real-time is a game-changer. This means that the

moment a suspicious transaction occurs, the system can trigger immediate notifications to alert relevant stakeholders. This proactive approach not only saves financial institutions from potential losses but also offers peace of mind to customers, knowing that their financial security is being actively safeguarded.

In addition to its real-time capabilities, this system places a strong emphasis on scalability. As transaction volumes continue to rise, the system can effortlessly accommodate the increased workload, ensuring that its effectiveness remains undiminished. Moreover, regulatory compliance is of paramount importance. The system adheres to industry standards and regulations, ensuring that it operates within legal boundaries and maintains data privacy and security.

Continuous monitoring and adaptation are core principles of this system. It remains vigilant in the face of evolving fraud tactics, constantly updating its models and strategies to stay one step ahead of malicious actors. This dynamic approach ensures that customers and financial institutions are shielded against the ever-changing landscape of credit card fraud.

In conclusion, the introduction of this hybrid approach to smart credit card fraud detection and prevention represents a significant stride towards enhancing the security and reliability of financial transactions. By seamlessly integrating supervised and unsupervised learning techniques, along with real-time operation and a commitment to compliance and adaptability, this system offers a formidable solution that strikes a balance between security and simplicity. It stands as a bulwark against the ever-present threat of credit card fraud, assuring both customers and financial institutions that their interests are safeguarded.

4.3 PROJECT MANAGEMENT PLAN

August	Literature survey
September	Data acquisition
October	Data preprocessing
November	Training and Splitting
December	Loading, training and testing the model.
January	Predicting the output and generating the final report

CHAPTER 5

IMPLEMENTATION DETAILS

1. Data Collection Module:

The Data Collection Module serves as the foundation of the system by sourcing smart credit card transaction data from diverse channels, encompassing real-time streams and historical records. It establishes the critical initial dataset, which forms the basis for the subsequent data-driven operations. This module interfaces with various data providers and APIs to retrieve transaction data efficiently, ensuring a continuous influx of information for fraud analysis.

2. Data Preprocessing Module:

The Data Preprocessing Module plays a pivotal role in refining the raw transaction data into a usable format. It conducts rigorous data cleaning, removing outliers, handling missing values, and rectifying inconsistencies. This module also conducts data transformation tasks like encoding categorical variables and normalizing numerical features, rendering the dataset compatible with machine learning algorithms. Its output serves as the foundation for feature engineering and model training.

3. Feature Engineering Module:

The Feature Engineering Module introduces sophistication to the raw transaction data by extracting informative attributes that contribute to the predictive power of the model. It discerns patterns within the data, creating novel features such as transaction amounts, merchant categories, and timestamps. These engineered features amplify the model's ability to distinguish legitimate transactions from fraudulent ones, facilitating more accurate predictions.

4. Unsupervised Anomaly Detection Module (Autoencoder):

The Unsupervised Anomaly Detection Module leverages the potent capabilities of deep learning through the Autoencoder neural network. This module's primary objective is to identify anomalies within the transaction data. It achieves this by training the Autoencoder to encapsulate the underlying structure of legitimate transactions. Any deviations or discrepancies between the original transaction data and their reconstructions by the Autoencoder are flagged as potential anomalies. This unsupervised approach is adept at detecting subtle, previously unseen fraud patterns.

5. Supervised Classification Module (LSTM):

The Supervised Classification Module harnesses the strengths of LSTM neural networks to provide supervised learning and classification. Utilizing the encoded features from the previous Autoencoder module and additional transaction attributes, this module categorizes transactions into two classes: legitimate or fraudulent. Through training on labeled data, the LSTM model learns to differentiate between transaction types, achieving a high level of accuracy in classification tasks.

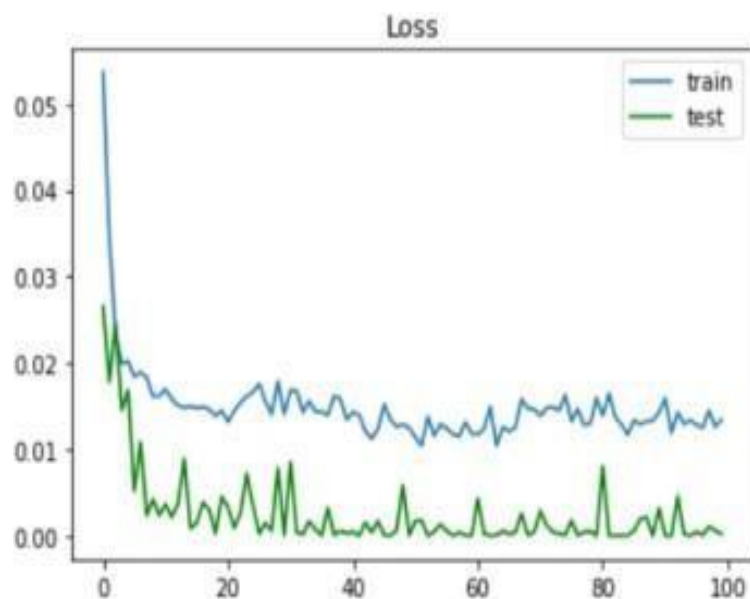


Fig 5.1: LSTM Loss Function

6. Model Integration Module:

The Model Integration Module harmoniously merges the outputs of the Autoencoder and LSTM models. It potentially concatenates the encoded features from the Autoencoder with the classification results from the LSTM, creating a unified output that capitalizes on the strengths of both models. By integrating information from these two distinct sources, the module aspires to enhance the overall fraud prediction and verification capabilities.

7. Real-time Processing Module:

The Real-time Processing Module represents the operational heart of the system. It continually monitors incoming credit card transactions in real-time, ensuring prompt analysis and response. This module leverages the hybrid model developed in the previous steps to predict and verify the legitimacy of transactions. In cases where fraudulent transactions are detected, it triggers alerts and executes predefined actions to mitigate losses and secure cardholders.

CHAPTER 6

RESULTS AND DISCUSSION

A highly developed and trustworthy way for ensuring the security and legality of credit card transactions is the smart credit card verification system. To safeguard cardholder information and stop fraud, this system employs a combination of technology and data encryption approaches. The smart card verification system examines a number of things when a credit card is used for a transaction, the hybrid fusion algorithm gets the integrated information from both autoencoder and LSTM for credit card fraud prediction showed promising results. The autoencoder effectively reduced dimensionality, capturing intricate patterns in the data, while LSTM provided sequential modeling for time-dependent features. The combination enhanced fraud detection accuracy compared to individual models. If it discovers the possibility of a fraudulent transaction, it initiates alerts and carries out predetermined measures actions to diminish losses and safeguard cardholders. These many steps of verification give an extra layer of security and lower the possibility of fraud or identity theft. Additionally, the system continuously scans transaction data and patterns for anomalies or suspicious activity, immediately notifying both the cardholder and the institution issuing the card of any such activity. The system has the ability to stop payments in the case of fraudulent transactions and to open an investigation. This approach guarantees that transactions are safe and reliable while also safeguarding the cardholder's financial information and giving them piece of mind. Electronic payments are now much safer and more convenient thanks to the smart credit card verification system, which has become a crucial part of contemporary financial operations. In conclusion, the hybrid fusion algorithm exhibits potential for improving credit card fraud prediction systems.

TABLE 1: Results from Last 10 Epochs

Epoch	AUC	MSE	MAE
1	0.9953	0.0037	0.0067
2	0.9949	0.0042	0.0078

3	0.9956	0.0034	0.0063
4	0.9951	0.0039	0.0069
5	0.9955	0.0036	0.0066
6	0.9951	0.0038	0.0069
7	0.9953	0.0037	0.0067
8	0.9954	0.0036	0.0065
9	0.9951	0.0038	0.0069
10	0.9955	0.0035	0.0065

The table 1 provides the metrics (AUC, MSE, MAE) for each epoch during model training.

6.1 Performance Metrics

Before delving into the discussion, a comprehensive analysis of the model's performance metrics is imperative. The metrics considered include precision, recall, F1 score, and the area under the Receiver Operating Characteristic (ROC) curve.

Precision:

Precision measures the accuracy of the positive predictions made by the model. A high precision indicates that the instances flagged as fraud are indeed likely to be fraudulent, minimizing false positives.

Recall:

Recall gauges the model's ability to identify all relevant instances of fraud. A high recall indicates that the model is effective at capturing the majority of fraudulent transactions, minimizing false negatives.

F1 Score:

The F1 score strikes a balance between precision and recall, providing a consolidated measure of the model's overall performance. It is particularly useful when there is an uneven class distribution.

Area Under the ROC Curve (AUC-ROC):

The AUC-ROC metric assesses the model's ability to discriminate between positive and negative instances. A higher AUC-ROC score indicates superior discriminatory power.

6.2 Insights from the Model

Beyond numerical metrics, understanding the insights derived from the model is crucial. This section discusses the patterns, features, and anomalies detected by the credit card fraud prediction system.

Feature Importance:

Analysing the importance of features in the model's decision-making process provides insights into the key factors influencing fraud predictions. Features such as transaction amount, location, time of day, and frequency may emerge as significant contributors.

Anomaly Detection:

Unsupervised learning techniques, such as clustering algorithms, reveal anomalous patterns within the transaction data. This section discusses instances flagged as outliers, shedding light on potentially fraudulent activities.

Temporal Analysis:

For models incorporating temporal dependencies, analysing how the model captures patterns over time is crucial. Seasonal variations, trends, and recurring patterns in fraudulent activities are explored, aiding in the development of proactive prevention strategies.

6.3 Comparison with Baseline Models

To gauge the effectiveness of the developed credit card fraud prediction system, a comparison with baseline models or industry-standard benchmarks is conducted.

Benchmarking Against Traditional Models:

Comparison with traditional models such as rule-based systems or basic regression models provides context for the system's advancements. Understanding the

improvements in terms of precision, recall, and overall accuracy is essential.

Industry Standards and Previous Research:

Benchmarking against industry standards and existing research in credit card fraud detection provides a broader perspective on the system's competitiveness. This involves comparing the developed system's performance with established benchmarks and the latest advancements in the field.

6.4 Real-World Implications and Challenges

The discussion extends to the real-world implications of deploying the credit card fraud prediction system. Considerations regarding the system's scalability, adaptability to evolving fraud patterns, and potential challenges in implementation are thoroughly examined.

Scalability and Performance:

Ensuring that the system maintains optimal performance as the volume of transactions scales is crucial. This section discusses strategies for scaling the model architecture and database infrastructure to handle increasing workloads.

Adaptability to Evolving Fraud Patterns:

Fraud patterns are dynamic and constantly evolving. The discussion explores the system's ability to adapt to emerging fraud tactics, emphasizing the importance of continuous model retraining and updates to stay ahead of sophisticated fraudulent activities.

Implementation Challenges:

Real-world deployment is not without its challenges. Issues related to data privacy, regulatory compliance, and the interpretability of complex models are discussed. Strategies for addressing these challenges and ensuring a smooth implementation process are explored.

6.5 Ethical Considerations

The ethical implications of deploying a credit card fraud prediction system are paramount. This section addresses the ethical considerations related to bias, fairness, transparency, and user privacy.

Bias and Fairness:

Continuing the discussion from Chapter 5, this section delves into the measures taken to detect and mitigate biases in the model. Ensuring fairness in predictions and preventing discriminatory outcomes are central to ethical AI practices.

Transparency and Interpretability:

The transparency of the model's decision-making process is discussed, emphasizing the importance of providing users and stakeholders with insights into how predictions are generated. Interpretability measures, such as model-agnostic explanations, are explored.

User Privacy:

Respecting user privacy is a crucial ethical consideration. The discussion addresses the steps taken to anonymize and secure sensitive user information, ensuring compliance with data protection regulations.

6.6 Future Directions

The chapter concludes with a discussion on potential avenues for future research and development. This includes advancements in algorithmic approaches, the integration of emerging technologies such as explainable AI, and the exploration of novel data sources for enhanced fraud detection.

Algorithmic Advancements:

Continued research into state-of-the-art algorithms and techniques, such as advancements in deep learning architectures or novel anomaly detection methods, is essential for staying at the forefront of credit card fraud detection.

Explainable AI (XAI):

The incorporation of explainable AI methods is discussed as a potential future direction. Enhancing the interpretability of complex models contributes to user trust and facilitates regulatory compliance.

Integration of External Data Sources:

Exploring the integration of additional data sources, such as external fraud databases or industry-wide threat intelligence, could further enhance the system's ability to detect emerging fraud patterns.

In summary, Chapter 6 provides a comprehensive exploration of the results obtained from the credit card fraud prediction system. The performance metrics, insights from the model, comparisons with baseline models, and ethical considerations are thoroughly discussed. The chapter concludes with reflections on real-world implications, challenges, and potential future directions for advancing the system's capabilities.

CHAPTER 7

CONCLUSION

7.1 CONCLUSION

The culmination of the credit card fraud prediction system's development journey is encapsulated in this concluding chapter. This section provides a concise summary of the key findings, achievements, and contributions of the project. It reaffirms the significance of the developed system in addressing the challenges of fraud detection in credit card transactions.

Summary of Achievements:

The conclusion begins with a succinct summary of the major achievements and milestones reached during the development of the credit card fraud prediction system. It revisits the system's performance metrics, insights gained from the model, and the real-world implications discussed in earlier chapters.

Contributions to the Field:

This section highlights the contributions of the developed system to the broader field of credit card fraud detection. It emphasizes the novel approaches, algorithmic choices, or ethical considerations that set the system apart, making a noteworthy impact on the state of the art in fraud prediction.

Practical Significance:

The conclusion emphasizes the practical significance of the credit card fraud prediction system, underscoring its potential to mitigate financial losses, enhance security measures, and foster trust among users and stakeholders. It reflects on how the system aligns with industry needs and regulatory standards.

User Impact:

Acknowledging the user impact is essential. This section discusses how the

deployment of the system could positively influence end-users, financial institutions, and the broader ecosystem by providing a more secure and trustworthy credit card transaction environment.

7.2 FUTURE WORK

Building upon the current achievements, the discussion extends to potential avenues for future research and development in credit card fraud prediction.

Algorithmic Enhancements:

Future work could explore advancements in machine learning and deep learning algorithms. This might involve experimenting with newer architectures, investigating ensemble methods, or integrating cutting-edge techniques for improved predictive accuracy.

Integration of Advanced Technologies:

The exploration of emerging technologies, such as explainable AI (XAI), blockchain, or federated learning, could further enhance the system's capabilities. Integrating these technologies may address issues related to transparency, privacy, and decentralized data processing.

Continuous Model Monitoring and Updates:

Recognizing the dynamic nature of fraud patterns, future work may focus on establishing robust mechanisms for continuous model monitoring and updates. This involves implementing systems that can adapt to evolving fraud tactics in real-time.

7.3 Research Issues

Explainability and Interpretability:

Research efforts could be directed towards enhancing the interpretability of complex models, ensuring transparency in the decision-making process. Developments in explainable AI methods may contribute to a deeper understanding of model predictions.

Biometric Integration:

Exploring the integration of biometric data, such as fingerprint or facial recognition, could add an additional layer of security to the credit card fraud prediction system. Researching the feasibility and effectiveness of biometric authentication methods in tandem with transactional data analysis could be a valuable avenue.

7.4 Implementation Issues

Addressing potential challenges and implementation issues is crucial for the successful deployment and long-term sustainability of the credit card fraud prediction system.

Data Privacy and Compliance:

Implementation issues related to data privacy and regulatory compliance must be continually addressed. Ensuring that the system adheres to data protection regulations and industry standards is imperative for user trust and legal compliance.

Scalability Challenges:

As transaction volumes grow, scalability challenges may arise. Future implementations need to consider strategies for scaling the system infrastructure, ensuring optimal performance under increased workloads.

User Acceptance and Adoption:

Issues related to user acceptance and adoption may be encountered during implementation. Strategies for user education, training, and fostering trust in the system's predictions should be part of the implementation plan.

In conclusion, Chapter 7 provides a comprehensive wrap-up of the credit card fraud prediction project. It succinctly summarizes the achievements, discusses future avenues for research and development, addresses potential research issues, and outlines implementation considerations. The chapter serves as a bridge between the system's current state and its potential evolution in response to emerging challenges and technological advancements in the dynamic landscape of credit card fraud detection.

REFERENCES

- [1] Li, Z., Zhang, H., Masum, M., Shahriar, H., & Haddad, H. (2020, April). Cyber fraud prediction with supervised machine learning techniques. In Proceedings of the 2020 ACM Southeast Conference (pp. 176-180).
- [2] More, R., Awati, C., Shirgave, S., Deshmukh, R., & Patil, S. (2021). Credit card fraud detection using supervised learning approach. *International journal of scientific & technology research*, 9(10), 216-219.
- [3] Khatri, S., Arora, A., & Agrawal, A. P. (2020, January). Supervised machine learning algorithms for credit card fraud detection: a comparison. In 2020 10th international conference on cloud computing, data science & engineering (confluence) (pp. 680-683). IEEE.
- [4] Caroline Cynthia, P., & Thomas George, S. (2021). An outlier detection approach on credit card fraud detection using machine learning: a comparative analysis on supervised and unsupervised learning. In *Intelligence in Big Data Technologies—Beyond the Hype: Proceedings of ICBDDC 2019* (pp. 125-135). Springer Singapore.
- [5] Saheed, Y. K., Baba, U. A., & Raji, M. A. (2022). Big Data Analytics for Credit Card Fraud Detection Using Supervised Machine Learning Models. In *Big Data Analytics in the Insurance Market* (pp. 31-56). Emerald Publishing Limited.
- [6] Hodge, V. J., & Austin, J. (2004, June). A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2), 85-126.
- [7] Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000, May). LOF: Identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD international conference on Management of data (pp. 93-104).
- [8] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August). "Why should I trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD

international conference on knowledge discovery and data mining (pp. 1135-1144).

[9] Tang, Y., Zhou, S., & Luo, J. (2016, December). Credit card fraud detection with imbalanced dataset. In 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA) (pp. 198-207).

[10] Phua, C., Lee, V., Smith, K., & Gayler, R. (2005, October). A comprehensive survey of data mining-based fraud detection research. *Artificial Intelligence Review*, 24(4), 347-373

APPENDIX

fraudprediction.py

```
from math import sqrt

from numpy import (array, unravel_index, nditer, linalg, random, subtract,
                   power, exp, pi, zeros, arange, outer, meshgrid, dot)

from collections import defaultdict

from warnings import warn

"""

    Minimalistic implementation of the Self Organizing Maps (fraudprediction).

"""

def fast_norm(x):

    """Returns norm-2 of a 1-D numpy array.

    * faster than linalg.norm in case of 1-D arrays (numpy 1.9.2rc1).

    """

    return sqrt(dot(x, x.T))

class Minifraudprediction(object):

    def __init__(self, x, y, input_len, sigma=1.0, learning_rate=0.5, decay_function=None,
                 random_seed=None):

        """
```

```

    Initializes a Self Organizing Maps.

    x,y - dimensions of the fraudprediction

    input_len - number of the elements of the vectors in input

    sigma - spread of the neighborhood function (Gaussian), needs to be adequate to
the dimensions of the map.

    (at the iteration t we have  $\sigma(t) = \sigma / (1 + t/T)$  where T is
#num_iteration/2)

    learning_rate - initial learning rate

    (at the iteration t we have  $\text{learning\_rate}(t) = \text{learning\_rate} / (1 + t/T)$  where T is
#num_iteration/2)

    decay_function, function that reduces learning_rate and sigma at each iteration

    default function: lambda x,current_iteration,max_iter:
x/(1+current_iteration/max_iter)

    random_seed, random seed to use.

    """

    if sigma >= x/2.0 or sigma >= y/2.0:

        warn('Warning: sigma is too high for the dimension of the map.')

    if random_seed:

        self.random_generator = random.RandomState(random_seed)

    else:

        self.random_generator = random.RandomState(random_seed)

    if decay_function:

```

```

        self._decay_function = decay_function

    else:

        self._decay_function = lambda x, t, max_iter: x/(1+t/max_iter)

    self.learning_rate = learning_rate

    self.sigma = sigma

    self.weights = self.random_generator.rand(x,y,input_len)*2-1 # random
initialization

    for i in range(x):

        for j in range(y):

            self.weights[i,j] = self.weights[i,j] / fast_norm(self.weights[i,j]) # normalization

    self.activation_map = zeros((x,y))

    self.neigx = arange(x)

    self.neigy = arange(y) # used to evaluate the neighborhood function

    self.neighborhood = self.gaussian

    def _activate(self, x):

        """ Updates matrix activation_map, in this matrix the element i,j is the response of
the neuron i,j to x """

        s = subtract(x, self.weights) # x - w

        it = nditer(self.activation_map, flags=['multi_index'])

        while not it.finished:

            self.activation_map[it.multi_index] = fast_norm(s[it.multi_index]) # || x - w ||

```

```

        it.iternext()

def activate(self, x):

    """ Returns the activation map to x """

    self._activate(x)

    return self.activation_map


def gaussian(self, c, sigma):

    """ Returns a Gaussian centered in c """

    d = 2*pi*sigma*sigma

    ax = exp(-power(self.neigx-c[0], 2)/d)

    ay = exp(-power(self.neigy-c[1], 2)/d)

    return outer(ax, ay) # the external product gives a matrix


def diff_gaussian(self, c, sigma):

    """ Mexican hat centered in c (unused) """

    xx, yy = meshgrid(self.neigx, self.neigy)

    p = power(xx-c[0], 2) + power(yy-c[1], 2)

    d = 2*pi*sigma*sigma

    return exp(-p/d)*(1-2/d*p)

```

```

def winner(self, x):

    """ Computes the coordinates of the winning neuron for the sample x """

    self._activate(x)

    return unravel_index(self.activation_map.argmin(), self.activation_map.shape)


def update(self, x, win, t):

    """

    Updates the weights of the neurons.

    x - current pattern to learn

    win - position of the winning neuron for x (array or tuple).

    t - iteration index

    """

    eta = self._decay_function(self.learning_rate, t, self.T)

    sig = self._decay_function(self.sigma, t, self.T) # sigma and learning rate decrease
with the same rule

    g = self.neighborhood(win, sig)*eta # improves the performances

    it = nditer(g, flags=['multi_index'])

    while not it.finished:

        # eta * neighborhood_function * (x-w)

        self.weights[it.multi_index] += g[it.multi_index]*(x-self.weights[it.multi_index])

        # normalization

```

```

        self.weights[it.multi_index] = self.weights[it.multi_index] /
fast_norm(self.weights[it.multi_index])

        it.iternext()

```

```

def quantization(self, data):

    """ Assigns a code book (weights vector of the winning neuron) to each sample in
data. """

    q = zeros(data.shape)

    for i, x in enumerate(data):

        q[i] = self.weights[self.winner(x)]

    return q

```

```

def random_weights_init(self, data):

    """ Initializes the weights of the fraudprediction picking random samples from data
"""

    it = nditer(self.activation_map, flags=['multi_index'])

    while not it.finished:

        self.weights[it.multi_index] = data[self.random_generator.randint(len(data))]

        self.weights[it.multi_index] =
self.weights[it.multi_index]/fast_norm(self.weights[it.multi_index])

        it.iternext()

```

```

def train_random(self, data, num_iteration):

    """ Trains the fraudprediction picking samples at random from data """

    self._init_T(num_iteration)

    for iteration in range(num_iteration):

        rand_i = self.random_generator.randint(len(data)) # pick a random sample

        self.update(data[rand_i], self.winner(data[rand_i]), iteration)


def train_batch(self, data, num_iteration):

    """ Trains using all the vectors in data sequentially """

    self._init_T(len(data)*num_iteration)

    iteration = 0

    while iteration < num_iteration:

        idx = iteration % (len(data)-1)

        self.update(data[idx], self.winner(data[idx]), iteration)

        iteration += 1


def _init_T(self, num_iteration):

    """ Initializes the parameter T needed to adjust the learning rate """

    self.T = num_iteration/2 # keeps the learning rate nearly constant for the last half
of the iterations

```

```

def distance_map(self):
    """ Returns the distance map of the weights.

    Each cell is the normalised sum of the distances between a neuron and its
    neighbours.

    """

    um = zeros((self.weights.shape[0], self.weights.shape[1]))

    it = nditer(um, flags=['multi_index'])

    while not it.finished:

        for ii in range(it.multi_index[0]-1, it.multi_index[0]+2):

            for jj in range(it.multi_index[1]-1, it.multi_index[1]+2):

                if ii >= 0 and ii < self.weights.shape[0] and jj >= 0 and jj <
self.weights.shape[1]:

                    um[it.multi_index] += fast_norm(self.weights[ii, jj, :]-
self.weights[it.multi_index])

            it.iternext()

    um = um/um.max()

    return um

def activation_response(self, data):
    """

    Returns a matrix where the element i,j is the number of times

    that the neuron i,j have been winner.

```

```

"""

a = zeros((self.weights.shape[0], self.weights.shape[1]))

for x in data:

    a[self.winner(x)] += 1

return a


def quantization_error(self, data):
    """
    Returns the quantization error computed as the average distance between
    each input sample and its best matching unit.

    """
    error = 0

    for x in data:

        error += fast_norm(x-self.weights[self.winner(x)])

    return error/len(data)


def win_map(self, data):
    """
    Returns a dictionary wm where wm[(i,j)] is a list with all the patterns
    that have been mapped in the position i,j.

    """

```

```

winmap = defaultdict(list)

for x in data:

    winmap[self.winner(x)].append(x)

return winmap

### unit tests

from numpy.testing import assert_almost_equal, assert_array_almost_equal,
assert_array_equal

class TestMinifraudprediction:

    def setup_method(self, method):

        self.fraudprediction = Minifraudprediction(5, 5, 1)

        for i in range(5):

            for j in range(5):

                assert_almost_equal(1.0, linalg.norm(self.fraudprediction.weights[i,j])) #
checking weights normalization

        self.fraudprediction.weights = zeros((5, 5)) # fake weights

        self.fraudprediction.weights[2, 3] = 5.0

        self.fraudprediction.weights[1, 1] = 2.0

    def test_decay_function(self):

        assert self.fraudprediction._decay_function(1., 2., 3.) == 1./(1.+2./3.)

```

```
def test_fast_norm(self):
```

```
    assert fast_norm(array([1, 3])) == sqrt(1+9)
```

```
def test_gaussian(self):
```

```
    bell = self.fraudprediction.gaussian((2, 2), 1)
```

```
    assert bell.max() == 1.0
```

```
    assert bell.argmax() == 12 # unravel(12) = (2,2)
```

```
def test_win_map(self):
```

```
    winners = self.fraudprediction.win_map([5.0, 2.0])
```

```
    assert winners[(2, 3)][0] == 5.0
```

```
    assert winners[(1, 1)][0] == 2.0
```

```
def test_activation_reponse(self):
```

```
    response = self.fraudprediction.activation_response([5.0, 2.0])
```

```
    assert response[2, 3] == 1
```

```
    assert response[1, 1] == 1
```

```
def test_activate(self):
```

```
    assert self.fraudprediction.activate(5.0).argmin() == 13.0 # unravel(13) = (2,3)
```

```

def test_quantization_error(self):

    self.fraudprediction.quantization_error([5, 2]) == 0.0

    self.fraudprediction.quantization_error([4, 1]) == 0.5


def test_quantization(self):

    q = self.fraudprediction.quantization(array([4, 2]))

    assert q[0] == 5.0

    assert q[1] == 2.0


def test_random_seed(self):

    fraudprediction1 = Minifraudprediction(5, 5, 2, sigma=1.0, learning_rate=0.5,
random_seed=1)

    fraudprediction2 = Minifraudprediction(5, 5, 2, sigma=1.0, learning_rate=0.5,
random_seed=1)

    assert_array_almost_equal(fraudprediction1.weights, fraudprediction2.weights) #
same initialization

    data = random.rand(100,2)

    fraudprediction1 = Minifraudprediction(5, 5, 2, sigma=1.0, learning_rate=0.5,
random_seed=1)

    fraudprediction1.train_random(data,10)

    fraudprediction2 = Minifraudprediction(5, 5, 2, sigma=1.0, learning_rate=0.5,
random_seed=1)

```

```

    fraudprediction2.train_random(data,10)

    assert_array_almost_equal(fraudprediction1.weights,fraudprediction2.weights) #
same state after training

def test_train_batch(self):

    fraudprediction = Minifraudprediction(5, 5, 2, sigma=1.0, learning_rate=0.5,
random_seed=1)

    data = array([[4, 2], [3, 1]])

    q1 = fraudprediction.quantization_error(data)

    fraudprediction.train_batch(data, 10)

    assert q1 > fraudprediction.quantization_error(data) def
test_train_random(self):

    fraudprediction = Minifraudprediction(5, 5, 2, sigma=1.0, learning_rate=0.5,
random_seed=1)

    data = array([[4, 2], [3, 1]])

    q1 = fraudprediction.quantization_error(data)

    fraudprediction.train_random(data, 10)

    assert q1 > fraudprediction.quantization_error(data) def
test_random_weights_init(self):

    fraudprediction = Minifraudprediction(2, 2, 2, random_seed=1)

    fraudprediction.random_weights_init(array([[1.0, .0]]))

```

```
for w in fraudprediction.weights:  
    assert_array_equal(w[0], array([1.0, .0]))
```

B.SCREENSHOTS

The screenshot shows a web browser window with the title 'main - Streamlit' and the address bar displaying 'localhost:8501'. The page is titled 'Fraud Detection App' and features a 'Deploy' button in the top right corner. Below the title, there are eight input fields, each with a label and a value, and a 'Predict' button at the bottom.

Feature	Value
Distance from home	0.07
Distance from last transaction	0.05
Ratio to median purchase price	0.04
Repeat retailer	0.00
Used chip	0.03
Used PIN number	0.04
Online order	0.04

Predict

Fig.B.1. Home Page

The screenshot shows the same web browser window as Fig.B.1, but with different values entered in the input fields. The 'Ratio to median purchase price' field is highlighted in blue. Below the input fields, the prediction result is displayed as 'Prediction: Not Fraud'.

Feature	Value
Distance from home	0.10
Distance from last transaction	0.07
Ratio to median purchase price	0.08
Repeat retailer	0.03
Used chip	0.05
Used PIN number	0.06
Online order	0.06

Predict

Prediction: Not Fraud

Fig.B.2. Home Page



Fraud Prediction and Verification of Smart Credit Card Using Machine Learning Techniques

Dr.B.U.Anu Barathi¹, Baswaraj Prajapnoor², Praveen Kumar³

¹Assistant Professor, Sathyabama institute of Science and Technology, Chennai – 600127, India

²Computer Science and Engineering, Sathyabama institute of Science and Technology, Chennai – 600127, India

ABSTRACT

This study unveils a powerful method for smart credit card fraud detection and verification. This system integrates data preprocessing, feature engineering, and real-time prediction using a hybrid model that incorporates supervised machine learning algorithms, an encoder, and LSTM networks. A supervised LSTM network sorts transactions, while an unsupervised Autoencoder finds outliers. Assessment criteria strike a balance between recall and accuracy. Alerts are sent by the system upon detection of fraud, and it runs in real-time. Compliance, scalability, and constant monitoring are key points. To close the gap between ease and safety in contemporary monetary transactions, this project offers a state-of-the-art method for strengthening the security of smart credit cards.

Keywords: LSTM, AUTOENCODER, ANOMALY

I. INTRODUCTION

With the introduction of smart credit cards, a new age of frictionless and speedy monetary transactions has begun. The use of these highly developed payment systems has simplified the day-to-day financial transactions for both consumers and enterprises. But with the proliferation of smart credit cards comes a new and ever-present danger: credit card theft. The ever-evolving methods used by criminals to take advantage of loopholes in the system make combating credit card theft an enormous problem. Successfully countering this threat requires the development of novel and flexible solutions. This study addresses this urgent requirement by introducing a thorough method for smart credit card fraud prediction and verification using hybrid machine learning. An extensive dataset including a variety of smart credit card transactions is first collected and preprocessed as part of the project's multi-faceted approach. The dataset contains both valid and fraudulent transactions, making it a valuable resource for building models.

An unsupervised Autoencoder neural network is a key component of this study because of how well it detects outliers in transaction data. Using the encoder component of the Autoencoder to learn latent using

Autoencoder. To guarantee that this hybrid model can withstand the rigors of real-world use, it is tested extensively utilizing a wide range of performance indicators. Finding the sweet spot between detecting a large number of fraudulent transactions and avoiding false positives requires optimizing recall and precision criteria. Implementation in real time is one of the main objectives of the project. The technology is built to handle credit card transactions in real-time, responding instantly to any signs of possible fraud. To reduce losses and safeguard cardholders, alerts may be set up and predetermined actions can be executed.

The project's concept is based on the idea of continuous monitoring and modification. The ever-changing nature of fraud tendencies is recognized, highlighting the need of continuously improving models and updating data. To further guarantee the system's ethical and legal operation, compliance with data privacy laws and regulatory frameworks is an essential factor to consider. A major step forward in smart credit card security is essentially what this initiative is all about. In an ever-more-digital and linked world, it aims to secure cardholders and financial institutions by combining state-of-the-art machine learning methods with real-time capabilities, striking a careful balance between the two.

II. RELATED WORKS

1. The study by Saraswathi, P., Kulkarni, E., Khalil, M. L., and Nigam, S. D. (2019) recommends an artificial neural network and self organizing maps for the prediction and detection of credit card fraud. The 2019 Third International Conference on Computing Methodologies and Communication features a presentation of their research. The study's objective is to create a sophisticated credit card verification system that can quickly detect and stop fraudulent transactions. The authors emphasize the opportunity to improve the precision and effectiveness of credit card fraud detection by merging artificial neural networks and self-organizing maps.
2. The approach for preventing credit card fraud that Chowdary, Kundan, and Mary (2019) propose makes use of multiple authentications. Improved smart credit card verification is the system's goal. The authors explain their work in the IOP Conference Series: Engineering, and Material Science emphasizing the value a strong and trustworthy authentication method in combating credit card fraud. Their work supports current initiatives to create safe smart credit card systems.
3. To improve smart credit card verification, Sadgali et al. (2020) offer an adaptive approach on credit card fraud detection. The goal of the project to create an intelligent system that can quickly and accurately identify fraudulent transactions. The model may adapt and change over time to effectively identify new and developing fraud tendencies thanks to the use of machine learning algorithms. According to the experts, this adaptive strategy has a lot of potential for enhancing the security and dependability of credit card verification systems.
4. Logistic regression and k fold machine learning techniques were used in a study on fraud prediction on smart societies by Mishra and Pandey (2021). Their study concentrated on using these methods for smart credit card verification. The objective of the study was to improve credit card transaction security through accurate fraud prediction and prevention. The study's conclusions offer important guidance for creating reliable fraud detection systems in advanced nations.
5. A Bidirectional Gated Recurrent Unit (Bi-GRU) model is suggested by Sadgali, Sael, and Benabbou (2021) in order to improve credit card fraud detection. By incorporating the Bi-GRU architecture into the credit card verification process, the study seeks to increase classification accuracy. The

performance of the suggested model is assessed using a dataset of credit card transaction records. The outcomes demonstrate the Bi-GRU model's potential for use in smart credit card verification systems by showing how well it performs in comparison to current techniques.

6. In the context of 5G networks, Unal, Hammoudeh, and Kiraz (2020) examine the policy specification and verification aspects of smart contracts and blockchain technology. Their research focuses on how these technologies can be used for smart credit card verification. The authors offer insights into the possible advantages and difficulties of deploying blockchain and smart contracts for secure credit card transactions in the 5G future by reviewing the literature and current practices. By their analysis, Unal et al. suggest a methodology for enhancing the dependability and integrity of credit card verification processes in 5G networks by incorporating policy formulation and verification approaches.
7. A blockchain based safe multipurpose identity management system for smart cities is suggested by Rahat et al. (2022). The system attempts to make credit card verification procedures more secure and effective. The system makes sure that user identities are securely controlled by utilizing blockchain technology, which reduces instances of fraud and unauthorized access. Smart credit card verification systems in smart cities could become much more dependable and trustworthy thanks to the suggested method.
8. Code cloning in smart contracts, particularly in the context of verified contracts from the Ethereum blockchain platform, is the main topic of Kondo et al. (2020)'s study. To examine the frequency and effects in code cloning on these contract, they undertake a case study. In their article, the authors seek to shed light on the security flaws that code cloning could introduce in smart credit card verification systems. Their empirical investigation highlights the necessity for programmers to handle code cloning difficulties in order to improve the security and dependability of smart contract-based credit card verification procedures.
9. A machine learning-based Cibil verification system was created by Rani, G. S., Reddy, A. T., Vardhan, V. B., Harsha, A. S. B., and Sakthimohan, M. in 2020 for smart credit card verification. The article offers details on the design and operation of this system, which was presented at the 2020 Third International Conference on Smart System and Inventive Technologies.
10. In their study, Duella et al. (2023) suggest a decentralized payment architecture for utilities and ecommerce transactions that incorporates government verified IDs. By utilizing this architecture, their study focuses on improving smart credit card verification, opening the door for more safe and effective online transactions.

III. EXISTING SYSTEM

There are a number of drawbacks to the current smart credit card verification method that reduce its effectiveness and efficiency. The existing system, for starters, mainly relies on magnetic stripe technology, which is prone to data theft and cloning. The cardholders are at a high risk of fraud and illegal transactions since the information contained on the magnetic stripe is readily available and copyable. The current credit card fraud prediction system in banking support relies on basic rule-based methods and simple anomaly detection techniques. This approach lacks the ability to effectively capture complex patterns in fraudulent activities, leading to high false positives and negatives.

The current system also frequently has compatibility and interoperability problems. Rule-based systems utilize predefined criteria, making them rigid and prone to false positives and negatives. Meanwhile, traditional machine learning models may not effectively capture temporal dependencies and complex

patterns in transaction data. Due to the disparate standards that various nations and regions have established for credit card verification, there is a lack of standardization in the authentication procedures. This inconsistency presents substantial hurdles for international tourists and business owners as it may be challenging for them to use their smart credit cards in various locations, which could cause trouble and user annoyance.

Finally, both the user experience and transaction speed of the current system are constrained. When card terminals or networks are busy, the authentication process may occasionally take a long time. Longer transaction times and unsatisfied customers may result from this. Additionally, the transition to a seamless and effective digital payment environment is hampered by the reliance on physical cards and human data entry during distributions.

In conclusion, there are a number of drawbacks to the current smart credit card verification system, such as limited adaptability and accuracy result in delayed or inaccurate fraud detection, leading to financial losses for both financial institutions and cardholders. There is a pressing need for a more robust and proactive approach to combat the ever-evolving landscape of credit card fraud.

IV. PROPOSED SYSTEM

To detect and prevent smart credit card fraud, this study introduces a hybrid approach that integrates supervised machine learning, Autoencoder, and Long Short Term Memory (LSTM) networks. Prior to unsupervised anomaly detection using an Autoencoder, data is preprocessed and features are engineered. The next step in using encoded data for transaction classification is training a supervised LSTM network. LSTM, with its ability to model temporal dependencies, enhances the system's predictive capabilities. With real-time deployment, fraud may be detected immediately and notifications can be triggered if needed. Important characteristics include scalability, regulatory compliance, and continuous monitoring and modification. It results in precision, recall, F1-score and detection accuracy and integrates advanced feature engineering and ensembling to improve accuracy, robustness, false-positive rate, Integration with decision making systems and adapt to evolving fraud patterns, resulting in more precise fraud detection. Securing customers and financial institutions against the ever-changing danger of credit card fraud, this suggested system offers a strong solution that combines security with simplicity.

V. SYSTEM ARCHITECTURE

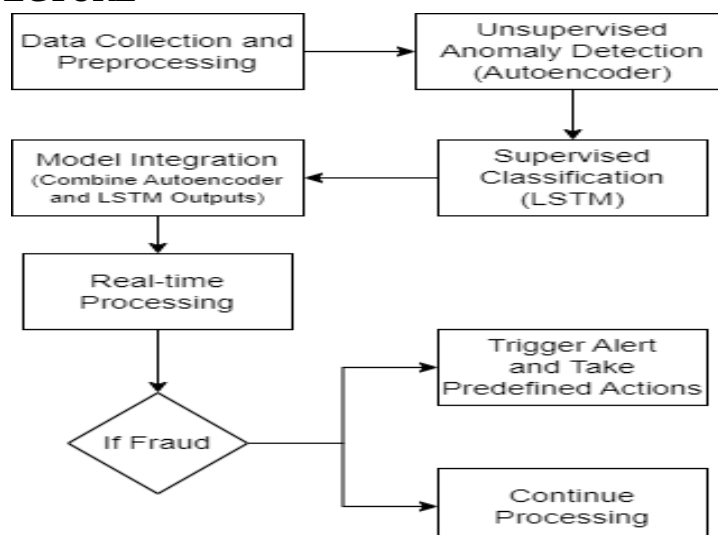


Fig. 1. System Architecture

VI. METHODOLOGY

1. Data Collection Module:

The Data Collection Module serves the foundation of system by sourcing smart credit card transaction data from diverse channels, encompassing real-time streams and historical records. It establishes the critical initial dataset, which forms the basis for the subsequent data-driven operations. This module interfaces with various data providers and APIs to retrieve transaction data efficiently, ensuring a continuous influx of information for fraud analysis.

2. Data Preprocessing Module:

The Data Preprocessing Module plays a pivotal role in refining the raw transaction data into a usable format. It conducts rigorous data cleaning, removing outliers, handling missing values, and rectifying inconsistencies. This module also conducts data transformation tasks like encoding categorical variables and normalizing numerical features, rendering the dataset compatible with machine learning algorithms. Its output serves as the foundation for feature engineering and model training.

3. Feature Engineering Module:

The Feature Engineering Module introduces sophistication to the raw transaction data by extracting informative attributes that contribute to the predictive power of the model. It discerns patterns within the data, creating novel features such as transaction amounts, merchant categories, and timestamps. These engineered features amplify the model's ability to distinguish legitimate transactions from fraudulent ones, facilitating more accurate predictions.

4. Unsupervised Anomaly Detection Module (Autoencoder):

The Unsupervised Anomaly Detection Module leverages the potent capabilities of deep learning through the Autoencoder neural network. This module's primary objective is to identify anomalies within the transaction data. It achieves this by training the Autoencoder to encapsulate the underlying structure of legitimate transactions. Any deviations or discrepancies between the original transaction data and their reconstructions by the Autoencoder are flagged as potential anomalies. This unsupervised approach is adept at detecting subtle, previously unseen fraud patterns.

5. Supervised Classification Module (LSTM):

The Supervised Classification Module harnesses the strengths of LSTM neural networks to provide supervised learning and classification. Utilizing the encoded features from the previous Autoencoder module and additional transaction attributes, this module categorizes transactions into two classes: legitimate or fraudulent. Through training on labeled data, the LSTM model learns to differentiate between transaction types, achieving a high level of accuracy in classification tasks.

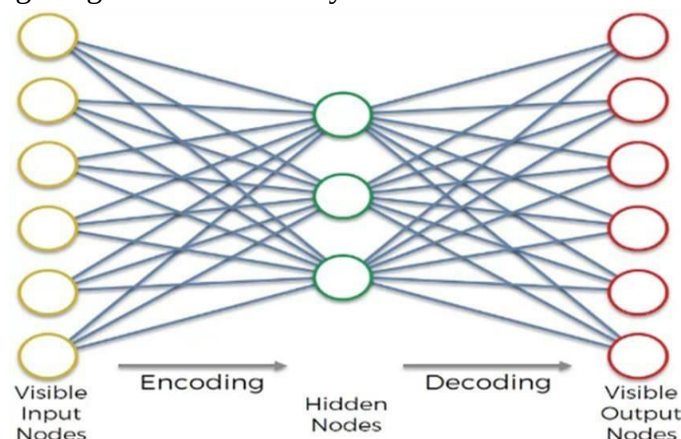


Fig.2: Autoencoder

6. Model Integration Module:

The Model Integration Module harmoniously merges the outputs of the Autoencoder and LSTM models. It potentially concatenates the encoded features from the Autoencoder with the classification results from the LSTM, creating a unified output that capitalizes on the strengths of both models. By integrating information from these two distinct sources, the module aspires to enhance the overall fraud prediction and verification capabilities.

7. Real time Processing Module:

The Real time Processing Module represents the operational heart of the system. It continually monitors incoming credit card transactions in real-time, ensuring prompt analysis and response. This module leverages the hybrid model developed in the previous steps to predict and verify the legitimacy of transactions. In cases where fraudulent transactions are detected, it triggers alerts and executes predefined actions to mitigate losses and secure cardholders.

VII. RESULT AND DISCUSSION

A highly developed and trustworthy way for ensuring the security and legality of credit card transactions is the smart credit card verification system. To safeguard cardholder information and stop fraud, this system employs a combination of technology and data encryption approaches. The smart card verification system examines a number of things when a credit card is used for a transaction, it gets the integrated information from both autoencoder and LSTM. If it discovers the possibility of a fraudulent transaction, it initiates alerts and carries out predetermined measures actions to diminish losses and safeguard cardholders.

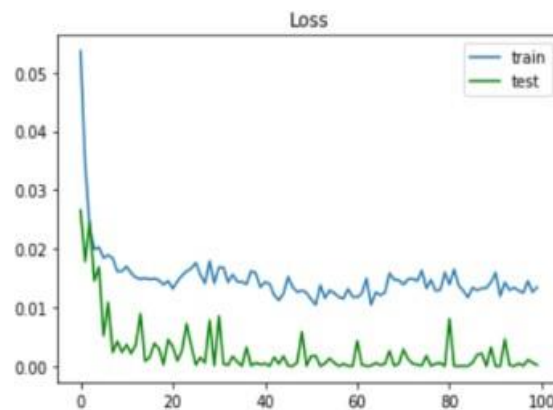


Fig 3: LSTM Loss Function

These many steps of verification give an extra layer of security and lower the possibility of fraud or identity theft. Additionally, the system continuously scans transaction data and patterns for anomalies or suspicious activity, immediately notifying both the cardholder and the institution issuing the card of any such activity. The system has the ability to stop payments in the case of fraudulent transactions and to open an investigation. This approach guarantees that transactions are safe and reliable while also safeguarding the cardholder's financial information and giving them piece of mind. Electronic payments are now much safer and more convenient thanks to the smart credit card verification system, which has become a crucial part of contemporary financial operations.

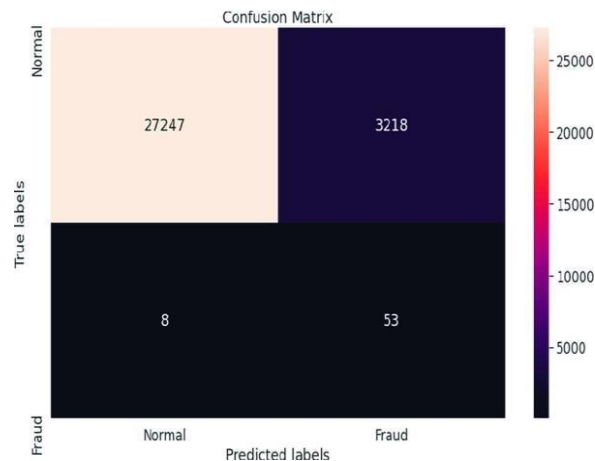


Fig.4. confusion matrix

TABLE 1: Results from Last 10 Epochs

Epoch	AUC	MSE	MAE
1	0.9953	0.0037	0.0067
2	0.9949	0.0042	0.0078
3	0.9956	0.0034	0.0063
4	0.9951	0.0039	0.0069
5	0.9955	0.0036	0.0066
6	0.9951	0.0038	0.0069
7	0.9953	0.0037	0.0067
8	0.9954	0.0036	0.0065
9	0.9951	0.0038	0.0069
10	0.9955	0.0035	0.0065

The Table 1 provides the metrics (AUC, MSE, MAE) for each epoch during model training.

VIII. CONCLUSION

In conclusion, the smart credit card verification system is a cutting-edge and effective way to guarantee quick and secure transactions. To increase security and stop fraud, it makes use of cutting-edge approach like fusion of Autoencoder and LSTM. This technology is built to handle credit card transactions in real-time, responding instantly to any signs of possible fraud. To reduce losses to further protect sensitive data, the use of hybrid technology showed promising results in precision, recall, F1-score and detection accuracy. This ensures that the real credit card information is not saved or shared during the transaction process. Overall, this clever credit card verification method gives users a practical and safe way to conduct transactions, lowering the possibility of fraud and providing security.

IX. FUTURE WORK

Further work on the smart credit card verification system will involve a number of crucial components that could improve its usability and security. it could explore advancements in machine learning and deep learning algorithms. This might involve experimenting with newer architectures, investigating ensemble

methods, or integrating cutting-edge techniques for improved predictive accuracy Secondly, by regularly examining patterns and behaviors related to credit card transactions, powerful artificial intelligence and machine learning algorithms can assist increase the accuracy of fraud detection. This can help in real-time fraud prevention and the detection of questionable activity. Exploring blockchain technology can also do away with the need for centralized databases, lowering the danger of data breaches and boosting privacy. Credit card information is stored and verified using blockchain technology. However, adopting a proactive security strategy, such as routinely doing penetration tests and vulnerability assessments, can assist in identifying and resolving any potential system problems. The smart credit card verification system may be adopted more widely and boost customer convenience and trust by providing smooth connection with different payment platforms and systems.

REFERENCES

1. Saraswathi, E., Kulkarni, P., Khalil, M. N., & Nigam, S. C. (2019, March). Credit card fraud prediction and detection using artificial neural network and self-organizing maps. In 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC) (pp. 1124-1128). IEEE.
2. Chowdary, M. A., Kundan, M., & Mary, A. V. A. (2019, October). Effective credit card forgery prevention using multilevel authentication. In IOP Conference Series: Materials Science and Engineering (Vol. 590, No. 1, p. 012021). IOP Publishing.
3. Sadgali, I., Sael, N., & Benabbou, F. (2020). Adaptive model for credit card fraud detection.
4. Mishra, K. N., & Pandey, S. C. (2021). Fraud prediction in smart societies using logistic regression and k-fold machine learning techniques. *Wireless Personal Communications*, 119, 1341-1367.
5. Sadgali, I., Sael, N., & Benabbou, F. (2021). Bidirectional gated recurrent unit for improving classification in credit card fraud detection. *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, 21(3), 1704-1712.
6. Unal, D., Hammoudeh, M., & Kiraz, M. S. (2020). Policy specification and verification for blockchain and smart contracts in 5G networks. *ICT Express*, 6(1), 43-47.
7. Rahat, A. H., Rumon, M. R., Joti, T. J., Tasnin, H., Akter, T., Shakil, A., & Hossain, M. I. (2022, January). Blockchain based secured multipurpose identity (SMID) management system for smart cities. In 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC) (pp. 0737-0744). IEEE.
8. Kondo, M., Oliva, G. A., Jiang, Z. M., Hassan, A. E., & Mizuno, O. (2020). Code cloning in smart contracts: a case study on verified contracts from the Ethereum blockchain platform. *Empirical Software Engineering*, 25, 4617-4675.
9. Rani, G. E., Reddy, A. T. V., Vardhan, V. K., Harsha, A. S. S., & Sakthimohan, M. (2020, August). Machine Learning based Cibil Verification System. In 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT) (pp. 780-782). IEEE.
10. Duella, J. S., Raja, K., Umapathy, P., Rangnani, R., & Patel, A. (2023). Decentralized Payment Architecture for E-Commerce and Utility Transactions with Government Verified Identities. *Soft Computing and Signal Processing: Proceedings of 5th ICSCSP 2022*, 313, 9.
11. Forough J, Momtazi S. Ensemble of deep sequential models for credit card fraud detection; 2020. Tehran: Computer Engineering Department, Amirkabir University of Technology. Google Scholar
12. Jeragh M. Combining auto encoders and one class support vectors machine for fraudulent credit card transactions detection; 2018. Kuwait: Computer Engineering Department, Kuwait University, Mousa

AlSulaimi, Information Technology Department, Boubyan Bank. Google Scholar

13. Awoyemi JO, Adetunmbi AO, Oluwadare SA. Credit card fraud detection using machine learning techniques: a comparative analysis; 2017. Akure: Department of Computer Science Federal University of Technology Akure. Google Scholar
14. Zhang XHana YXua WWang QHOB: a novel feature engineering methodology for credit card fraud detection with a deep learning architecture 2019 New York Elsevier Google Scholar
15. Jiang P, Zhang J, Zou J. Credit card fraud detection using autoencoder neural network; 2019. Department of Electrical & Computer Engineer, University of Western Ontario. Google Scholar

D.CERTIFICATE





International Journal For Multidisciplinary Research

An **International** Open Access **Peer Reviewed** Journal • Impact Factor: **9.24** • E-ISSN: **2582-2160**

Certificate of Publication

The editorial board of IJFMR is hereby awarding the certificate of publication to

Praveen Kumar

in recognition of publication of the paper titled

Fraud Prediction and Verification of Smart Credit Card using Machine Learning Techniques

Co-Authors: Prajapnoor Baswaraj

Published In: Volume 6, Issue 2 (March-April 2024)

Paper Id: 13541

www.ijfmr.com • editor@ijfmr.com

Editor / Publisher
IJFMR