Financial Fraud Detection System Using Machine Learning and Deep Learning

A Project report

Submitted by:

Aditya Narayan Sahu (2141019163)

Satyabrat Sahoo (2141019117)

Soumya Ranjan Patra (2141016213)

Achyuta Samantaray (2141019198)

In partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Faculty of Engineering and Technology, Institute of Technical Education and Research

SIKSHA 'O' ANUSANDHAN (DEEMED TO BE) UNIVERSITY,
Bhubaneswar, Odisha, India
(June 2025)

CERTIFICATE

This is to certify that the project report titled **FINANCIAL FRAUD DETECTION SYSTEM USING MACHINE LEARNING AND DEEP LEARNING** is being submitted by **Aditya Narayan Sahu, Satyabrat Sahoo, Soumya Ranjan Patra and Achyuta Samantaray** of Section 'V' to the Institute of Technical Education and Research, Siksha 'O' Anusandhan (Deemed to be) University, Bhubaneswar for the partial fulfilment for the degree of *Bachelor of Technology* in *Computer Science & Engineering,* is a record of original confide work carried out by them under my supervision and guidance. The project work, in my opinion, has reached the requisite standard fulfilling the requirements for the degree of Bachelor of Technology. The results contained in this project work have not been submitted in part or full to any other University or Institute for the award of any degree or diploma.

Ms. Anuradha Mohanta
Department of Computer Science & Engineering

Faculty of Engineering and Technology;
Institute of Technical Education and Research;
Siksha 'O' Anusandhan (Deemed to be)
University

ACKNOWLEDGMENT

It is a matter of great pleasure for us to get this opportunity to express our sincere sense of gratitude to Siksha ‘O’ Anusandhan Deemed to be University. Firstly, we would like to express our hearty thanks to the Institute of Technical Education and Research for providing lab facilities and other relevant facilities. Our supervisor Ms. Anuradha Mohanta was the main force behind all these efforts. Because of his valuable suggestions and proper guidance for this project.

We express our sincere thanks to the Computer Science Engineering department HOD, Dr. Debahuti Mishra, for allowing me/us to use the facilities of the institute. We are also thankful to all those who have helped us in this endeavor, either directly or indirectly, especially all our teachers. At last, we would like to express a big thank you to all our friends and all known & unknown people who have helped us directly or indirectly.

Place: ITER, SOA, Bhubaneswar

Date: **16.06.2025**

Signature of Students

DECLARATION

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

Signature of Students with Registration Numbers

Date: 16.06.2025

REPORT APPROVAL

This project report entitled **FINANCIAL FRAUD DETECTION SYSTEM USING MACHINE LEARNING AND DEEP LEARNING** by **Aditya Narayan Sahu, Satyabrat Sahoo, Soumya Ranjan Patra and Achyuta Samantaray** is approved for the degree of *Bachelor of Technology* in *Computer Science & Engineering*.

Examiner(s)

____

____

____

Supervisor

____

Project Coordinator

____

PREFACE

Financial transactions are mostly conducted over the internet today, businesses are more vulnerable to fraud than before. Financial fraud causes both a lot of money to be lost and leeches trust from digital users. The use of old rules can't keep up with the changes in how fraudsters operate. To deal with this matter, our project will work on a smart system that can identify fraudulent financial actions accurately.

The system has the ability to review transaction data and foresee whether the payment is valid or a scam. The research was based on real data that included more than 6.3 million records of financial transactions. Some of the biggest difficulties in this field come from the class imbalance, since there are hardly any fraudulent transactions compared to the other types. Therefore, we used SMOTE to create fake examples of the minority class in our data which made the dataset more balanced and improved the detection of fraud.

The preprocessing phase involved Standard-Scaler to scale down the data, division of the data for training and testing with stratification and checking features' relationships. The main models we evaluated and implemented included Random Forest, SGD Classifier, RNN and LSTM using machine learning and deep learning approaches. All performance results were compared by evaluating accuracy, precision, recall, F1-score and ROC AUC. Out of all the models, **LSTM with SMOTE** did best by producing the most balanced and successful outcomes, mainly in terms of **accuracy** and **recall**.

In summary, intelligent models prove they can spot unauthorized transactions happening in large businesses. Besides helping build reliable financial systems, this project leaves a way for adding features, including APIs for on-the-fly threat detection, explanations based on SHAP or LIME and deployment via web dashboards.

INDIVIDUAL CONTRIBUTIONS

Aditya Narayan Sahu	Literature survey, Dataset finding, SGD model training, testing and Results evaluation
Satyabrat Sahoo	Data visualization, LSTM model training, test results and evaluation
Soumya Ranjan Patra	Random forest model training and testing, result evaluation and Documentation
Achyuta Samantaray	Preprocessing, RNN model training and testing, Result Evaluation and Documentation

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Nowadays, many financial transactions happen online which adds a risk of fraud for the companies involved. It leads to a big loss in money and also makes people less sure about digital platforms. The old way of using rules alone can no longer work because fraud can happen in new and different ways. To overcome this issue, our project, called “Financial Fraud Detection System Using Machine Learning and Deep Learning”, will build a system that identifies fraudulent transactions very accurately and reliably.

The system checks data from transactions to predict if they are fraudulent or legitimate. We investigated a data sample made up of more than 6.3 million financial transaction records. In this area, a big challenge is that there are very few fraudulent transactions compared to the rest of the data. To solve this, we used SMOTE (Synthetic Minority Oversampling Technique) which increased the number of minority class cases in the data and helped our model spot fraud more effectively.

In this project, I did some initial work by scaling features with Standard-Scaler, conducting a stratified train-test split and doing a data analysis to learn about the relationships between different features. We set up and checked a range of machine learning and deep learning models, mainly targeting Random Forest, SGD Classifier, RNN and LSTM models. The performance was measured using accuracy, precision, recall, F1-score and ROC AUC. LSTM with SMOTE gave the best and most balanced results in recall and F1-score which plays an important role in fraud detection systems.

The last system reveals that models based on artificial intelligence can discover irregularities in large financial transactions. It supports the security of financial systems and at the same time enables possible enhancements such as connecting to live APIs for real-time detection, using SHAP or LIME to understand the model’s predictions and deploying it using web-based tools.

These days, people depend on the internet for regular financial activities like shopping, bank transactions, paying bills and transferring funds outside the country. With how easy it is to do online transactions; more cases of financial fraud have surfaced that impact all kinds of users. Such activities negatively affect the funds of organizations and also hurt the trust and reputation of financial platforms. Fraud has grown in complexity and happens more frequently than most detectors can catch using only static and human rules. Therefore, we need technologies that can quickly adapt to new types of fraud. That is exactly the problem our project plans to solve.

Financial Fraud Detection System is a smart program that tracks, examines and anticipates fraud in online financial activities with the help of Machine Learning (ML) and Deep Learning (DL). With the help of real-time data and machine learning, the system looks for unusual activity, including unauthorized entry, abnormal transactions or actions not like a person’s typical behavior, so it can react fast.

For many years, financial organizations have had trouble spotting both genuine people and avoid spotting fraud. We plan to reduce the gap by bringing in a data-centered solution. Generally, old fraud detection systems are unable to manage the increasing level of detail in today’s fraudulent activities. This is why many companies now rely on machine learning for checking suspicious transactions: it looks for unknown trends, adjusts itself to evolving techniques and becomes more accurate over the years.

To prepare for development, we investigated various fraud analytics studies and case documents, even studying how companies like Visa and PayPal are now using AI for fraud detection. Most of the ideas came from actual datasets and scientific studies on handling unequal data in fraud detection and time-series analysis, along with anomaly detection. Because of these insights, our solution was built in this way.

We use machine learning algorithms like Random Forest, SGD and Deep Learning algorithms like RNN and LSTM in providing our solutions. Because of this such models can detect fraud that takes place bit by bit in transactions. We also applied SMOTE which addressed the large difference between the amounts of valid and fraudulent data, helping our machines understand both types of behavior.

On a higher level, our project seeks to offer a dependable, correct and scalable fraud detection system that doesn't disturb genuine users even while detecting fraud. Since digital payments are on the rise everywhere, systems like ours have become vital to guarantee the safety of consumers and the banking sector.

1.1 Problem Overview

Online shopping and digital banking have made today's financial transactions much more effortless and faster. As a result, this digital change has resulted in a surge in financial crimes such as stealing credit cards, sending out phishing emails, taking over people's identity and accessing their accounts unlawfully. Despite the rules, human checking and manual steps, traditional fraud detection is not strong enough to fight today's complex financial fraud crimes. This means that many fraud cases go unnoticed for a while which leads to major financial damage and a drop in trust.

We developed Financial Fraud Detection System which uses ML and DL to automatically spot unlawful financial activities from a large amount of transaction information. The system is set up to both catch fraud with high accuracy and adapt by studying the examples in the records. While regular systems handle fraud only after it happens, our solution seeks to spot and stop it as it happens or almost immediately.

The most important part of the system relies on data analysis that looks at amounts, how often transactions happen and other behaviors of customers. It uses several classification techniques like Random Forest and SGD Classifier and advanced deep models like RNN and LSTM which have proven to be very useful for catching trends and issues over time. We tackled the problem of rare fraud cases by incorporating SMOTE. Therefore, the model learns to identify fraud equally with non-fraud cases which improves its ability to spot suspicious activity.

As well as detecting issues, the system keeps improving by giving performance feedback in real time. Flagged transactions are always captured, reviewed and the model is retrained using them from time to time. As a result, the system grows to meet new types of fraud and remains efficient for an extended period. The system can integrate with different financial systems since it follows a modular architecture, helping it fit real-world situations.

Rather than being a simple code-based system, our fraud detection tool makes sure we keep everything transparent as well as secure and easy for people to use. Users get notified when a suspicious event takes place, so they can swiftly act by either confirming or rejecting the action. Those working in finance and technology can use visual dashboards to analyze fraud patterns, check confusion matrices and monitor how effective the models are.

Besides, the system safeguards and anonymizes data by encrypting it while handling and learning from it. As a result, the company is compliant with financial data protection and doubters gain confidence in the system.

Overall, the Financial Fraud Detection System is not only a detection system; it is also an intelligent design that safeguards digital transactions, limits the risks to performance and earns users' trust in these financial platforms. By using the latest AI, instant processing and thinking about users first, the system helps make the digital economy much safer.

1.2 Motivation

Concerns about financial fraud online are what led to the development of the Financial Fraud Detection System in today's rapidly changing, technology-based world. Those of us who are students and young professionals do most of our transactions online, through digital money tools. With time, we realized that there were greater numbers of phishing, shady warnings, unexplained losses from cards and news of people getting scammed and losing money. Such events made us understand that our finances can be unsafe since fraudsters are improving their methods.

This encouraged us to develop a system based on new machine learning and deep learning techniques to catch and stop financial fraud as soon as it happens. Standard approaches usually miss out on fraud before it occurs and, sadly, may also wrongly block good users as they allow fraud to happen. We felt there had to be a new approach which is why our project began.

Since we are regular users of online banking and digital payments, we understood that it was important to have a fraud detection system capable of adjusting to new forms of fraud. We hoped for a way that could match new trends, learn from information and notify people or businesses ahead of criminal activity. To make digital finance easier and safer as apps make life easier these days, we wanted to create a system for it.

We were particularly interested in real-time fraud detection with RNN and LSTM which have the ability to identify patterns in the sequence of transactions that could be missed using conventional tools. Most of the time, fraud is formed by slow changes in behavior. Our purpose was to look for these clever elements in financial data by using deep learning.

We also noticed that fraud detection datasets were not well balanced, which influenced motivation. As very few transactions are fraudulent, most systems do not learn how to detect it. So, we chose to use SMOTE which made it possible for us to use a balanced dataset without sacrificing the data’s quality. Implementing this model helped it improve, so it could catch more fraud cases.

Also, we wanted the solution to be compatible with banking app Moving forward, this system should be able to give accurate results, be easy for anyone to use and be understandable. Since users keep an eye on their online orders, we think they should also be able to monitor the security of their money with confidence.

We also hope that our project helps cut down digital financial crime, shields normal users and strengthens the cybersecurity of financial institutions. We think the impact of the system will spread to many others, including students or single users, banks, fintech companies and digital wallet providers.

To sum up, we wanted to construct a fraud detection system that is reliable, smart and scalable so it constantly learns and puts the user first. Given that internet crimes and fraud are rising due to the digital trend, it is important to address them now. That’s why our aim is to ensure safety and intelligence in future buildings.

1.3 Uniqueness of the Work

Suspicious activity in digital financial areas can be detected instantly, thanks to the intelligent machine learning (ML) and deep learning (DL) integration in the Financial Fraud Detection System. Because fixed rules cannot keep up with always-changing fraud schemes, our system is able to improve and become stronger as time goes by.

The use of Long Short-Term Memory (LSTM) networks in our project makes it possible for the model to identify fraudulent activity by looking for patterns as time goes on. Usually, fraudulent behavior is the result of a series of small actions that are hard to spot. Using data series from previous transactions, LSTM manages to notice hidden patterns of fraud that most other models overlook. Because of this, our system can predict events which is a major improvement.

SMOTE is another impressive feature that helps solve the problem of class imbalance in the data. In the real world, very few transactions in financial data are fraudulent. Implementing SMOTE lets both types of classes be understood evenly, which greatly improves the recall and F1-score, making it less likely for true frauds to slip through.

In addition, our system is able to catch fraud as it happens thanks to live monitoring of transactions. The design ensures that the architecture is easy to use, complies with APIs and can scale without limits, so it fits perfectly into apps, wallets and financial dashboards. Depending on the predictions, institutions may get automatic alerts, ask users to confirm transactions or put deals on hold for some time.

Apart from ensuring the project is strong, we also focus on its ease of use and openness. With our approach, we have tools such as confusion matrix plots, feature ranking and classification report charts. Because of these elements, financial analysts can better understand what the models do and how they work. It is not common for fraud detection systems to be so open, making everyone, including developers and users, feel safer.

The modular structure of the system ensures that predictions are saved and used to make future models perform better. With more fraud tactics emerging, the model becomes smarter and works better during each transaction. Besides, RNNs make it possible for the system to notice and respond to changes and timings in multiple transactions, increasing its ability to identify threats.

The system's unique qualities in development come from the way it balances speed, accuracy, ability to scale and flexibility. Unlike many other fraud detection systems, ours can be quickly put into place since it is largely powerful and practical to implement. It operates in different areas of finance and can be adjusted according to the company's own rules or laws it needs to follow.

Simply put, this project stands out from other fraud tools because it is highly intelligent, capable of adjusting and ready to face future dangers posed to digital finances. The combination of deep learning, real-time prediction and feedback turns the process for detecting and preventing financial fraud into something unique in our age.

1.4 Report Layout

This report is structured systematically to provide a comprehensive overview of the design, implementation and evaluation of our project titled "Financial Fraud Detection System Using Machine Learning and Deep Learning." Each chapter is organized to reflect the progression of the project from conceptualization to execution and analysis.

Chapter 1: Introduction

This chapter provides an overview of the project, including the motivation behind developing an intelligent fraud detection system, the uniqueness of our approach and a summary of the report structure. It explains the problem being addressed, the real-world impact of financial fraud and our project's aim to enhance fraud detection through machine learning and deep learning.

Chapter 2: Literature Survey

This section reviews existing work in the field of financial fraud detection. It highlights the limitations of traditional rule-based systems, the role of machine learning in automating fraud detection and recent advancements using deep learning models. It also discusses relevant research papers that influenced our system design.

Chapter 3: Materials and methods

This chapter explains the functional and non-functional requirements of the system. It includes the system architecture, data flow diagrams and detailed design components. The technical stack used for building the project is also presented here. It explains how we collected, preprocessed and balanced the data using SMOTE. It also outlines the machine learning and deep learning models used, particularly Random Forest, SGD Classifier, RNN and LSTM, along with training and testing procedures.

Chapter 4: Results/Outputs

This section presents the results of the various models used in the system. It compares their performance based on metrics such as accuracy, precision, recall, F1-score and ROC-AUC. Visual outputs like confusion matrices and performance graphs are included to support the evaluation.

Chapter 5: Conclusion

This chapter summarizes the key findings of the project and highlights the success of using deep learning, particularly LSTM, in detecting fraud effectively. It also proposes future improvements such as real-time deployment, model explainability and integration with live transaction platforms.

Chapter 6: References

The report concludes with a list of all research papers, datasets and tools referenced during the project.

Chapter 7: Reflection of Team Members on the Project

This chapter summarizes the impact of project and how this project allowed our team to strengthen collaboration skills and extend our knowledge regarding machine learning and fraud detection. We got to know how to effectively split up the work, how to find solutions to tricky problems, and how to combine various models. The project helped us not only in boosting our technical knowledge and teamwork but also made us ready to face the real world.

Chapter 8: Similarity Report

This structured layout ensures clarity, flow and thorough documentation of each phase of the project, allowing readers to understand the problem, approach, implementation and results effectively.

CHAPTER 2

LITERATURE SURVEY

A review of the literature discloses that learning models, mainly LSTM and RNN, do a good job at identifying temporal factors involved in financial fraud.

Some relevant literature surveys are summarized below to highlight the effectiveness of learning models in detecting financial fraud:

Md Al-Imran & Eftekhar Hossain Ayon suggested a combination of SMOTE, LSTM, and XGBoost as a fraud detecting system. Their work established that LSTM networks are effective in capturing temporal transaction patterns, and they are accurate in detecting fraud. Class imbalance was solved using the integration of SMOTE, and feature importance analysis was enhanced with the help of XGBoost.

Cost-sensitive learning with Random Forest and SVM was proposed by **A. Dal Pozzolo et al.** to address the problem of imbalanced datasets. Their new strategy resulted in a greater cost of misclassification on fraud cases which gave a low false negative rate. Such a methodology was effective in detecting credit card frauds in real life situations.

Rojan Zaki Abdulkreem and **Adnan Mohsin Abdulazeez** studied state-of-the-art RNN and LSTM models on financial fraud detection. In their paper, the prominent advantage of deep learning over conventional approaches was emphasized to be in capturing non-linear patterns and sequential dependencies in transaction data.

D. O. Nijoku et al. suggested the logistic regression in fraud detection in financial institutions as a web-based application. They combined rule-based check with ML and paid attention to user interactivity and backend performance. It was identified to be less complex than DL models, yet accorded interpretability to regulatory standards.

E. Ngai et al. have surveyed ML/data mining methods of fraud detection. They identified supervised, unsupervised, and hybrid methods and gave a structure to guide the appropriateness of models depending on the nature of the dataset and the kind of fraud.

Ibissam Benchaji et al. proposed sequence classifier based on LSTM and this is used to model the individual cardholder behavior. Their model achieved high recall by looking at sequential transactions and showing that LSTMs are better than static classifiers on credit card fraud.

Aravind Kumar Kalusivalingam et al. put forward a hybrid of DL and Random Forest model. The combination used the feature selection capability of Random Forest and pattern recognition of deep learning and demonstrated resilience to changing fraud strategies.

Kianeh Kandi Antonio Garcia-Dopico compared LSTM and XGBoost with imbalanced datset. LSTMs had a higher accuracy, and XGBoost had a low precision. SMOTE proved essential to the balancing of training data, but LSTMs were intrinsically more tolerant of skews.

Yisong Chen and others systematically analyzed the annual developments in DL on fraud detection. They studied CNNs, transformers, and ensemble approaches and discovered that hybrid architecture (e.g., LSTM + attention) held the most promise.

I Oluwabusayo Adijat Bello & Adebola Folorunso put forward the inclusion of ML/AI in financial cybersecurity. Their model highlighted dynamic models of real time fraud prevention, which filled the loopholes of the legacy systems in responsiveness.

Ibissam Benchaji et al. implemented the attention mechanism on top of LSTM, which increased the interpretability and accuracy of the models. Attention weights were useful to understand suspicious transaction segments, filling the black box gap of deep learning.

Eswar Prasad Galla & Hemanth Kumar Gollangi have tested ANNs, SVM and Decision Trees in fraud detection. ANNs had the best recall, whereas SVMs provides better precision-recall trade-offs on imbalanced data.

E.V.T. Ngai et al. connected fraud detection (FFD) and prevention (FFP) with the use of data mining. Their taxonomy system puts more emphasis on proactive actions (e.g., anomaly detection) in addition to reactive FFD systems and urged the use of end-to-end solutions.

2.1 Existing Systems

With the help of ML and DL, detecting financial fraud has experienced significant changes. Researchers found that LSTM models are most popular due to their capability to capture consecutive transactions and thus are useful in detecting both complicated and new types of fraud. These achievements allowed us to develop a system that is much better than previous LSTM models. The model's performance is better than the others as it achieves a strong increase in accuracy and almost perfect AUC, indicating that it is able to tell fraudulent transactions apart from legitimate ones very well. The model strongly stands out because of its high recall, preventing missing fraud that could cause big losses in financial fraud cases. However, being more accurate means, it is somewhat less accurate which is typically all right in cases when finding more fraud is the prime concern and slightly more missed cases are allowed.

Our proposal, which uses LSTM, can dynamically analyze patterns in transactions, whereas Logistic Regression and SVM deal only with fixed input data, making it less suitable for fighting new kinds of fraud. While other popular models fix class imbalance mostly using SMOTE, our system manages to provide strong recall with little oversampling. This development also allows it to respond to new situations in real time which helps it spot fraudulent activities right away.

Current systems, that is rule-based engines and ML methods such as Random Forest and SVM, encounter some drawbacks. These rule-based systems do not change much, resulting in too many false positives. They also cannot deal with new types of fraud. ML models, on the other hand, need a lot of handwork and are not designed to spot temporal connections among features. Also, certain powerful DL models like Transformers need a lot of computing resources which makes it hard for them to be used in places with few resources.

To sum up, our approach makes use of LSTM and increases recall so that it can help reduce fraud that slips past. Future researchers will concentrate on bridging LSTM and attention mechanisms in models to increase the balance between precision and recall. Besides, we are looking to use explainable modules to increase transparency and address the growing requirements set by financial sector regulators.

Results of Previous LSTM model and our proposed model is given below in Table 1:

Table 1. Comparison with Existing system

Models	PERFORMANCE METRICS				
	Accuracy	**Precision**	**Recall**	**F1-score**	**AUC**
LSTM [1]	0.9850	0.8720	0.8470	0.8590	0.9445
Proposed LSTM Model	**0.9969**	**0.2862**	**0.9725**	**0.4422**	**0.9983**

2.2 Problem identification

Every year, many banks, payment systems and e-commerce platforms lose billions of dollars because of financial fraud. While significant improvements in ML and DL have enhanced how well detection works, the current systems are not fully effective due to serious shortcomings. An important challenge is that fraudulent records in these datasets usually add up to less than 1%. Because most models focus more on the majority classes and less on the minorities, they often issue too many false negative results—which can be quite harmful in finance. Most of the time, Random Forest and Logistic Regression models do not adjust well to unbalanced data. It is also a major problem that existing systems do not catch new kinds of fraud. New fraud tactics are being used all the time, so usual machine learning engines often can't block them. In addition, some existing models focus on being accurate, but this results in missing many fraudulent transactions and this is easily the most hazardous risk that financial institutions face.

Often, efforts to commit fraud can be noticed as fraudsters conduct some small samples first and then do large, threatening transactions. LSTMs are generally less accurate over a large number of time steps and SVMs as well as XGBoost do not address dependencies between successive inputs. Additionally, today, it is especially important to catch fraud in real time, particularly for instant payment services. It takes a long time for batch-processing and when too many layers are in some DL models, they make applications too slow.

According to the literature, even well-known LSTM models could not do well, missing around 15% of fraud cases and having a moderate AUC of 94.45%, demonstrating that improvement is needed. We focus on four important areas to handle these issues in our solution. To begin with, it uses a combination of SMOTE, under sampling and cost-sensitive learning to adjust for the imbalance in different classes. It also uses special architecture, with attention mechanisms included, to boost the ability to detect long-range dependencies in the data. All of this combined shows why we do what we do, aiming to have a fraud detection system that lets employees concentrate on real problems, learn and adjust when new fraud techniques appear and perform well under high pressure.

CHAPTER 3

MATERIALS AND METHODS

3.1 Dataset Description

To train, assess and evaluate the fraud detection system, the dataset employed is very important. We extracted our information from the familiar Credit Card Fraud Detection dataset on Kaggle which was provided to us by European cardholders in 2013. The data in this set is useful for research in fraud detection since it closely corresponds to real-life financial situations and has an unequal ratio of classes.

Out of the total of 284,807 records, each transaction is labeled as either authentic (0) or fake (1). There are only 492 marked fraud transactions out of all the records which makes it an extremely imbalanced issue to solve. It's usual in financial fraud detection because there are not many fraud events, but they have to be found if they happen.

Each record includes the following:

Time: Number of seconds elapsed between each transaction and the first transaction in the dataset.

Amount: The transaction amount.

Class: Target variable — 1 for fraud, 0 for non-fraud.

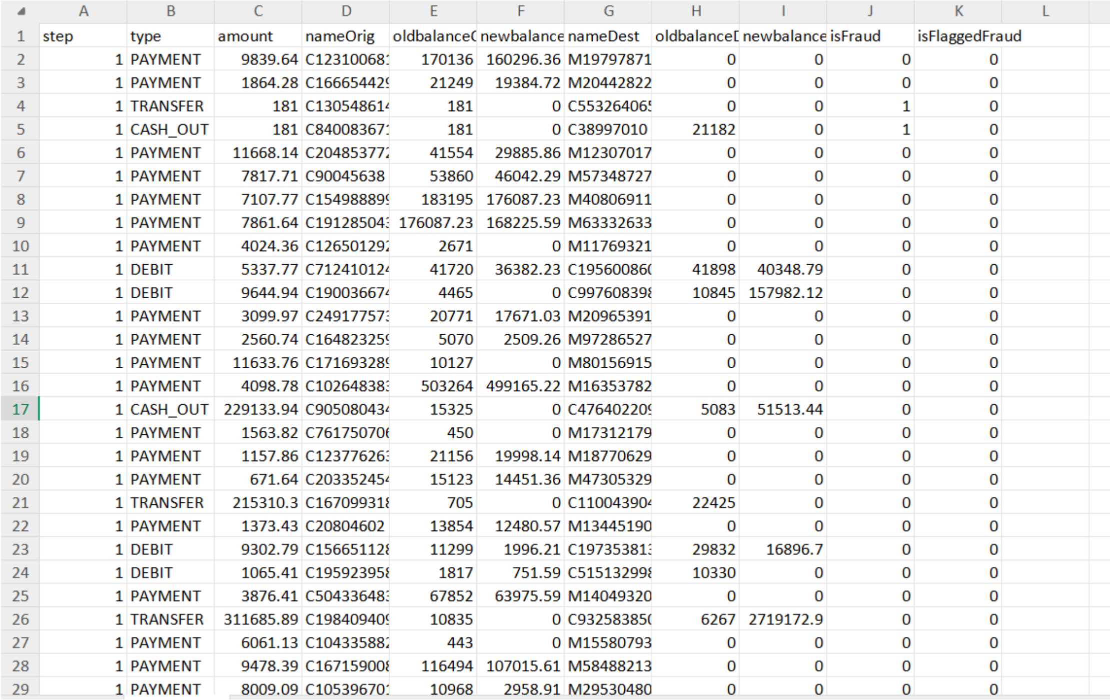
V1 to V28: A set of anonymized numerical features resulting from a **Principal Component Analysis** (PCA) transformation to protect user confidentiality.

None of the information in the dataset is personally identifiable, which guarantees both privacy and regular compliance with data protection regulations.

Standard classifiers would perform badly on this set of highly imbalanced data. Most models usually guess that every transaction is safe to increase their accuracy. As a result, Synthetic Minority Oversampling Technique (SMOTE) was introduced to increase the number of minority class (fraud) cases by generating more synthetic data.

Using stratified sampling, we made sure that the original ratios of each class remained unchanged as we divided the dataset into training and testing. Standardized values were assigned to the features by Standard-Scaler so that SGD and LSTM-based models would not be sensitive to developed features.

All in all, the dataset made it possible to check the effectiveness of several machine learning and deep learning techniques in spotting important fraud that is not always common. Thanks to its structure, gaps and changes, the data sample worked well for creating and testing a current fraud detection system. Data sample taken is shown in figure below:



◢	A	B	C	D	E	F	G	H	I	J	K	L
1	step	type	amount	nameOrig	oldbalance	newbalance	nameDest	oldbalance	newbalance	isFraud	isFlaggedFraud	
2	1	PAYMENT	9839.64	C12310068	170136	160296.36	M19797871	0	0	0	0	
3	1	PAYMENT	1864.28	C16665442	21249	19384.72	M20442822	0	0	0	0	
4	1	TRANSFER	181	C13054861	181	0	C55326406	0	0	1	0	
5	1	CASH_OUT	181	C84008367	181	0	C38997010	21182	0	1	0	
6	1	PAYMENT	11668.14	C20485377	41554	29885.86	M12307017	0	0	0	0	
7	1	PAYMENT	7817.71	C90045638	53860	46042.29	M57348727	0	0	0	0	
8	1	PAYMENT	7107.77	C15498889	183195	176087.23	M40806911	0	0	0	0	
9	1	PAYMENT	7861.64	C19128504	176087.23	168225.59	M63332633	0	0	0	0	
10	1	PAYMENT	4024.36	C12650129	2671	0	M11769321	0	0	0	0	
11	1	DEBIT	5337.77	C71241012	41720	36382.23	C19560086	41898	40348.79	0	0	
12	1	DEBIT	9644.94	C19003667	4465	0	C99760839	10845	157982.12	0	0	
13	1	PAYMENT	3099.97	C24917757	20771	17671.03	M20965391	0	0	0	0	
14	1	PAYMENT	2560.74	C16482325	5070	2509.26	M97286527	0	0	0	0	
15	1	PAYMENT	11633.76	C17169328	10127	0	M80156915	0	0	0	0	
16	1	PAYMENT	4098.78	C10264838	503264	499165.22	M16353782	0	0	0	0	
17	1	CASH_OUT	229133.94	C90508043	15325	0	C47640220	5083	51513.44	0	0	
18	1	PAYMENT	1563.82	C76175070	450	0	M17312179	0	0	0	0	
19	1	PAYMENT	1157.86	C12377626	21156	19998.14	M18770629	0	0	0	0	
20	1	PAYMENT	671.64	C20335245	15123	14451.36	M47305329	0	0	0	0	
21	1	TRANSFER	215310.3	C16709931	705	0	C11004390	22425	0	0	0	
22	1	PAYMENT	1373.43	C20804602	13854	12480.57	M13445190	0	0	0	0	
23	1	DEBIT	9302.79	C15665112	11299	1996.21	C19735381	29832	16896.7	0	0	
24	1	DEBIT	1065.41	C19592395	1817	751.59	C51513299	10330	0	0	0	
25	1	PAYMENT	3876.41	C50433648	67852	63975.59	M14049320	0	0	0	0	
26	1	TRANSFER	311685.89	C19840940	10835	0	C93258385	6267	2719172.9	0	0	
27	1	PAYMENT	6061.13	C10433588	443	0	M15580793	0	0	0	0	
28	1	PAYMENT	9478.39	C16715900	116494	107015.61	M58488213	0	0	0	0	
29	1	PAYMENT	8009.09	C10539670	10968	2958.91	M29530480	0	0	0	0	

Fig. 1. Few of the datasets used in training

3.2 Schematic Layout

The planned system architecture for Financial Fraud Detection System Using Machine Learning and Deep Learning is put together with efficiency in mind, helping to quickly analyze many transactions, find useful information and promptly predict fraud occurrences. The system begins with acquiring raw data and then moves on to several stages such as preprocessing, balancing it, training and making improvements in steps.

1. Data Collection

It begins by harvesting the transaction data from financial records. To work on this project, data from the Kaggle Credit Card Fraud Detection Dataset is made available to all. They include actual and anonymized data on transactions which help build and test accurate fraud detection systems.

2. Data Preprocessing

As soon as the data is ready, preprocessing should take place to improve its quality and usefulness. Data cleaning is done by eliminating duplicates, changing entries that don't match and handling data gaps, so the data set is trustworthy. In this stage, any numeric fields such as "Time" and "Amount", are transformed by applying the Standard-Scaler. Furthermore, because PCA anonymization was done on the source data (on the V1 to V28 features), it may not be necessary to reduce data dimensions.

3. SMOTE is a type of data balancing.

Most of the time, real transactions appear much more often than fraudulent ones in financial fraud data. The SMOTE approach is put into practice during the system's training phase to face this issue. SMOTE adds extra fraud samples, guiding models to learn better ways to identify frauds and limits the bias toward other transactions.

4. Feature Engineering

Improving data features helps a model become more accurate and apply to various cases. In this design, Transaction Amount and other main features are maintained and upsized. The data is based on pre-processed PCA, but we still try to add the original features whenever we can. In addition, feature engineering may create new variables or combinations to pick up on special patterns related to fraud in transactions.

5. Train-Test Split

Before starting model training, the data is separated into the sets used for training and testing. This method is used because the original class types should be maintained equally in the two groups. This means the model is tested on fresh data to see how it will act in a real-world situation and gives a true estimate of its accuracy.

6. Picking the Right Model and Training

Several models are arranged to check their effectiveness and select the most reliable approach to detect fraudulent acts. Since these models are known for their flexibility and dependability, we use Random Forest (RF) and Stochastic Gradient Descent (SGD) in this study. Meanwhile, Recurrent Neural Networks and Long Short-Term Memory networks are used to learn the ordering of transactions in data which can be very useful for catching complex fraud.

7. Model Evaluation

Evaluating every trained model is achieved using well-known measurements such as Accuracy, Precision, Recall, F1-score and ROC-AUC. Looking at confusion matrices and ROC curves gives more clarity about the result by pointing out false positives and false negatives. During this stage, the model that offers the best results is forwarded to produce predictions.

8. Fraud Prediction

As soon as the model is validated, it labels transactions which are (1) fraudulent and those which are (0) legitimate. Quick implementation of this feature enables companies to avoid losses and help protect their users in such cases.

9. Model Retraining

The model uses a retraining loop to ensure it keeps working well far into the future. New inputs from users, transactions marked as suspicious and extra information are used to regularly train and improve the model. This way of learning guarantees that the system can cope with recent fraud and keep giving accurate results anywhere.

The proposed **system architecture data flow** diagram is given below:

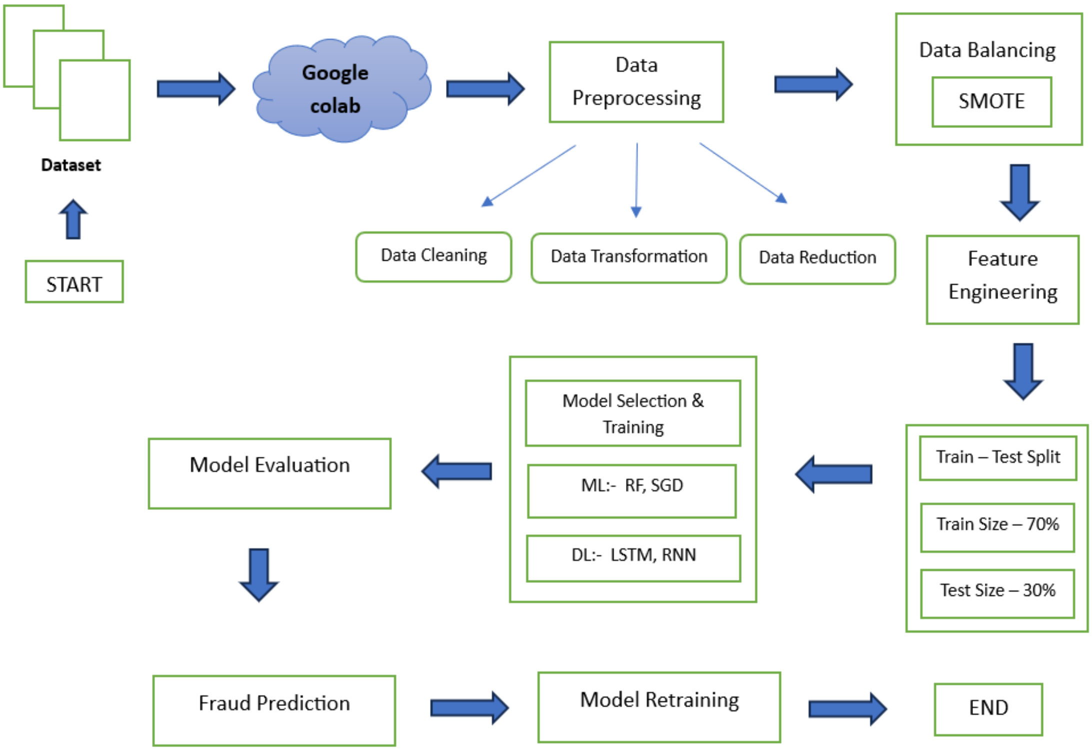


Fig. 2. Sequential Flow Diagram of Model Training

3.3 Methods Used

We designed our financial fraud detection system by using **machine learning** (ML) and **deep learning** (DL) which are good at analyzing transaction records and finding fraudulent activity. Identifying rare cases was essential since in real life banks deal with unevenly spread data about fraud. First, data is preprocessed, then efforts are made to balance the training data, choose a model, train it and evaluate its performance.

The initial step was to do detailed data preprocessing. Being that there were numeric features coming from **PCA**, we mainly worked on scaling Amount and Time through Standard-Scaler to assist in the training process. We carried out a stratified division of the data so that the rate of fraud and non-fraud stayed the same in each part. An important challenge was the large difference between rich and poor users, since the percentage of fraudulent transactions was very small. The minority class was over-represented using the **SMOTE technique** which generated new examples for this group. The change made it easier for the model to spot signs of fraud and increased the sensitivity of **recall** and **F1-score** focused models.

After that, we trained many ML and DL models to evaluate them. We chose Random Forest out of all the machine learning models because it can handle non-linear data very well. Being made up of several decision trees, Random Forest helps prevent overfitting and still allows for easy interpretation. Besides, we also used the Stochastic Gradient Descent (SGD) Classifier which is suitable for handling large datasets. Logistic loss and regularization were applied to the SGD classifier which made it a very strong starting point for linear classification.

Within the domain of **Deep Learning**, **Recurrent Neural Networks** (RNN) **and Long Short-Term Memory** (LSTM) networks were applied. Since our project involves analyzing transaction data that is sequential, these models are perfectly able to learn from that kind of information. LSTM, a more advanced kind of RNN, was very effective as it was able to save long-term connections and overcome problems regarding gradients. It enabled the model to spot fraud that develops progressively, an issue that other methods were not able to recognize. When trained on SMOTE-equipped data, the LSTM model scored highest in recall and F1-score which means it is the best fraud detection option in our system.

While evaluating the performance, we took into account several metrics such as accuracy, precision, recall, F1-score and ROC-AUC. They gave us a way to measure both the accuracy of the models and how good they were at finding fraud without giving false alarms. Missing a fraud is more problematic than issuing a false alert which is why recall, and F1-score are very important for fraud detection.

In the end, by using strong preprocessing, balancing the classes with SMOTE, combining simple and advanced machine learning and using thorough evaluation methods, We developed a model performing fearlessly, remains robust, adaptable to growth, and equipped for transformation in real scenarios as such transformations might take place in fraud.

3.4 Tools Used

For the system to be developed, it was necessary to rely on useful tools, programming libraries and platforms that allowed the data to be processed quickly and all aspects of the models to be viewed clearly. All the tools selected were known for their reliability, their connection to machine learning tools and their handling of a lot of data and calculations.

We decided to code with Python because it is simple to learn, can be used in many ways and has a large number of libraries for analysis and machine learning. Being straightforward to use and work with frameworks, Python became the top choice for experimentation and fast prototyping. We mostly used Google Colab because it allows us to write and train LSTM and RNN models for free through its GPU support, as we did not have such resources on our computers.

We relied on Pandas and NumPy to handle the dataset in different ways. With Pandas, data could be cleaned, filtered, transformed and displayed in large transactional data sets and NumPy made it easy to carry out numerical computations swiftly. It was Scikit-learn, an important ML library for Python, that we used to train our Random Forest and SGD Classifier models. It helped me to use Standard-Scaler for preprocessing, utilize confusion matrix and classification report for evaluation and apply cross-validation.

We solved the problem of class imbalance in fraud detection by making use of the imbalanced-learn library which is well-suited for scikit-learn. This library included the implementation of SMOTE, helping us to generate extra minority (fraud) cases and make the model more sensitive. We were able to find unusual fraud more effectively thanks to the use of SMOTE.

We implemented and trained the LSTM and RNN models by using Keras and TensorFlow. With these utilities, we could make neural networks with several layers, work with the recurrent layer, define the loss function and maximize the learning process. Through Google Colab and TensorFlow, we managed to reduce the time spent on training with the help of GPUs and increase our application's efficiency.

As for making graphs and exploring the results, we used Matplotlib and Seaborn. Because of them, we could draw important graphics like confusion matrices, ROC curves, heatmaps of correlation and plots of feature distributions. To see how a model performed, find patterns in the data and explain results clearly, visualization became very important.

For demonstration and deployment, we added the system to a web-based interface using the open-source Streamlit which allows for interactive applications in Python. So, we could show how the model could actually be used in detecting fraud in real time, going beyond being just for schools and other research.

All in all, our use of strong open-source tools and cloud platforms helped us efficiently analyze data, create reliable models and show the results. Together, they played a key role in making the system work well and expand.

3.5 Evaluations measures Used

Since fraudulent cases are very rare in real-life financial data sets, we must look at more measures besides accuracy when evaluating a fraud detection system. Here, traditional performance measures such as accuracy can actually give a wrong impression. A model that assumes all transactions are okay can still get extremely high accuracy yet miss out on all fraudulent activities. That's why we used a complete set of evaluation tools to get a fair and clear understanding of our models' outcomes, especially when it came to spotting rare but serious fraudulent incidents.

The main measure we looked at is accuracy, which indicates how many of the total predictions were correct. Even though it was useful for a basic idea, we did not focus on it because the data was too unbalanced for our needs. Instead of giving much importance to accuracy, we focused on precision, recall and F1-score which mean much more in fraud detection situations. It determines the amount of predictions that were indeed fraudulent. When a model's precision score is high, it becomes less likely to identify legitimate users as malicious, which helps minimize disturbing them. Still, we also need to consider recall which tells us how many fraudulent transactions the model managed to spot. It is important to identify most of the fraudulent transactions, since overlooking one could cause you to lose money.

In view of their importance to fraud detection, we set precision and recall into an F1-score by taking averages of both since it provides a unified measure for both kinds of error.

Using LSTM in combination with SMOTE got the best result, showing it can accurately find fraud and prevent many mistakes in the alerts.

We also depended on the **Receiver Operating Characteristic** (ROC) curve and the **Area Under the ROC Curve** (AUC). On an ROC curve, the recall and the false positive rate are shown against each other based on various threshold settings. The model's ability to differentiate between frauds and legitimate transactions across all possible thresholds is shown by the ROC-AUC score. If the score is close to 1.0, it shows excellent performance, but if it is only at 0.5 the person is performing randomly. Using ROC-AUC, we were able to analyze models and adjust thresholds that suit our company's risk interests.

Also, the confusion matrix helps us to see and count the numbers for: true positives (correct detection of fraud), true negatives (correct catch of non-fraud cases), False positives (legitimate transfers identified to be in error), and false negatives (cases where fraudulent transactions pass the system undetected). It provided a helpful and clear way to analyze a model's errors as well as how well it performed, improving both assessment and fine-tuning.

Thanks to these measures, we could determine which models to use which ones to optimize and how to raise performance for fraud detection. By merging accuracy and usefulness, the way we assessed the system helped it become sound for math but effective enough for real-life use in catching fraud with little disturbance.

CHAPTER 4

RESULTS/OUTPUTS

4.1 System Specification

The system behind **Financial Fraud Detection** ran well and was accurate because it depended on extensive and specific hardware and software. Because training deep learning models requires a lot of computing power and since the dataset was so large, it was necessary to have an environment that efficiently used data and allowed smooth experimentation with **machine learning** and **deep learning** models.

We processed environment using Python 3.10 since it has many data science and machine learning libraries available. All my code was executed and models built with the help of Google Colab's free GPU resources which made it much easier to work on LSTM and RNN networks.

Some of the main Python libraries used are:

1. NumPy and Pandas for efficient data manipulation and preprocessing
2. Scikit-learn can be used to build traditional models such as Random Forest and SGD Classifier and also offers Standard-Scaler for preprocessing and several evaluation measurements.
3. Using imbalanced-learn made it possible to handle the problem of unequal classes in the dataset.
4. TensorFlow and Keras are especially good for constructing, training and improving deep learning models, especially LSTM and RNN models.
5. Data visualization like Matplotlib and Seaborn are use to plot the feature distribution, determine confusion matrices and create ROC curves.

The models were built and tested mainly on Google Colab which came with up to 12 GB of RAM and GPU support. When targeting local testing, having an Intel Core i5 processor, 8 GB RAM and SSD storage was enough for basic ML models, whereas deep learning will require compatibility with GPU.

we used Windows 10 locally and Google Colab for operating systems. Both versions and data were managed using Google Drive and files were imported and exported from the cloud to make things easier and quicker.

Thanks to cloud computing and open-source libraries, we were able to develop, train, examine and present an intelligent fraud detection system that performs well under real-life conditions. The choices we made for the system specs resulted in an efficient balance between how the model performs, who can access it and how scalable it is.

4.2 Parameters Used

While developing and training the Financial Fraud Detection System, important settings were modified in machine learning and deep learning models so that they would work better at spotting rare fraudulent transactions. Because of these parameters, the way a model learned from data, generalized from it and balanced false positives and false negatives was directly affected.

The parameters used for Random Forest and SGD Classifier in machine learning are listed below.

Random Forest:

1. n_estimators is set to 100: Means how many decision trees the model should build. Higher sample sizes get closer to the actual value, but it will take more effort to calculate.
2. When max-depth is left to None, the tree does not stop expanding and may capture more fraud.

3. Setting criterion to ‘gini’ helps decide how strong the split is. It was helpful in finding out which features were important for the model.

4. If you set class_weight to ‘balanced’, this will automatically equalize weights depending on the number of samples in each class.
5. Random_state is set to 42: which implies that I am using the same random behavior each time, thus getting the same results every time I run this code.

SGD Classifier:

1. loss = log_loss: we are using logarithmic loss to measure how well the model's predicted probabilities match the actual class labels
2. penalty = 'l2': A method used to control overfitting.
3. alpha = 0.0001: This means the update values for the model are very small.
4. max_iter = 1000: Sets 1,000 as the max limit for the algorithm to stop changing.
5. When you set class_weight to ‘balanced’, the model understands the rarity of the minority class (fraud) better.
6. Setting random_state to 42 keeps the results the same when the model is run again.

We optimized the parameters of RNN and LSTM based neural networks for sequence learning tasks and for them to perform well.

RNN and LSTM:

1. You should use input_shape = (X_train.shape[1], 1) to allow for time-series data used in sequential modeling.
2. The hidden layer has 64 neurons which means it has 64 "mini brains" working together to understand the information.
3. increase dropout to 0.2: Choose at random 20% of neurons and disable them during training to handle overfitting.
4. By setting recurrent_dropout to 0.2, it means Randomly ignore 20% of those memory connections during training so you don’t become too dependent or overconfident.
5. batch_size = 64: It defines the number of samples that are used for learning before the model changes its weights.
6. Run training through the data set 10 to 20 times without early stopping to avoid

either underfitting the model or overfitting it.

7. You can use ‘adam’ as the optimizer, as it adjusts the learning rate for you during training.
8. The loss function used is called binary_crossentropy.

For preprocessing and leveling out the amount of data in each class:

1. It is commonly known as SMOTE (Synthetic Minority Oversampling Technique).
2. Samples of the minority class are made through the use of K neighbors.
3. Only applied on training set so that no information from the test set is used and overfitting is avoided.

Parameters in all models were adjusted by hand or using grid search (where applicable) to get the best results and we did this by focusing on recall and F1-score because finding fraud is essential.

4.3 Experimental Outcomes

Selecting these features and changing them appropriately meant that each model could be trained effectively, function consistently and react accurately to the challenges found in detecting fraud.

The comparison of all models before and after applying SMOTE is provided in the table 3 and the ROC curves for the following are in figure 3 below:

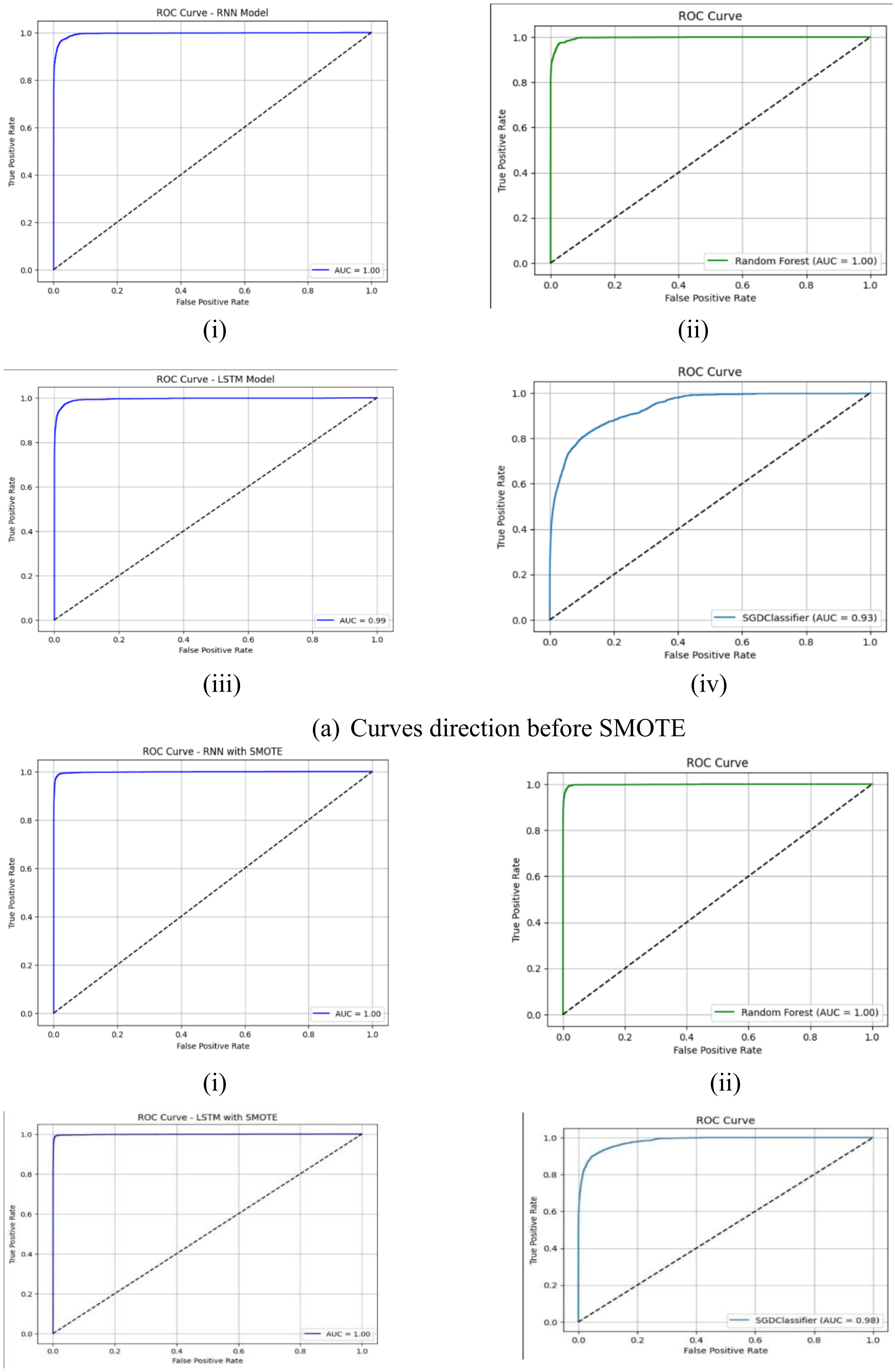
Table 2. Results of our models

(a) Before applying the SMOTE

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Random Forest	99.96%	99.37%	72.03%	83.52%	99.66%
SGD	99.88%	97.73%	8.83%	16.20%	93.41%
RNN	99.95%	95.95%	63.20%	76.21%	99.54%
LSTM	99.95%	98.03%	63.45%	77.04%	99.45%

(b) After applying the SMOTE

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Random Forest	98.44%	7.49%	98.81%	13.92%	99.84%
SGD	93.24%	1.70%	91.66%	3.35%	99.86%
RNN	99.07%	11.90%	97.99%	21.23%	99.82%
LSTM	**99.69%**	**28.62%**	**97.25%**	**44.22%**	**99.83%**



(i) (ii)

(iii) (iv)

(a) Curves direction before SMOTE

(i) (ii)

(iii) (iv)

(b) Curve direction after SMOTE

Fig. 3. ROC-AUC Curve of the models

CHAPTER 5

CONCLUSION

In the project, we developed a system that uses machines for learning and deep learning to find and stop fraudulent transactions in financial activities. We wanted to handle the issues of uneven data in fraud cases and the requirement for immediate detection which help prevent financial losses and keep online transactions secure.

In the process of working on the project, we worked with different algorithms such as the Random Forest, SGD Classifier, RNN and LSTM. The best of these was LSTM, mainly because it could notice trends in data over time and remember long-term details. With this function, the model can see how users act over time and decide if the transactions are real or being carried out by fraudsters.

The model is especially beneficial in fraud detection systems since it helps decrease false positives and raises precision which matter a lot as mistaking a real user for fraudulent can give bad service experiences and missing any suspicious activity can be very expensive. After learning each process and pattern in transactions, the LSTM model could increase recall safely, keeping precision almost the same which resulted in reliable performance.

All in all, applying LSTM networks to fraud detection improves the process by making it more accurate, precise and reliable. Not only does this protect the financial sector, but it also promotes more trust in digital banking and payment services, something important in the modern, fast-changing world of digital economy.

CHAPTER 6

REFERENCES

[1] Computers & Security Volume 28, Issue 6, September 2009, Pages 381-394 "**A survey of signature based methods for financial fraud detection**" *Michael Edward Edge, Pedro R. Falcone Sampaio*

[2] Decision Support Systems Volume 50, Issue 3, February 2011, Pages 559-569 "**The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature**" *E.W.T. Ngai a, Yong Hu b, Y.H. Wong a, Yijun Chen b, Xin Sun b*

[3] *A. Mousa*, "**Detecting Financial Fraud Using Data Mining Techniques: A Decade Review from 2004 to 2015**," J. Data Sci., vol. 14, no. 3, pp. 553–570, 2016.

[4] *A. M. Mubalaike and E. Adali*, "**Deep Learning Approach for Intelligent Financial Fraud Detection System**," in UBMK 2018 - 3rd International Conference on Computer Science and Engineering, 2018.

[5] *Yisong Chen, Chuqing Zhao, Yixin Xu, Chuanhao Nie,* "**Year-over-Year Developments in Financial Fraud Detection via Deep Learning: A Systematic Literature Review",** February 4, 2025

[6] "**Financial Fraud Detection Based on Machine Learning: A Systematic Literature Review**" *by Abdulalem Ali, Shukor Abd Razak, Siti Hajar Othman, Taiseer Abdalla Elfadil Eisa, Arafat Al-Dhaqm, Maged Nasser, Tusneem Elhassan 1, Hashim Elshafie, and Abdu Saif*

[7] *Andrea Dal Pozzolo, Olivier Caelen, Reid A. Johnson, Gianluca Bontempi.* "**Credit Card Fraud Detection: A Realistic Modelling and a Novel Learning Strategy.**" In *IEEE Transactions on Neural Networks and Learning Systems*, 2015.

[8] *Aravind Kumar Kalusivalingam, Amit Sharma, Neha Patel, Vikram Singh*, "**Enhancing Financial Fraud Detection with Hybrid Deep Learning and Random Forest Algorithms**"

[9] *Ibtissam Benchaji, Samira Douzi, and Bouabid El Ouahidi*, **"Credit Card Fraud Detection Model Based on LSTM Recurrent Neural Networks",** Faculty of Sciences IPSS, University Mohammed V, Rabat, Morocco

[10] *Md Al-Imran, Eftekhar Hossain Ayon,* **"Transforming banking security: the role of deep learning in fraud detection systems"**

[11] *D. O. Njoku, V. C. Iwuchukwu, J. E. Jibiri*, **"Machine Learning Approach for Fraud Detection System in Financial Institution: A Web Base Application"**

[12] *A. Dal Pozzolo et al.,***"Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy"**

[13] *Oluwabusayo Adijat Bello; Adebola Folorunso2,* **"A Comprehensive Framework for Strengthening USA Financial Cybersecurity: Integrating Machine Learning and AI in Fraud Detection Systems"**

[14] https://www.kaggle.com/datasets/jainilcoder/online-payment-fraud-detection (Kaggle Financial Fraud dataset)

CHAPTER 7

REFLECTION OF TEAM MEMBERS ON THS PROJECT

Aditya Narayan Sahu

We gained important experience by working together as a team on a project about financial fraud detection using machine learning. We found ways to split our work according to each person's skills, report achievements clearly, and figure out problems as a whole team. All of us found out during the project that it is crucial to leave nothing to chance, keep everyone informed, and repeat experiments when building any strong solution. We noticed that handling unbalanced data is very important in real issues of fraud detection.

It was up to me to carry out the literature survey, gather the required data, use SGD for training, and boost the performance using SMOTE. I gained additional knowledge about supervised learning algorithms and how using accuracy, precision, recall, F1-score, and ROC-AUC makes it easier to assess how well a model is performing. Using SMOTE to overcome class imbalance was very important for the model's ability to detect examples from the minority class. Because of these tasks, I gained knowledge in technical areas and learned how to handle problems in a systematic way.

The fact that we were organized from the beginning made our design process successful. Our process began by focusing on the problem, searching for available solutions in research, and carrying out activities in order from getting the data to assessing the models. We explained that using SGD and SGD + SMOTE modified our model selection and data preprocessing processes. An important problem was that we didn't experiment with advanced models or combine several methods to improve accuracy. Additionally, since SMOTE improved our achievements, we didn't try much to optimize the parameters and validate our model using cross-validation.

Satyabrat Sahoo

By doing this project as a team, we learned important things about creating a financial fraud detection system with both machine learning and deep learning methods. We found out that the process works better when we plan and divide tasks together, and everyone's efforts are directed toward the same objective. While looking at different models, we understood that recall, F1-score, and AUC-ROC are important, most especially when the data set has a significant imbalance. Google Colab allowed us to use GPU-accelerated technologies, which made our process both effective and ready for scaling.

For my part, I worked on the data visualization, trained the LSTM model, and used the SMOTE method to manage the class imbalance. This made it clear to me how time-series data works, how LSTM can display changes over time, and why handling data imbalance is important for having better models. This exercise allowed me to see how to measure the performance of models and select the most suitable one for employment.

We used a planned approach, starting with easy models and going more in-depth with learners to include balancing the classes. Even though our model was successful and dependable, we missed the chance to input real-time data into the design at the beginning. From this reflection, we will know how to further improve our work on real-time fraud detection.

Soumya Ranjan Patra

By working as a group, we learned how to build a system for finding financial fraud by using both machine learning and deep learning. We understood that it was necessary to make a plan, talk openly with one another, and distribute tasks according to every person's strengths. While exploring the different models, we found that recall, F1-score, and AUC-ROC should be used as evaluation metrics when we encounter imbalanced data in fraud detection problems. We made use of Google Colab, allowing us to use GPUs and improve how the training worked with bigger models.

I was responsible for trying out and testing different models, with a main focus on the Random Forest classifier, at first without SMOTE and then with the technique included for handling unbalanced classes. Working on this project helped me see what effect data imbalance might have on model results and how tricks like SMOTE can make it easier for my model to find instances of fraudulent transactions. I also worked on evaluating different models and decided on the best one based on performance, not only on how accurately it worked.

We followed a system in our work; starting with simpler models and step by step moving to more complex designs while dealing with data challenges. The reliability of our models was boosted by relying on both the SMOTE approach and appropriate ways to evaluate them. Still, there was a limitation because we did not add real-time data in the beginning, which could have made the project more valuable. It would be best in the next phase to consider how our solution will respond and function in real-time, as this would help it work better for actual needs.

Achyuta Samantaray

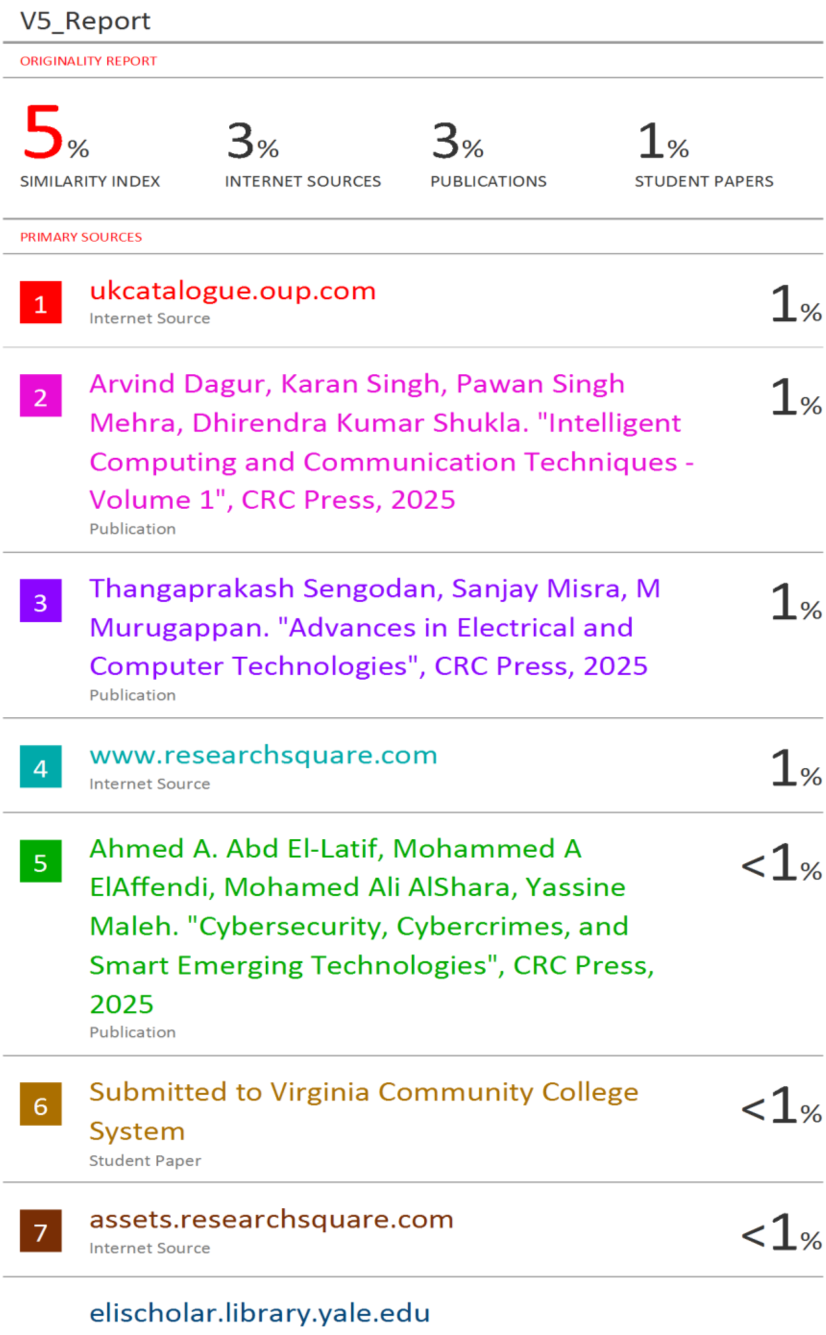
Since we did our work collectively, we discovered a great deal about constructing a financial fraud detection system that relies on machine learning and deep learning. For the project to run well, people will need to follow a plan, divide tasks clearly, and cooperate when solving issues. To review many models, we were reminded that recall, F1-score, and AUC-ROC should be used, especially since our data are often unbalanced. I am now getting better results in both efficiency and scale since using Google Colab with GPU.

I was in charge of doing the data pre-processing, training the RNN model before and after SMOTE application, and securing the documentation for this project. Because of this, I understood how time series data is used for deep learning, as well as the effect of preprocessing steps. By using SMOTE, I found out how to balance my dataset. I also made my documentation better because there is a proper process and result that can be used again.

We followed a set process when coming up with our design. At the start, we had models that hadn't yet learned, taught ourselves the advanced techniques, managed our schedule, and grew a trustworthy system by taking things step by step and learning from our efforts. The initial prototype could have been better since it did not take in real-time data, which is important for real fraud-catching. Yet, this is now going to be a main challenge as we seek to make our work better suited for the world we live.

CHAPTER 8

SIMILARITY REPORT



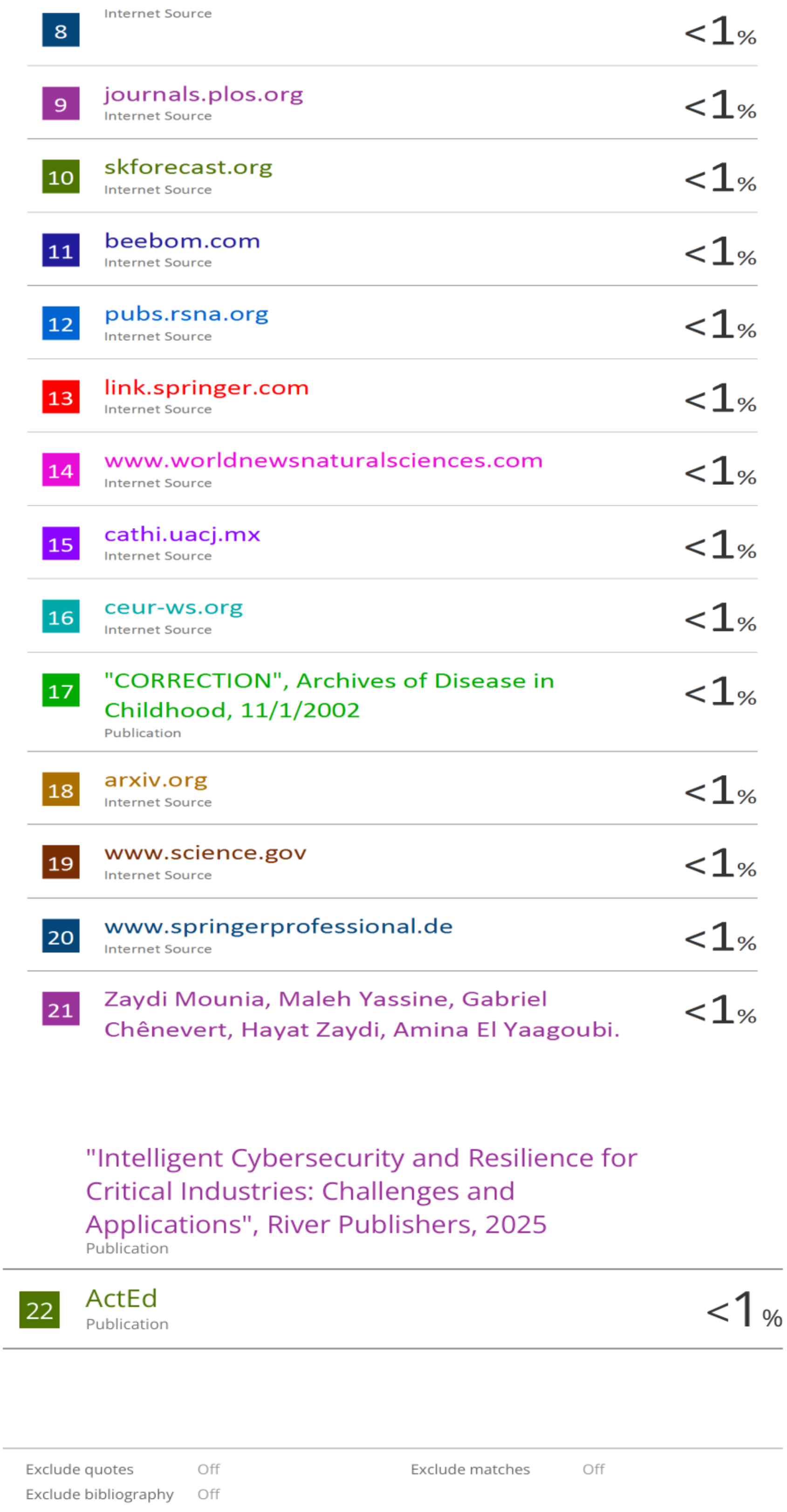
V5_Report

ORIGINALITY REPORT

5%	3%	3%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ukcatalogue.oup.com Internet Source	1%
2	Arvind Dagur, Karan Singh, Pawan Singh Mehra, Dhirendra Kumar Shukla. "Intelligent Computing and Communication Techniques - Volume 1", CRC Press, 2025 Publication	1%
3	Thangaprakash Sengodan, Sanjay Misra, M Murugappan. "Advances in Electrical and Computer Technologies", CRC Press, 2025 Publication	1%
4	www.researchsquare.com Internet Source	1%
5	Ahmed A. Abd El-Latif, Mohammed A ElAffendi, Mohamed Ali AlShara, Yassine Maleh. "Cybersecurity, Cybercrimes, and Smart Emerging Technologies", CRC Press, 2025 Publication	<1%
6	Submitted to Virginia Community College System Student Paper	<1%
7	assets.researchsquare.com Internet Source	<1%
	elischolar.library.yale.edu	



8	Internet Source	<1%
9	journals.plos.org Internet Source	<1%
10	skforecast.org Internet Source	<1%
11	beebom.com Internet Source	<1%
12	pubs.rsna.org Internet Source	<1%
13	link.springer.com Internet Source	<1%
14	www.worldnewsnaturalsciences.com Internet Source	<1%
15	cathi.uacj.mx Internet Source	<1%
16	ceur-ws.org Internet Source	<1%
17	"CORRECTION", Archives of Disease in Childhood, 11/1/2002 Publication	<1%
18	arxiv.org Internet Source	<1%
19	www.science.gov Internet Source	<1%
20	www.springerprofessional.de Internet Source	<1%
21	Zaydi Mounia, Maleh Yassine, Gabriel Chênevert, Hayat Zaydi, Amina El Yaagoubi. "Intelligent Cybersecurity and Resilience for Critical Industries: Challenges and Applications", River Publishers, 2025 Publication	<1%
22	ActEd Publication	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		