RV Educational Institutions ®
RV College of Engineering ®
Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi
Approved by AICTE, New Delhi

Go, change the world®

“Fraudulent Credit Card Detection Using Ensemble Machine Learning Models”

PROJECT REPORT ON

ARTIFICIAL INTELLIGENCE & MACHINE LEARNING Lab(18CS62)

VI SEMESTER

2021-2022

Submitted by

Sinchana Raj USN: 1RV19CS158

T J S L Savitri USN: 1RV19CS171

Under the Guidance of

Dr. Hemavathy R

Associate Professor

Department of CSE, RVCE.

RV COLLEGE OF ENGINEERING®, BENGALURU - 560059

(Autonomous Institution Affiliated to VTU, Belagavi)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the Lab Project report work titled "Fraudulent Credit Card Detection Using Ensemble Machine Learning Models" has been carried out by Sinchana Raj (1RV19CS158) and T J S L Savitri (1RV19CS171), bonafide students of RV College of Engineering, Bengaluru, have submitted in partial fulfillment for the **Assessment of Course: Artificial Intelligence & Machine Learning (18CS62) – Lab Component** during the year 2021-2022. It is certified that all corrections/suggestions indicated for the internal assessment have been incorporated in the report.

Dr. Hemavathy R,	**Head of Department**
Associate Professor,	Department of CSE,
Department of CSE,	RVCE, Bengaluru–59

External Viva

Name of Examiners	**Signature with Date**
1.	
2.	

RV COLLEGE OF ENGINEERING, BENGALURU® - 560059

(Autonomous Institution Affiliated to VTU, Belagavi)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We, **Sinchana Raj (1RV19CS158) and T J S L Savitri (1RV19CS171),** the students of 6th Semester B.E., Department of Computer Science and Engineering, R.V. College of Engineering, Bengaluru hereby declare that the Lab -project titled **"Fraudulent Credit Card Detection Using Ensemble Machine Learning Models"** has been carried out by us and submitted in partial fulfillment for the **Assessment of Course: Artificial Intelligence & Machine Learning (18CS62) lab component** during the year 2021-2022.

Place: Bengaluru

Date: 13th August, 2022

Name	**Signature**
1. SINCHANA RAJ (1RV19CS158)	
2. T J S L Savitri (1RV19CS171)	

ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped us in carrying out this project work. We would like to take this opportunity to thank them all.

We deeply express our sincere gratitude to our guide **Dr. Hemavathy R,** Associate Professor**,** Department of CSE, RVCE, Bengaluru, for her able guidance, regular source of encouragement and assistance throughout this project.

We would like to thank **Dr.Ramakanth Kumar P**, Head of Department, Computer Science & Engineering, R.V.C.E, Bengaluru, for his valuable suggestions and expert advice.

We express sincere gratitude to **Dr. Subramanya K N**, Principal, R.V.C.E, Bengaluru, for his moral support towards completing our project work.

We thank all the **teaching staff and technical staff** of the Computer Science and Engineering department, RVCE for their help.

Lastly, we take this opportunity to thank the **family** members and **friends** who provided all the backup support throughout the project work.

Abstract

In recent times, e-commerce has become an inevitable part of people's lives. Due to the digitalization of modern life, consumers from all over the world benefit from the perks of online transactions. Due to the continuous growth of the internet and its accessibility, the number of digital buyers keeps increasing every year. One of the most used modes of digital transactions includes credit cards which act as a significant payment tool due to the convenience of an instant line of short-term credit while making transactions. However, they also run the risk of being targeted for fraud. Credit card frauds are easy and friendly targets. E-commerce and many other online sites have increased the online payment modes, increasing the risk for online frauds. Increase in fraud rates, researchers started using different machine learning methods to detect and analyse frauds in online transactions. The main aim of this paper is to design and implement a detection mechanism for fraudulent credit card transactions using machine learning techniques in an e-commerce website.

The technique we used for dealing with imbalanced data is Autoencoder. The dataset collected has a lot less positive class. In order to increase the positive (fraudulent) class we are oversampling. Oversampling comes with a lot of noise. So, we are going with dimensional reduction. We need to reduce data in such a way that we preserve the important data too. This can be achieved through the technique of auto encoder ANN which is used to extract a latent representation of the training data. Compression of the data might improve the inherent information. All models will be evaluated using the Area Under the Receiver-Operating Characteristic Curve (ROC curve) score, because confusion matrix accuracy is not meaningful for unbalanced classification. Results are observed under three different situations. First is, when the data is oversampled and directly fed to the ANN. Second is when the data is standardized and scaled before giving to the ANN model. Third is when an autoencoder is used to sample the data and the sampled data is fed into the autoencoder.

The results from the three cases obtained are analysed. When the oversampled data is fed to the ANN model, it has classified all the classes as non-fraud classes making the AROC score for at all the thresholds equal to 0.5. When the data is standardized and scaled, the AROC score is comparatively low and decreases as the threshold increases from 0.1 to 0.9. On using the auto-encoder, the AROC improved significantly to around 0.95.

List of Tables

List of Figures

Table of Contents

1. Introduction

Credit cards play a powerful role in carrying out online transactions. It is one of the major modes of online transactions. It is a financial instrument issued by banks with a present credit limit helping a customer make cashless transactions. Credit cards allow a customer to easily avail an instant line of short-term credit while making transactions. This helps increase the purchasing power of the customer while also offering benefits like ease of use, reward points and cashback. Regular, on-time payments can also improve your credit score which leads to easier long-term loan approvals. However, alongside all these merits, credit cards also run the risk of being targeted for fraud. There are different types of credit card fraud based on the nature of fraudulent activities such as card getting stolen, obtaining cards using false information, individuals using credit cards while being unable to pay debts, bank employees stealing card details to use it remotely, individuals using skimming devices to hack credit card details, etc.,

1.1. Project domain and problem addressing

With different frauds, mostly credit card frauds, often in the news for the past few years, frauds are at the top of mind for most of the world's population. The credit card dataset is highly imbalanced because there will be more legitimate transactions when compared with fraudulent ones. As advancement, banks are moving to EMV cards, which are smart cards that store their data on integrated circuits rather than on magnetic stripes, have made some on-card payments safer, but still leaving card-not-present frauds at higher rates. According to a 2017 US Payments Forum report, criminals have shifted their focus on activities related to CNP transactions as the security of chip cards has increased. Even then there are chances for thieves to misuse the credit cards. There are many machine learning techniques to overcome this problem. Our main area of domain in credit card frauds are Card-Not-Present (CNP) frauds.

1.2. Issues and Challenges

- Enormous data is processed every day and the model build must be fast enough to respond to the scam in time
- There is an issue of imbalanced data. Most of the transactions are not fraudulent which makes it really hard for detecting the fraudulent ones
- Data availability is difficult as the data is mostly private.

- Misclassified Data can be another major issue, as not every fraudulent transaction is caught and reported.
- After facing numerous solutions to tackle the earlier frauds, scammers have developed adaptive techniques against the model improvising their fraud techniques

1.3. Need for AI-based solutions

According to a 2017 US Payments Forum report, criminals have shifted their focus on activities related to CNP transactions as the security of chip cards has increased. Even then there are chances for thieves to misuse the credit cards. There are many machine learning techniques to overcome this problem. Machine learning has become an increasingly accessible and reliable method to detect fraudulent transactions. Using a historical dataset, a machine learning model can be trained to learn patterns behind fraudulent behaviour. A model can then be applied to filter out fraudulent transactions and stop them from occurring in real time. Artificial neural network (ANN) models are much better than conventional fraud detection models. They can recognise thousands of patterns from large datasets. ANN offers an insight into how users behave by understanding their app usage, payments, and transaction methods. Some of the benefits of fraud detection using ANN are faster detection, higher accuracy and improved efficiency with larger data.

1.4. Problem statement

The problem statement that we tried to explore is detection of fraudulent credit cards using machine learning models like the autoencoder model to balance the imbalance data and then feed the balanced data to four-layered ANN and analyse the AROC score at different thresholds.

1.5. Summary

This chapter summarizes the introduction of the project, the problem addressing, project domain, issues and challenges, need for AI and the problem statement in detail.

2. Literature Survey

A literature review surveys books, scholarly articles, and any other sources relevant to a particular issue, area of research, or theory, and by so doing, provides a description, summary, and critical evaluation of these works in relation to the research problem being investigated. Literature reviews are designed to provide an overview of sources you have explored while researching a particular topic and to demonstrate to your readers how your research fits within a larger field of study. A literature review for the credit card detection has been thoroughly done and table 2.1. shows the tabulated analysis of the survey consisting of the title, authors, objectives, results and gaps identified.

Table 2.1. Literature survey for the project

Sl. No.	Title of the work	Authors & Publication Details	Objectives of the work carried out	Results obtained	Gaps identified
1	Credit Card Fraud Detection using Machine Learning Algorithms	Varun Kumar K S, Vijaya Kumar V G, Vijay Shankar A, Pratibha K Published in IJERT, ISSN: 2278-0181, Vol. 9, Issue 07, July 2020	To implement machine learning algorithms to detect credit card fraud with respect to time and amount of transaction.	By using the time and amount feature in the data set given in the Kaggle. First, we build the model using some machine learning algorithms such as logistic regression, decision tree, and support vector machine, these all are supervised machine learning algorithms in	Artificial neural networks in deeps learning can be used to replace the machine learning algorithms for better prediction, ANN is having different types of layers such as an input layer, a number of middle layers having activation functions for the action of neurons and the output layer having some kind of activation

				machine learning.	function like sigmoid and weight initialization and reinitialization in backward propagation for reducing the error between actual and predicted values.
2	Credit card fraud detection using machine learning algorithm s.	Vaishnavi Nath Dornadulaa, Subbiah Geetha, Vellore Institute of Technology , Chennai-600127, India Published in the journal: Procedia Computer Science, in the year 2019	To design and develop a novel fraud detection method for Streaming Transaction Data, with an objective, to analyze the past transaction details of the customers and extract the behavioral patterns.	Logistic regression, decision tree and random forest are the algorithms that gave better results. It was also observed that the Matthews Correlation Coefficient was the better parameter to deal with imbalance dataset	Performance of Logistic Regression, K-Nearest Neighbor, and Naïve Bayes are analyzed on highly skewed credit card fraud data. Through supervised learning, methods can be used they may fail at certain cases of detecting fraud cases.
3	Enhanced credit card fraud	Benchaji, Ibtissam and Douzi,	To develop a novel system for credit card	The proposed model is capable of catching	Lack of a model that relies solely on attention and

	detection based on attention mechanism and LSTM deep model	Samira and Ouahidi, Bouabid and Jaafari, Jaafar Published in the Journal: Journal of Big Data, in the year 2021	fraud detection based on sequential modelling of data, using attention mechanism and LSTM deep recurrent neural networks.	useful patterns within consumer behaviour which helps to distinguish effectively fraudulent transactions from the normal ones.	transformers architecture without using any recurrent networks to process sequences.
4	Credit Card Fraud Detection in Card-Not-Present Transactions: Where to Invest?	Igor Mekterović ,Mladen Karan,Damir Pintar and Ljiljana Brkić. Faculty of Electrical Engineering and Computing, University of Zagreb, 10000 Zagreb, Croatia Published in the journal: Applied	To detect challenges in the fraud detection problem such as feature engineering and unbalanced datasets and distinguish between more and less lucrative areas to invest in when upgrading fraud detection systems	The research shows room for improvement in the existing system and that one should foremost invest in feature engineering and model tuning. All data mining models performed better than the existing system, whereas random forest performed best.	The main advantage of ensemble models is their increased accuracy, but this comes at a raised computation cost and less intuitive or non-existent interpretation. However, the work doesn’t deal with practical issues that we considered here, like cost-efficiency, scalability, maintenance, etc.

		Sciences in the year 2021			
5	Credit Card Fraud Detection System	Kartik Madkaikar, Manthan Nagvekar, Preity Parab, Riya Raikar, Supriya Patil Published in IJRTE, ISSN:2277-3878, Volume – 10 Issue – 2, July 2021	To compare and analyze Machine Learning algorithms such as Logistic Regression, Naïve Bayes, Random Forest, K-Nearest Neighbor, Gradient Boosting, Support Vector Machine, and Neural Network algorithms for fraud detection and to identify an optimal solution.	Of the various Machine Learning algorithms implemented in this paper, the Gradient Boosting algorithm provides an edge over the other algorithms. Gradient Boosting outperformed with an accuracy of 95.9%.	Used two methods under random forests namely Random-tree-based random forest and classification and regression tree (CART)-based to train the behavioural features of normal and abnormal transactions. The random forest algorithm performed better on a small dataset, but imbalanced data reduced the accuracy.
6	A survey paper on credit card fraud detection	Aisha Mohammad Fayyomi, Derar Eleyan,	This paper aims in using the multiple algorithms of	In this model, using an artificial neural network (ANN) which	One limitation is the use of single models for developing the fraud detection

	techniques.	Amina Eleyan Published in IJSTR, ISSN: 2277-8616, Volume-10, Issue-9, September 2021 edition	Machine learning such as support vector machine (SVM), k-nearest neighbour (Knn) and artificial neural network (ANN) in predicting the occurrence of the fraud. It also aims to conduct differentiation of the accomplished supervised machine learning and deep learning techniques to differentiate between fraud and non-fraud transactions.	gives accuracy approximately equal to 100% is best suited for credit card fraud detection. It gives accuracy more than that of the unsupervised learning algorithms. In this research work, data pre-processing, normalization and under-sampling were carried out to overcome the problems faced by using an imbalanced dataset.	framework in this study. To further enhance the developed framework, hybrid models can be formed using a combination of two or more models (Jiang et al., 2020). Hybrid models enable the use of more than one model to determine the transaction legitimacy, in order to further improve the fraud detection rate.
7	An intelligent payment card fraud	M. Seera, C. Lim, Ajay Kumar,	A statistical hypothesis test is conducted to evaluate	In our analysis, a total of thirteen statistical and machine learning	The current study can be improved from several angles. Firstly, the

	detection system	Lalitha Damotharan , K. Tan, Published in the journal: Annals of Operations Research, in the year 2021	whether the aggregated features identified by a genetic algorithm can offer better discriminative power, as compared with the original features, in fraud detection. The outcomes positively ascertain the effectiveness of using aggregated features ndertaking real-world payment card fraud detection problems.	methods, ranging from ANN to deep learning models have been used for evaluation. Three benchmark credit card data sets obtained from a public repository have been used for performance assessment. The AUC metric is employed, which indicates statistical differences in the performance of various detection methods. The best AUC score achieved is 0.937 from GBT for the Australian data set.	real payment card database used is limited to a financial institution in Malaysia. The transactions mostly occurred in the Asia region. It would be useful to acquire more real-world data from different regions, in order to fully evaluate the effectiveness of the developed method for detecting fraud in other regions around the world.
8	A Survey of Online Card Payment	Bemali Wickraman ayake and Dakshi	This survey proposes a taxonomy based on	With this survey, we were able to identify the main areas considered	In the card payment fraud detection domain, few areas remain

	Fraud Detection using Data Mining-based Methods	Kapugama Geeganage and Chun Ouyang and Yue Xu Published in the journal: ArXiv, in the year 2020	existing research attempts and experiments, which mainly elaborates the approaches taken by researchers to incorporate the (i) business impact of fraud (and fraud detection) into their work, (ii) the feature engineering techniques that focus on cardholder behavioural profiling to separate fraudulent activities happening with the same card, and (iii) the adaptive efforts taken to address the changing	were handling the cost sensitivity of the problem, as well as handling the speed of processing and transaction authentication. Further, we evaluated different approaches taken to profile the cardholder behaviour to enable machine learning models to distinguish fraudulent transactions better. We classified these methods based on the logic used to profile the cardholder and the breadth of information used.	challenges for developing a perfect solution that could safeguard the cardholders and institutions against fraud. Data collection, due to availability of data due to confidentiality and sufficiency of the information. Data Labeling, due to the reliability of human labelling and Model Latency, due to its impact on commercial grade operation are the issues that are discussed in this section.

			nature of fraud.		
9	Credit Card Fraud Detection Using Autoencoder Neural Network	Ping Jiang, Jinliang Zhang, Junyi Zou Published in August 2019	This paper proposes a denoising autoencoder neural network (DAE) algorithm which can not only oversample minority class samples through misclassification cost, but it can denoise and classify the sampled dataset.	This study combined stacked denoising autoencoder neural networks with oversampling to build the model, which can achieve minority class sampling on the basis of misclassification cost, and denoise and classify the sampled datasets. The proposed algorithm increases the classification accuracy of minority classes compared to the former methods; we can achieve different accuracy by controlling the threshold. In this	Do not give importance to the dimensionality reduction of high dimensional data. The area needs to be further researched.

				study, when the threshold equals 0.6, we can achieve the best performance, which is 97.93%.	
10	Credit Card Fraud Detection Using Autoencoder Model in Unbalanced Datasets	August 2019 Journal of Advances in Mathematics and Computer Science Authors: Mohammed Abdulhameed Al-Shabi Taibah University	This paper aims to propose an efficient approach that automatic detects fraud credit card related to insurance companies using deep learning algorithm called Autoencoders. The effectiveness of the proposed method has been proved in identifying fraud in actual data from transactions	In this paper, some advanced techniques have been introduced to detect the fraud credit card of the insurance company. This study reviewed how machine learning can be used to address some of the issues of financial fraud detection in credit cards. The focus on the design model is capable of reporting the most fraud transactions for investigators using an autoencoder	The recommendation of the paper lies in the following suggestions for improvements to the current algorithm: Appling fraudulent work to different classification algorithms and compare them with this model; inserting a random value in an attempt to confuse the fraudsters and disrupt their previously acquired knowledge; and applying this algorithm to the

			made by credit cards in September 2013 by European cardholders. In addition, a solution for data unbalancing is provided in this paper, which affects most current algorithms.	algorithm that can deal with unbalanced datasets. The algorithm was able to detect between 64% at the threshold = 0.5, 79% at the threshold = 0.3 and 91% at threshold= 0.07	data of Saudi companies and financial institutions.

3. Design Details

The design details are an early phase of a project where the project's key features, structure, criteria for success, and major deliverables are planned out. The aim is to develop one or more designs that can be used to achieve the desired project goals. Stakeholders can then choose the best design for the execution of the project. The project design steps might generate various outputs, such as sketches, flowcharts, site trees, HTML screen designs, prototypes, photo impressions, and more.

3.1. Architecture

The proposed architecture is basically designed to detect online payment credit card fraud, and emphasis is placed on providing a system of fraud prevention to verify a transaction as fraudulent or legitimate. It is assumed that the issuer and the acquirer bank are linked with each other for implementation purposes. To enforce this program in a real-time scenario, sharing best practices and increasing consumer awareness among people can be very helpful in reducing the losses caused by fraudulent transactions. It is depicted in Fig 3.1.1

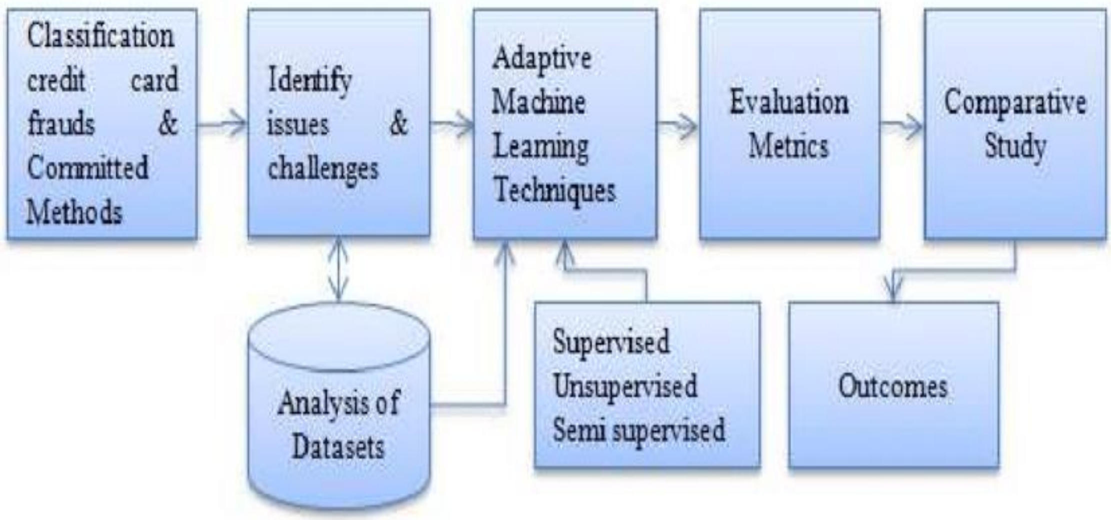

Fig 3.1.1 Proposed architecture of the system

Autoencoders are a specific type of feedforward neural networks where the input is the same as the output. They compress the input into a lower-dimensional code and then reconstruct the output from this representation. The code is a compact “summary” or “compression” of the input, also called the latent-space representation.

An autoencoder consists of 3 components: encoder, code and decoder as show in Fig 3.1.2. The encoder compresses the input and produces the code, the decoder then reconstructs the input only using this code.

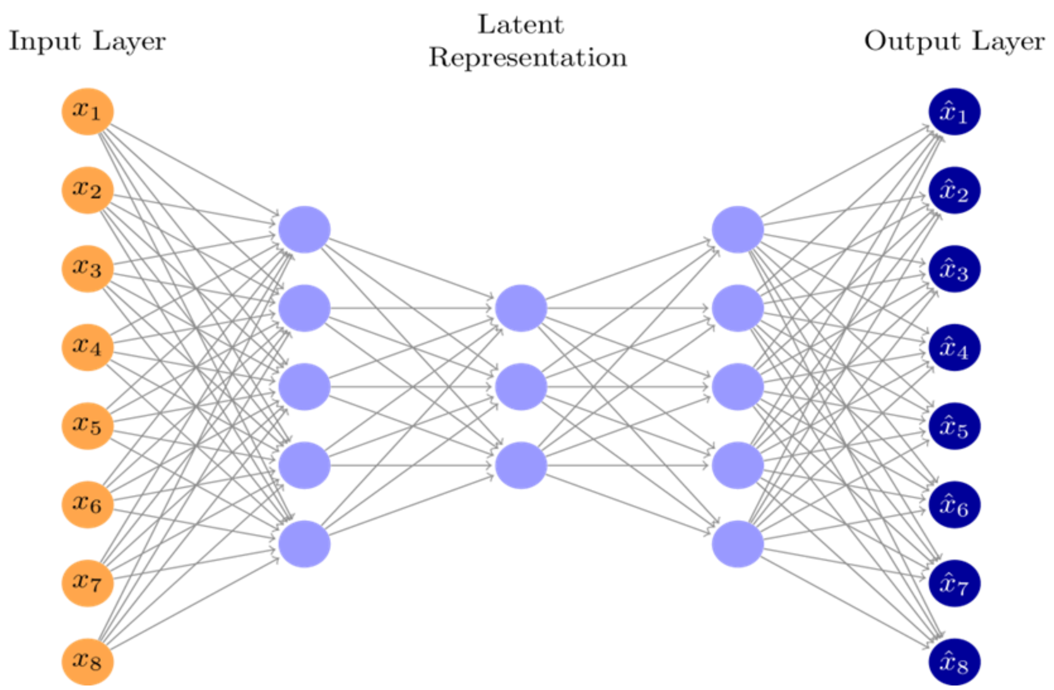

Fig 3.1.2 Diagrammatic representation of the autoencoder layers

3.2. Methodology

Essentially, a methodology is a collection of methods, practices, processes, techniques, procedures, and rules. In project management, methodologies are specific, strict, and usually contain a series of steps and activities for each phase of the project's life cycle. They are defined approaches showing us exactly what steps to take next, the motivation behind each step, and how a project stage should be performed. Methodology of the project is depicted in the Fig 3.2.1. It basically has two main steps – Balancing the data and inputting the data to the ANN model.

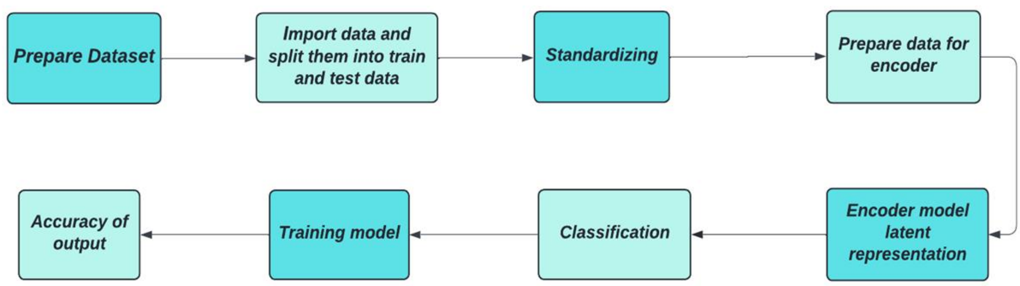

Fig 3.2.1 Methodology chart of the project

3.2.1. Balancing the imbalanced dataset

The technique we used for dealing with imbalanced data is Autoencoder. The dataset collected has a lot less positive class. In order to increase the positive (fraudulent) class we are oversampling. Oversampling comes with a lot of noise. So, we are going with dimensional reduction. We need to reduce data in such a way that we preserve the important data too. This can be achieved through the technique of auto encoder ANN which is used to extract a latent representation of the training data. Compression of the data might improve the inherent information.

- **Prepare data for encoder:** For training the Autoencoder only needs samples of the majority class because it is supposed to learn a compressed representation of those. Therefore, the training data set is separated by class and a fraction of the normal samples are used for the Autoencoder training. To improve convergence, they are standardized as well. The samples showed to the Autoencoder should not be used to train the Logistic Regression model. Therefore, a new training set for the estimator is created.
- **Autoencoder Model:** The Autoencoder consists of 5 layers: an input layer and the encoding part, the latent representation itself and the decoder part as well as an output layer. Furthermore, no optimization has been performed on this model. As we will see later, it has a nice and converging training curve and does the job.
- **Encoder Model and Latent Representation:** The encoder part of the Autoencoder consists of all the layers from the input layer up to the latent layer, which in this case are the first three layers. By predicting the compressed representation of the data one can easily extract the latent data for classification. The training history of the autoencoder is depicted in Fig. 3.2.1.1

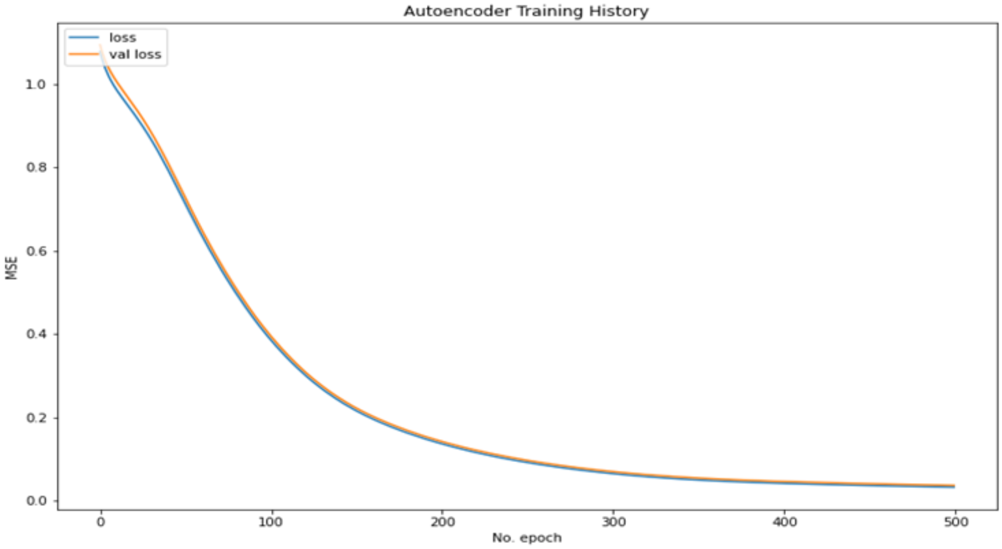

Fig 3.2.1.1 Autoencoder training history

3.2.2. Inputting the balanced data to the ANN model

The data is inputted to an ANN model. ANN model chosen has four layers. It has 50 inputs and one output. The depiction of a simple ANN model is shown in the below Fig 3.2.2.1

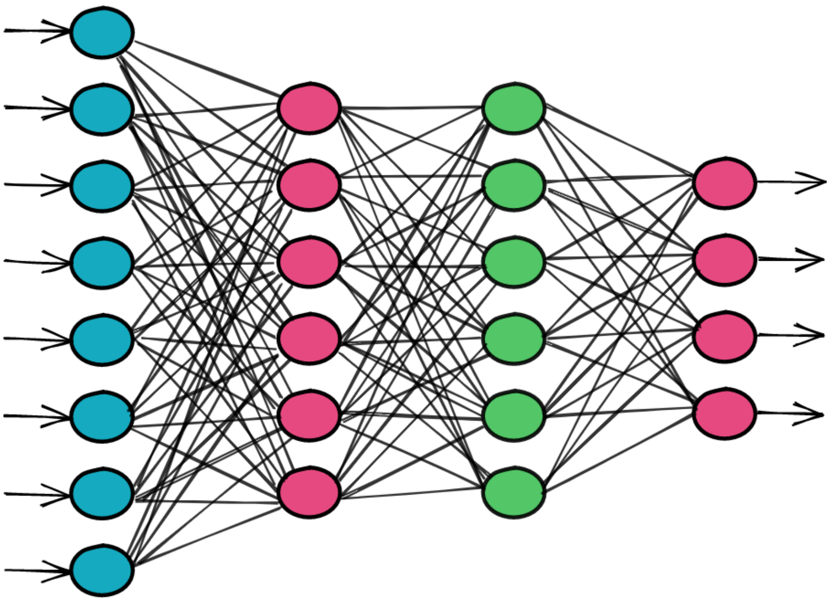

Fig 3.2.2.1. ANN model with eight inputs and 4 target variables

3.3. Dataset Details

The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

§Time – Number of seconds elapsed between the current transaction and the first transaction in the dataset

§v1 – v28 - Dimensionality reduction to protect user identities and sensitive features

§Amount – Transaction amount

§Class – 1 for fraudulent transactions, 0 otherwise

3.4. ML/DL techniques used

ANN model is used to classify the variables. The following explanation goes with the Fig 3.2.2.1.

We can see that the first layer, the input layer, consists of eight nodes. Each of the eight nodes in this layer represents an individual feature from a given sample in our dataset. This tells us that a single sample from our dataset consists of eight dimensions. When we choose a sample from our dataset and pass this sample to the model, each of the eight values contained in the sample will be provided to a corresponding node in the input layer. We can see that each of the eight input nodes are connected to every node in the next layer. Each connection between the first and second layers transfers the output from the previous node to the input of the receiving node (left to right). The two layers in the middle that have six nodes each are hidden layers simply because they are positioned between the input and output layers.

Each connection between two nodes has an associated weight, which is just a number. Each weight represents the strength of the connection between the two nodes. When the network receives an input at a given node in the input layer, this input is passed to the next node via a connection, and the input will be multiplied by the weight assigned to that connection. For each node in the second layer, a weighted sum is then computed with each of the incoming connections. This sum is then passed to an activation function, which performs some type of transformation on the given sum. For example, an activation function may transform the sum to be a number between zero and one. The actual transformation will vary depending on which activation function is used.

Once we obtain the output for a given node, the obtained output is the value that is passed as input to the nodes in the next layer. This process continues until the output layer is reached. The number of nodes in the output layer depends on the number of possible output or prediction classes we have. In our project, we have one output class.

3.5. Hardware and Software requirements

- **Processor required**: Minimum 2.0 GHz
- **Hard-Disk space required**: 80 GB of available hard disk space
- **RAM required**: Minimum 1 GB
- **Display:** 1024*768 or higher resolution display
- **Other Hardware:** DVD-ROM Drive and Software requirement is Jupyter Notebook, the platform to run the code.

3.6. Summary

The chapter summarizes the architecture of the model used, the workflow of the project, the ML/DL techniques used and the hardware and software requirements for the project.

4. Implementation details

Implementation is often used in the tech world to describe the interactions of elements in programming languages. One aspect of implementing an interface that can cause confusion is the requirement that to implement an interface, a class must implement all of the methods of that interface. This can lead to error messages due to insufficient implementation of methods.

4.1. Language/tools/ API used

The implementation of this project is entirely based on python because of its flexibility and support provided for machine learning projects. The various APIs and libraries used in the implementation of this project are:

- ***Matplotlib:*** It is one of the most powerful plotting libraries in Python. It is a cross-platform library that provides various tools to create 2D plots from the data in lists or arrays in python.
- ***Pandas:*** Made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series.
- ***Imbalance-learn*****:** Helps in balancing the datasets which are highly skewed or biased towards some classes. Thus, it helps in resampling the classes which are otherwise oversampled or under sampled.
- ***TensorFlow:*** Fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow.
- ***Scikit-learn (Sklearn):*** Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib. This library is mainly used for importing and preprocessing the database.

4.2. Use cases

The main use of this project is to enable people to get the prediction of detection of credit card frauds. Currently, the main core, i.e., the machine learning model for detection is trained. Users can access these as API through web interfaces where they enter the data and get the results back.

4.3. Workflow diagrams

Workflow is the series of activities that are necessary to complete a task. Each step in a workflow has a specific step before it and a specific step after it, with the exception of the first and last steps. In a linear workflow, an outside event usually initiates the first step.

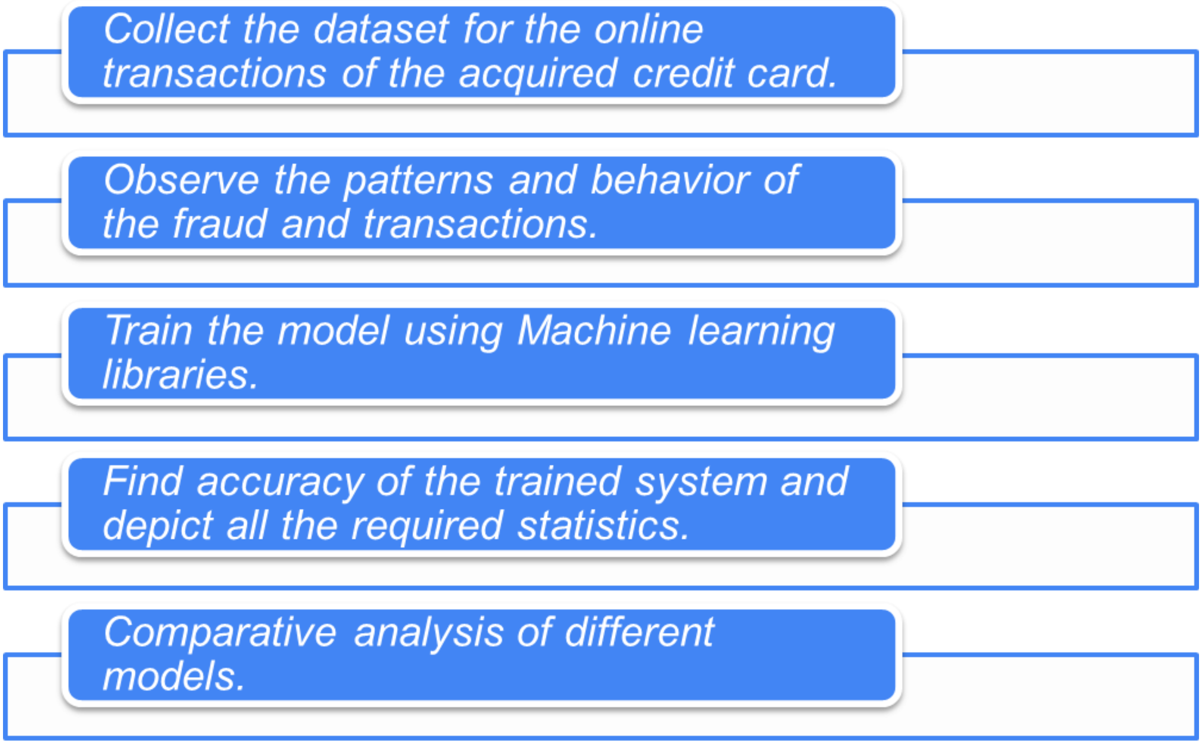

Fig 4.3.1 Workflow of the project

4.4. Data pre-processing

Data pre-processing can refer to manipulation or dropping of data before it is used in order to ensure or enhance performance, and is an important step in the data mining process. The phrase "garbage in, garbage out" is particularly applicable to data mining and machine learning projects.

- **Data pre-processing:** The Data available in the Data set is not clean. It has to be refined. A process of preparing the raw data and making it suitable for models to learn.

There are several steps like Data cleaning, Data transformation, Data reduction etc to be followed in this process.

- **Import Data and Spilt into test and train set:** For splitting we use sklearns train_test_split with the stratify option in order to keep the ratio of normal and fraudulent transactions in the test and training data equal. Importing and splitting the dataset is shown in the figures Fig 4.4.1. amd Fig 4.4.2.

```
df = pd.read_csv("../input/creditcardfraud/creditcard.csv")

assert(df.shape[0] == 284807) # make sure the data is loaded
as expected
assert(df.shape[1] == 31
```

Fig 4.4.1 Importing data into the training set

```
Training data class counts:
0    227451
1       394
Name: Class, dtype: int64

Test data class counts:
0    56864
1       98
Name: Class, dtype: int64
```

Fig 4.4.2 Splitting the data into training and test set

- **Standardizing (avoiding data leakage):** The dataset contains two features ('Amount' and 'Time') that are on a totally different scale than the rest of the features. who is the result of a Principal Component Analysis). In theory it isn't required to standardize data for Logistic Regression (e.g., here). Nonetheless, tests have shown better performance of Logistic Regression when the data has been standardized in advance and because of that the data here will be scaled. However, we did not scale the data before the split. In order to avoid information from the training data leaking into the test-set, this is done afterwards. Data Scientists far too often neglect the effects of data leakage, scaling their whole dataset before the model training. This may improve the model's performance in an unwanted manner and result in worse accuracy when dealing with data in production. Here is a simple "recipe" for standardization and is shown in the Fig 4.4.3.

  - Create a standard scaler object.

- Fit scaler on the training data and transform the data
- Transform the test data without fitting the scaler again.

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

test_data_scaled = [X_test_scaled, y_test]
```

Fig 4.4.3. Standardization of data

4.5. Validation methodology

In machine learning, there is always the need to test the stability of the model. It means based only on the training dataset; we can't fit our model on the training dataset. For this purpose, we reserve a particular sample of the dataset, which was not part of the training dataset. After that, we test our model on that sample before deployment, and this complete process comes under cross-validation. This is something different from the general train-test split. Hence the basic steps of cross-validations are:

- Reserve a subset of the dataset as a validation set.
- Provide the training to the model using the training dataset.
- Now, evaluate model performance using the validation set. If the model performs well with the validation set, perform the further step, else check for the issues. In this project, the dataset taken was already split into training and testing data, so we used the data provided as the validation data to validate the model. We even used another validation set which is a subset of the training dataset for more efficient validation.

4.6. Summary

Implementation is often used in the tech world to describe the interactions of elements in programming languages. One aspect of implementing an interface that can cause confusion is the requirement that to implement an interface, a class must implement all of the methods of that interface. This can lead to error messages due to insufficient implementation of methods.

5. Results and Analysis

This section compares the AROC values on the application of ANN on the dataset under different cases.

- First case: When the dataset is standardized, oversampled, denoised with autoencoder and given to the ANN model
- Second case: When the data set is standardized and given as an input to the ANN model
- Third case: When the dataset is oversampled and given to the ANN model

The results are tabulated in the table 5.1

Table 5.1. AROC values on the application of ANN model on the dataset under different conditions

	AROC score		
Threshold	ANN applied on autoencoder	ANN on scaled data	ANN on oversampled data
0.1	0.9506	0.908	0.5
0.2	0.9516	0.9046	0.5
0.3	0.9471	0.9012	0.5
0.4	0.9474	0.8978	0.5
0.5	0.9425	0.8944	0.5
0.6	0.9428	0.874	0.5
0.7	0.9429	0.8502	0.5
0.8	0.9432	0.8366	0.5
0.9	0.9383	0.7924	0.5

The confusion matrices at threshold equals to 0.1 under different case mentioned are shown in the figures Fig 5.1, Fig 5.2 and Fig 5.3.

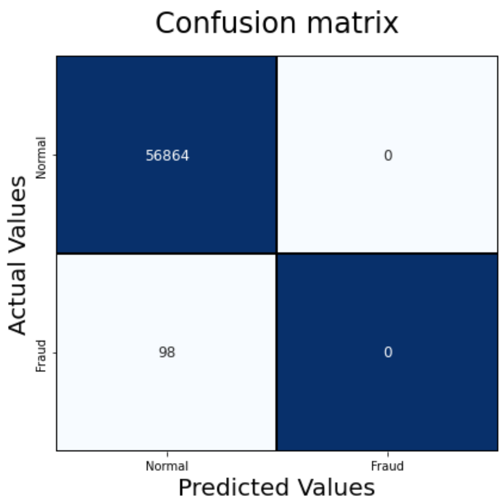

Fig 5.1. Confusion matrix when ANN is applied on oversampled data and at threshold = 0.1

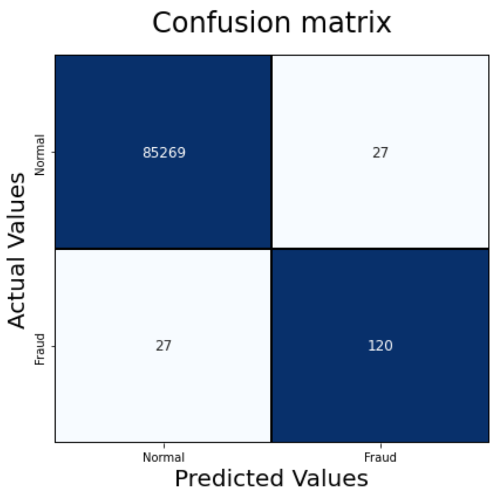

Fig 5.2. Confusion matrix when ANN is applied on standardized data and at threshold = 0.1

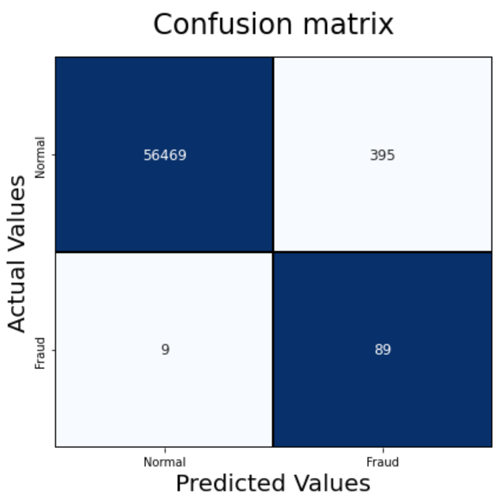

Fig 5.3. Confusion matrix when ANN is applied on the data outputted by autoencoder and at threshold = 0.1

6. Conclusions and Future Enhancements

The results obtained show that, fully connected ANN model is partially tolerant to the imbalanced data, classifying all the classes as non-fraud class (majority class), but it is of no use. When the data is scaled, the AROC score is around 90%. On using the autoencoder model the accuracy improved to around 95%.

6.1. Novelty in the proposed solution

- There is no substantial work done on ensemble models for card not present fraud detection.
- As the dataset is very imbalanced, we use an ANN algorithm (auto-encoder neural network algorithm) to balance the data by oversampling the minority class by taking care of the noise in the data.
- Now, this balanced data is fed to the ANN model under different dataset conditions to prove the best use of autoencoder
- The ensemble ANN modelling work is a new kind to improve accuracy and understandability

6.2. Limitations of the project

Machine learning algorithms work only for huge sets of data. For smaller amounts of data, the results may be not accurate. It takes a significant amount of data for machine learning models to become accurate. For large organizations, this data volume is not an issue but for others, there must be enough data points to identify legitimate cause and effect relations.

6.3. Future enhancements

While we couldn't reach our goal of 100% accuracy in fraud detection, we did end up creating a system that can, with enough time and data, get very close to that goal. As with any such project, there is some room for improvement here. The very nature of this project allows for multiple algorithms to be integrated together as modules and their results can be combined to increase the accuracy of the final result. This model can further be improved with the addition of more algorithms into it. However, the output of these algorithms needs to be in the same format as the others. Once that condition is satisfied, the modules are easy to add as done in the code. This provides a great degree of modularity and versatility to the project. More room for improvement can be found in the dataset. As demonstrated before, the precision of the algorithms increases when the size of the dataset is increased. Hence, more data will surely make the model more accurate in detecting frauds and reduce the number of false positives. However, this requires official support from the banks themselves. Another facility can also be provided where the machine learning algorithms can be implemented with websites using Flask to produce a system where input card fields are given as input and the system determines if the card used is fraud or not.

7. References

[1] Varun Kumar K S , Vijaya Kumar V G , Vijayshankar A , Pratibha K, 2020, Credit Card Fraud Detection using Machine Learning Algorithms, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 09, Issue 07 (July 2020),

[2] Dornadula, Vaishnavi Nath and Subbiah Geetha. "Credit Card Fraud Detection using Machine Learning Algorithms." Procedia Computer Science (2019): n. pag

[3] Benchaji, Ibtissam & Douzi, Samira & Ouahidi, Bouabid & Jaafari, Jaafar. (2021). Enhanced credit card fraud detection based on attention mechanism and LSTM deep model. Journal of Big Data. 8. 10.1186/s40537-021-00541-8.

[4] Mekterovi´c, I.; Karan, M.; Pintar, D.; Brki´c, L. Credit Card Fraud Detection in Card-Not-Present Transactions: Where to Invest? Appl. Sci. 2021, 11, 6766. https://doi.org/10.3390/app11156766

[5] Kartik Madkaikar, Manthan Nagvekar, Preity Parab, Riya Raikar, Supriya Patil, Credit Card Fraud Detection System ISSN:2277-3878, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT), Volume – 10 Issue – 2, July 2021

[6] Fayyomi, Aisha & Eleyan, Derar & Eleyan, Amina. (2021). A Survey Paper On Credit Card Fraud Detection Techniques. International Journal of Scientific & Technology Research. 10. 72-79.

[7] RB, Asha & K R, Suresh. (2021). Credit Card Fraud Detection Using Artificial Neural Network. Global Transitions Proceedings. 2. 10.1016/j.gltp.2021.01.006.

[8] Seera, Manjeevan & Lim, Chee & Kumar, Ajay & Dhamotharan, Lalitha & Tan, Kim. (2021). An intelligent payment card fraud detection system. Annals of Operations Research. 10.1007/s10479-021-04149-2.

[9] Pundir, Amit & Pandey, Rajesh. (2021). Data Quality Analysis based Machine Learning models for Credit Card Fraud Detection. Journal of University of Shanghai for Science and Technology. 23. 318-344. 10.51201/JUSST/21/05263.

[10] Zou, Junyi & Zhang, Jinliang & Jiang, Ping. (2019). Credit Card Fraud Detection Using Autoencoder Neural Network.

[11] ACFE. Report to the nations 2018 global study on occupational fraud and abuse. 2019. https://doi.org/10.1002/ 9781118929773.oth1.

[12] Zafar A, Sirshar M. A survey on application of Data Mining techniques; it's profciency in fraud detection of credit card. Res Rev J Eng Technol. 2018;7:15–23

[13] Mohammed E, Far B. Supervised machine learning algorithms for credit card fraudulent transaction detection: a comparative study. In: IEEE annals of the history of computing. IEEE; 2018. https://doi.org/10.1109/IRI.2018. 00025

[14] Carcillo F, Le Borgne Y-A, Caelen O, et al. Combining unsupervised and supervised learning in credit card fraud detection. Inf Sci. 2019.

[15] Forough J, Momtazi S. Ensemble of deep sequential models for credit card fraud detection. Appl Soft Comput J. 2020. https://doi.org/10.1016/j.asoc.2020.106883.

Appendix

Confusion matrices when the dataset is oversampled and given to ANN model:

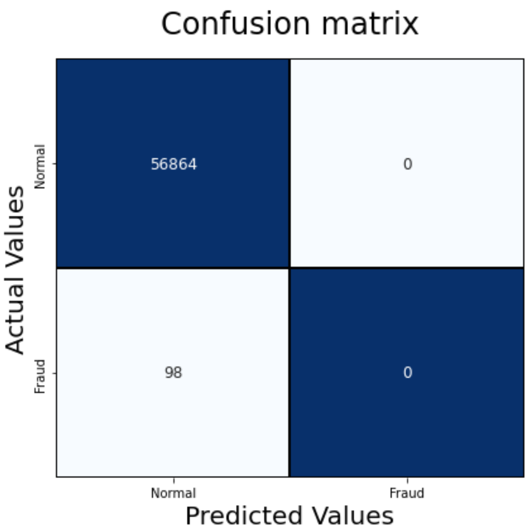

Fig A1: Classified 0 out of 98 fraud cases correctly
Misclassified 0 out of 56864 normal cases
AROC score: 0.5
Threshold = 0.1

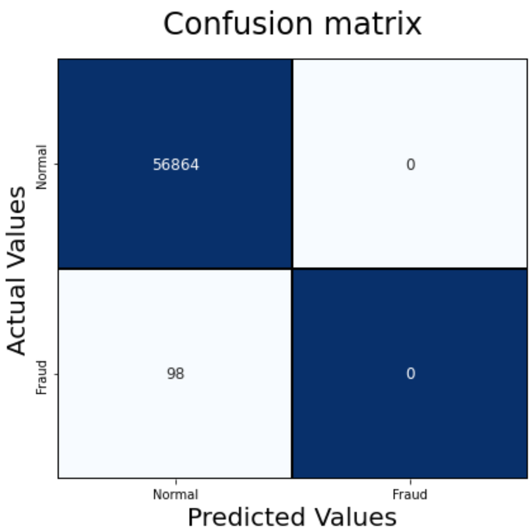

Fig A2: Classified 0 out of 98 fraud cases correctly
Misclassified 0 out of 56864 normal cases
AROC score: 0.5
Threshold = 0.9

Same values are obtained at other thresholds 0.2, 0.3, 0.4, 0.5, 0.6, 0.7 and 0.8

Confusion matrices when the dataset is standardized and given to ANN model:

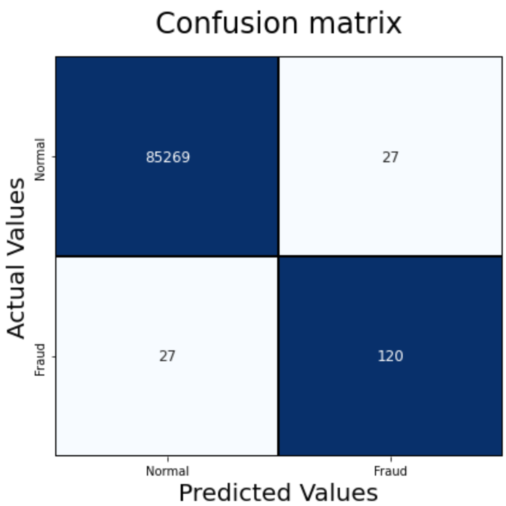

Fig A3: Classified 120 out of 147 fraud cases correctly
Misclassified 27 out of 85296 normal cases
AROC score: 0.9080

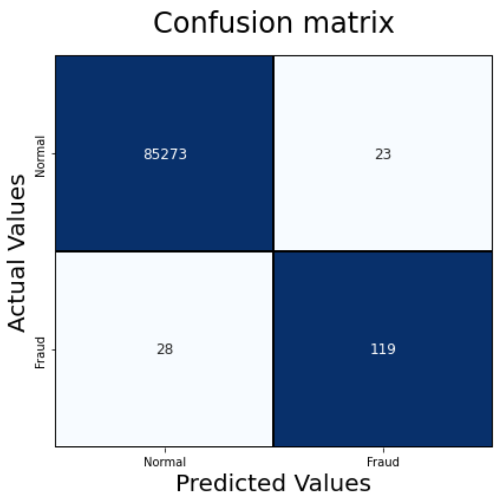

Fig A4: Classified 119 out of 147 fraud cases correctly
Misclassified 23 out of 85296 normal cases
AROC score: 0.9046

Threshold = 0.1

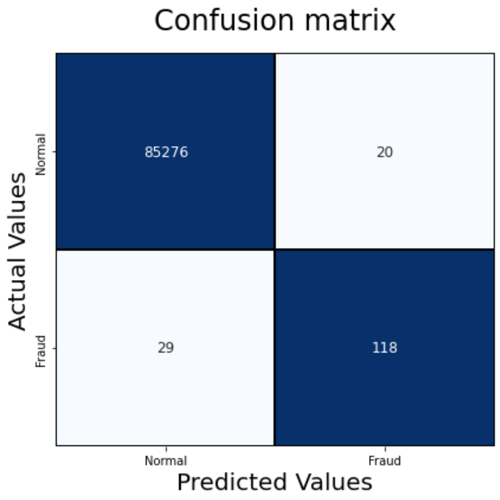

Fig A5: Classified 118 out of 147 fraud cases correctly
Misclassified 20 out of 85296 normal cases
AROC score: 0.9012

Threshold = 0.2

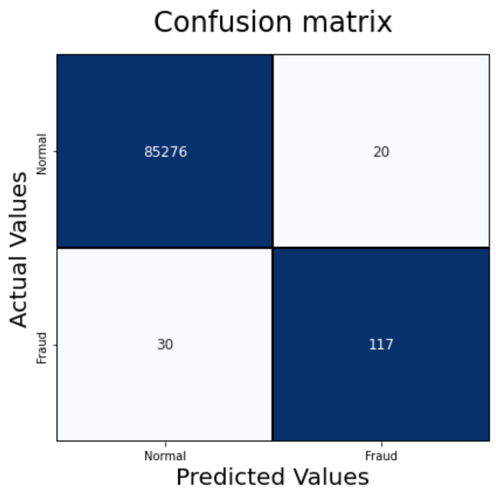

Fig A6: Classified 117 out of 147 fraud cases correctly
Misclassified 20 out of 85296 normal cases
AROC score: 0.8978

Threshold = 0.3

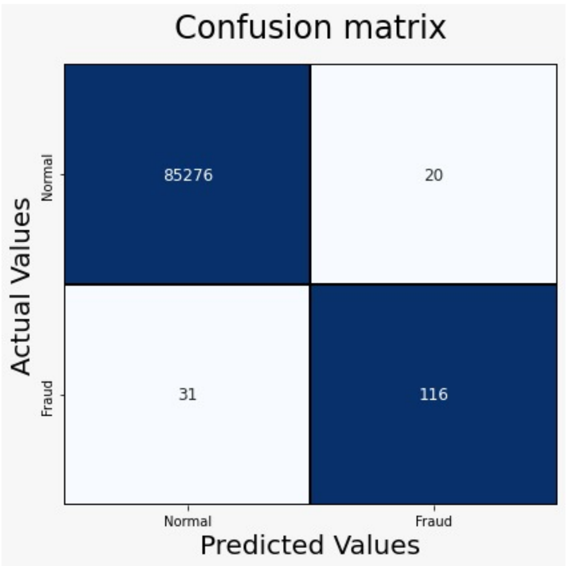

Fig A7: Classified 116 out of 147 fraud cases correctly
Misclassified 20 out of 85296 normal cases
AROC score: 0.8944

Threshold = 0.4

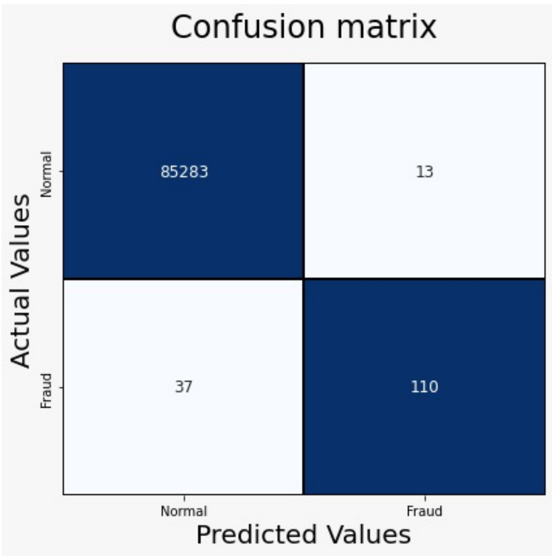

Fig A8: Classified 110 out of 147 fraud cases correctly
Misclassified 13 out of 85296 normal cases
AROC score: 0.8740

Threshold = 0.5

Threshold = 0.6

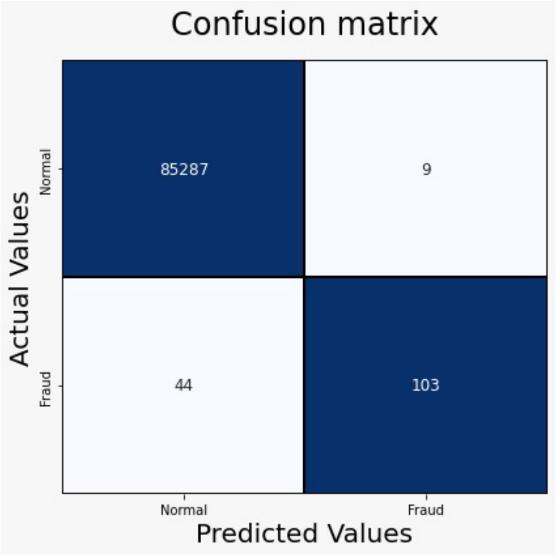

Fig A9: Classified 103 out of 147 fraud cases correctly
Misclassified 9 out of 85296 normal cases
AROC score: 0.8507
Threshold = 0.7

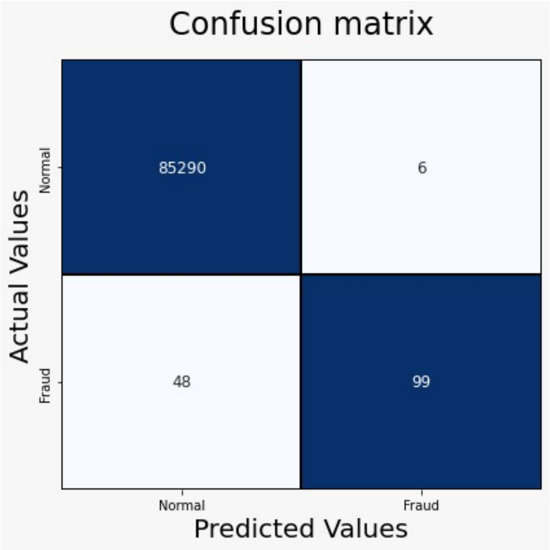

Fig A10: Classified 99 out of 147 fraud cases correctly
Misclassified 6 out of 85296 normal cases
AROC score: 0.8366
Threshold = 0.8

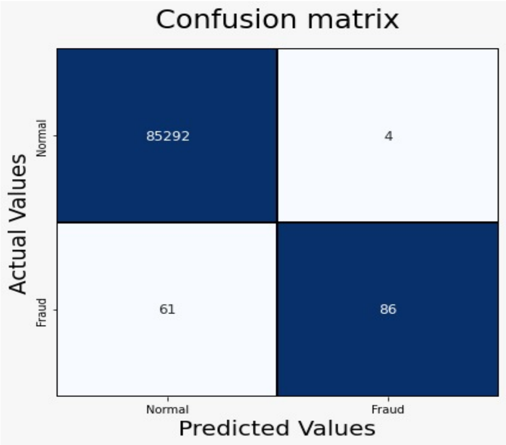

Fig A11: Classified 86 out of 147 fraud cases correctly
Misclassified 4 out of 85296 normal cases
AROC score: 0.7924
Threshold = 0.9

Confusion matrices when the dataset is outputted by the autoencoder:

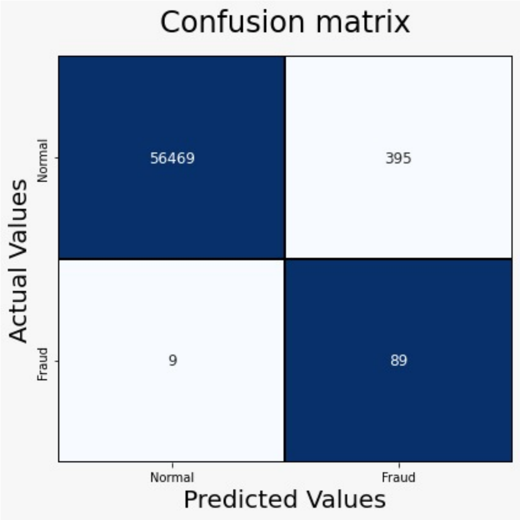

Fig A12: Classified 89 out of 98 fraud cases correctly
Misclassified 395 out of 56864 normal cases
AROC score: 0.9506
Threshold = 0.1

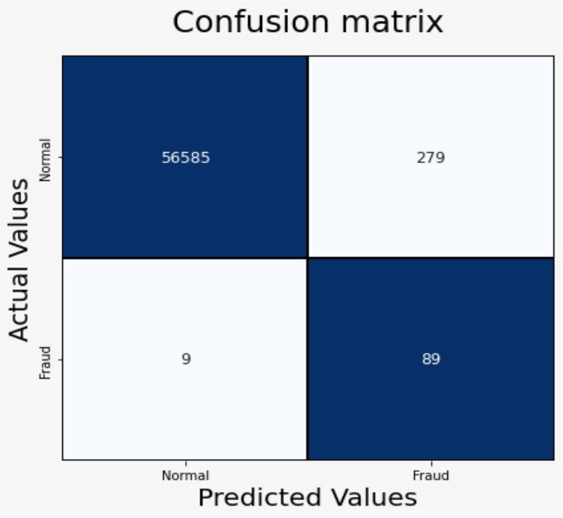

Fig A13: Classified 89 out of 98 fraud cases correctly
Misclassified 279 out of 56864 normal cases
AROC score: 0.9516
Threshold = 0.2

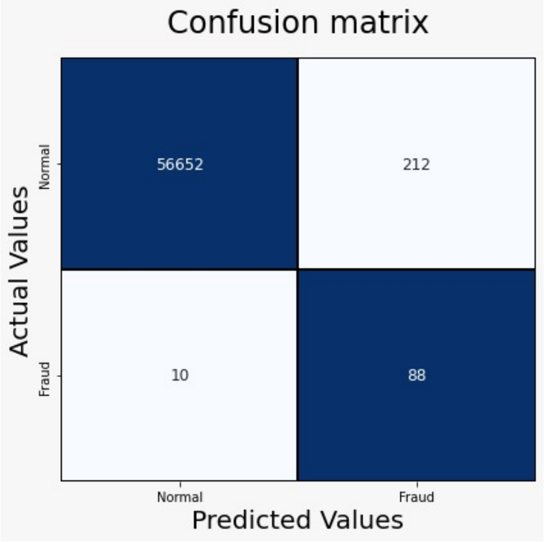

Fig A14: Classified 88 out of 98 fraud cases correctly
Misclassified 212 out of 56864 normal cases
AROC score: 0.9471
Threshold = 0.3

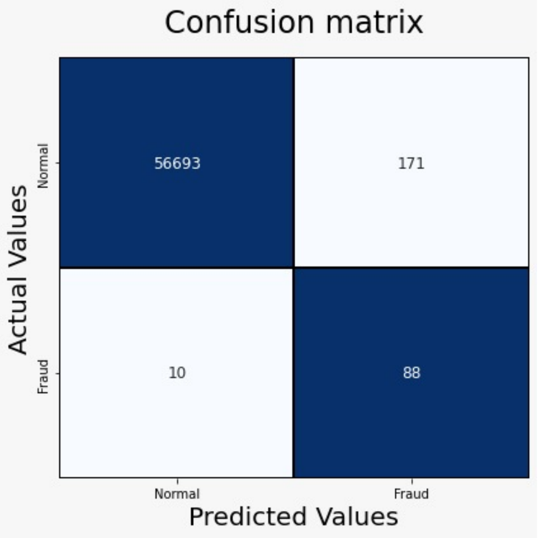

Fig A15: Classified 88 out of 98 fraud cases correctly
Misclassified 171 out of 56864 normal cases
AROC score: 0.9474
Threshold = 0.4

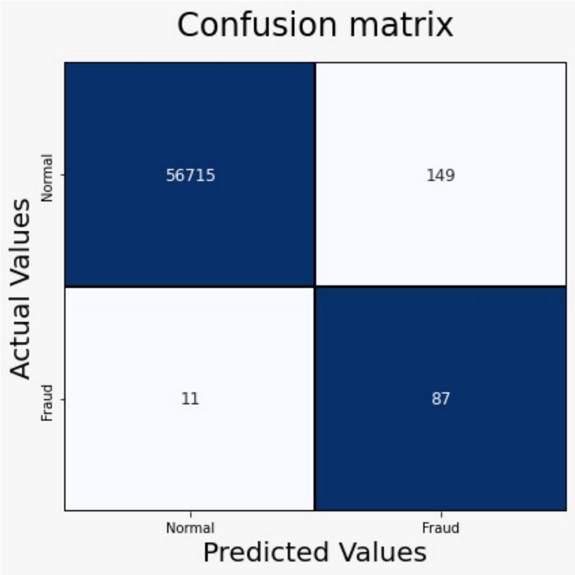

Fig A16: Classified 87 out of 98 fraud cases correctly
Misclassified 149 out of 56864 normal cases
AROC score: 0.9425
Threshold = 0.5

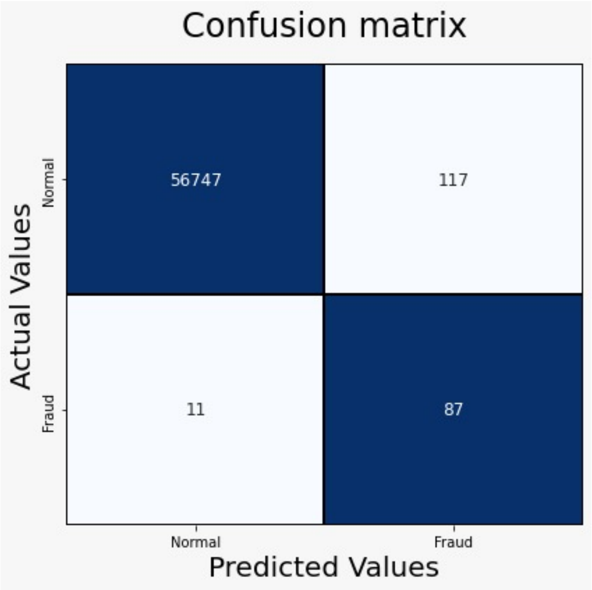

Fig A17: Classified 87 out of 98 fraud cases correctly
Misclassified 117 out of 56864 normal cases
AROC score: 0.9428
Threshold = 0.6

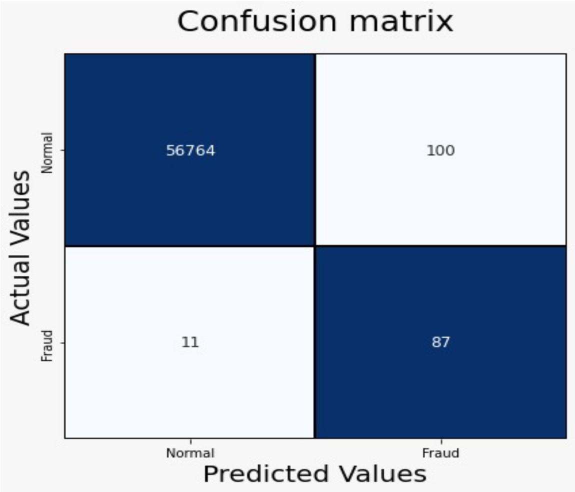

Fig A18: Classified 87 out of 98 fraud cases correctly
Misclassified 100 out of 56864 normal cases
AROC score: 0.9429
Threshold = 0.7

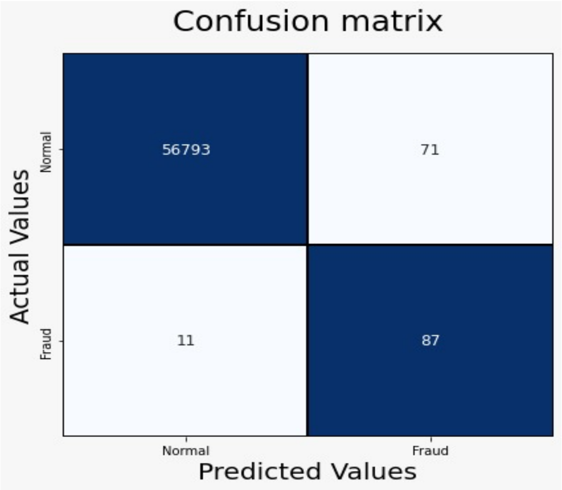

Fig A19: Classified 87 out of 98 fraud cases correctly
Misclassified 71 out of 56864 normal cases
AROC score: 0.9432
Threshold = 0.8

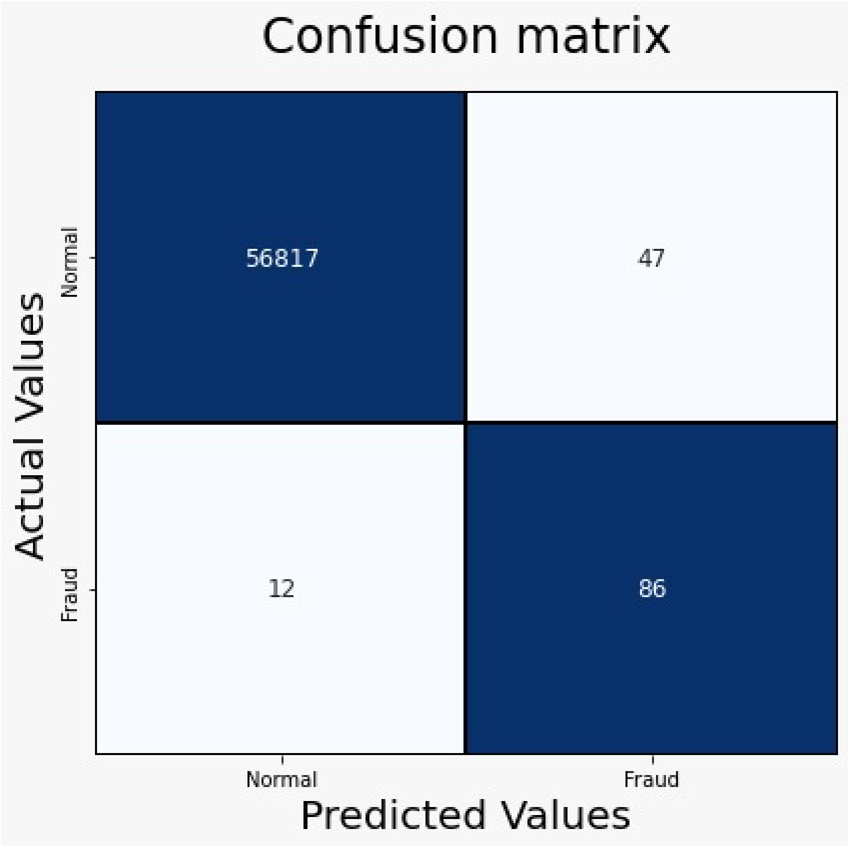

Fig A20: Classified 86 out of 98 fraud cases correctly
Misclassified 47 out of 56864 normal cases
AROC score: 0.9383
Threshold = 0.9