REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

DEEP LEARNING-BASED AND COST-AWARE FRAUD DETECTION SYSTEM USING MULTI-MODAL PROFILING APPROACH TO DETECT FRAUD ATTEMPTS IN AIRLINE TICKET SALES

Mehmed Taha ARAS

DOCTOR OF PHILOSOPHY THESIS
Department of Computer Engineering
Program of Computer Engineering

Supervisor
Assoc. Prof. Mehmet Amaç GÜVENSAN

March, 2024

REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

DEEP LEARNING-BASED AND COST-AWARE FRAUD DETECTION SYSTEM USING MULTI-MODAL PROFILING APPROACH TO DETECT FRAUD ATTEMPTS IN AIRLINE TICKET SALES

A thesis submitted by Mehmed Taha ARAS in partial fulfillment of the requirements for the degree of **DOCTOR OF PHILOSOPHY** is approved by the committee on 12.03.2024 in Department of Computer Engineering, Program of Computer Engineering.

Assoc. Prof. Mehmet Amaç GÜVENSAN
Yildiz Technical University
Supervisor

Approved By the Examining Committee

Assoc. Prof. Mehmet Amaç GÜVENSAN, Supervisor
Yildiz Technical University _____

Professor Mehmet Fatih AMASYALI, Member
Yildiz Technical University _____

Asst. Prof. Hafiza İrem TÜRKMEN ÇİLİNGİR, Member
Yildiz Technical University _____

Assoc. Prof. Gökhan İNCE, Member
Istanbul Technical University _____

Professor Ali Gökhan YAVUZ, Member
Turkish-German University _____

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of Deep Learning-Based and Cost-Aware Fraud Detection System Using Multi-Modal Profiling Approach to Detect Fraud Attempts in Airline Ticket Sales supervised by my supervisor, Assoc. Prof. Mehmet Amaç GÜVENSAN. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Mehmed Taha ARAS

Signature

Dedicated to my wife
and my family

ACKNOWLEDGEMENTS

I would like to express my gratitude to my precious wife, Beyza, who supported and encouraged me in my thesis studies and all the difficult processes.

I would like to thank my father and my dear family for providing all kinds of support throughout my doctorate process.

I would like to express my endless thanks to my advisor, Assoc. Prof. Mehmet Amaç GÜVENSAN, who guided me in detail in my studies, made suggestions and enabled me to produce a valuable thesis.

I would like to thank Professor Mehmet Fatih AMASYALI and Assoc. Prof. Gökhan İNCE, who was on the thesis monitoring committee, guided me in my research with their valuable suggestions.

We sincerely thank Turkish Airlines company for sharing their fraud data and enabling us to produce a valuable scientific thesis study.

Mehmed Taha ARAS

TABLE OF CONTENTS

LIST OF SYMBOLS viii

LIST OF ABBREVIATIONS ix

LIST OF FIGURES xi

LIST OF TABLES xiii

ABSTRACT xvi

ÖZET xviii

1 INTRODUCTION 1

1.1 Literature Review 4

1.2 Fraud Domains 8

1.2.1 Banking 10

1.2.2 E-Commerce 10

1.2.3 Insurance 11

1.2.4 Aviation 11

1.3 Objective of the Thesis 15

1.4 Hypothesis 15

2 DATASET 17

3 CHALLENGES 22

3.1 Imbalanced Dataset 22

3.2 Smart Instance Selection 23

3.3 Cost-Sensitive Measurement 23

3.4 Fraud Detection on Transactional Information 24

4 DATA PREPARATION 25

4.1 Data Pre-Processing 25

4.1.1 Data Cleaning 25

4.1.2 Data Transformation 29

4.2 Train and Test Set Distribution 36

4.2.1 Random Distribution 36
4.2.2 Chronological Distribution 36
4.2.3 Monthly Sequential Distribution 37
4.2.4 Comparison of Random, Chronological, and Monthly Sequential Distribution 37
4.2.5 Monthly Cumulative Distribution 38
4.3 Sampling 40
4.3.1 Balanced Random Undersampling (BRUS) 40
4.3.2 Oversampling 42
4.3.3 Hybrid BRUS + Oversampling 42
4.4 Feature Selection 45

5 COST-BASED MEASUREMENT SYSTEM 49
5.1 Impact Analysis of Transaction Amount on Fraud Risk 50
5.2 Currency Standardization 53
5.3 Amount Normalization 53
5.4 Cost-Based Confusion Matrix 53
5.5 Quantity and Cost-Based Success Comparison 53

6 PROFILE-BASED FRAUD DETECTION 55
6.1 Profile-Based Transformation 57
6.2 Profiles 57
6.2.1 Credit Card Profiling 57
6.2.2 PNR Profiling 59
6.2.3 Record Profiling 61
6.3 Profiling Modes 63
6.3.1 Credit Card Profiling Mode 64
6.3.2 PNR Profiling Mode 64
6.3.3 Record Profiling Mode 64
6.3.4 Only Profiling Mode 65
6.3.5 Multi-Modal Profiling 65
6.4 Architecture of the Developed Fraud Detection System 66

7 RESULTS AND DISCUSSION 69
7.1 Performance Improvements on Transactional Dataset 70
7.2 Profile-Based Fraud Detection Results 73
7.3 Ensemble Learning Results on Profiling Approach 83
7.4 Fraud Detection Characteristics of the Developed System 84
7.5 Time Performances of the Developed System 86
7.6 High Scale Fraud/Legal Ratio Experiments 88

8 CONCLUSION **90**

REFERENCES **92**

A EMBEDDING LAYER EXPERIMENTS **98**

B TRAIN TEST DISTRIBUTION EXPERIMENTS **99**

C SAMPLING EXPERIMENTS **105**

D FEATURE SELECTION EXPERIMENTS **108**

E PROFILING EXPERIMENTS **114**

F DEEP LEARNING EXPERIMENTS **126**

G ENSEMBLE LEARNING EXPERIMENTS **129**

PUBLICATIONS FROM THE THESIS **133**

LIST OF SYMBOLS

c	USD Amount
cScore	Normalized Cost

LIST OF ABBREVIATIONS

ADASYN	Adaptive Synthetic Sampling
AE	Auto Encoder
BRUS	Balanced Random Undersampling
BKM	Interbank Card Center
CHI^2	Chi-Squared Test
CNN	Convolutional Neural Network
C-Accuracy	Cost-Based Accuracy
C-F1 Score	Cost-Based F1 Score
C-Precision	Cost-Based Precision
C-Recall	Cost-Based Recall
DBSCAN	Density-Based Spatial Clustering Of Applications With Noise
DNN	Deep Neural Network
DT	Decision Tree
ERP	Enterprise Resource Planning
EXP	Experiment
FN	False Negative
FP	False Positive
GMT	Greenwich Mean Time Zone
K-NN	K-Nearest Neighbors
LSTM	Long Short-Term Memory
MAX	Maximum
ML	Machine Learning
MLP	Multi-Layer Perceptron

NB	Naive Bayes
N/A	Not Available
OWL	Web Ontology Language
PCA	Principal Component Analysis
PNR	Passenger Name Record
POS	Point Of Sale
RF	Random Forest
SMOTE	Synthetic Minority Over-sampling Technique
SMOTENC	SMOTE-Nominal Continuous
SOM	Self-Organizing Map
SVM	Support Vector Machine
SWRL	SemanticWeb Rule Language
THY	Turkish Airlines
TN	True Negative
TP	True Positive
XG-BOOST	Extreme Gradient Boosting

LIST OF FIGURES

Figure 1.1 Fraud types according to their domains 9
Figure 1.2 Fraud Rates in Aviation 15
Figure 4.1 Single-Layer Embedding Process 33
Figure 4.2 Multi-Layer Embedding Process 34
Figure 4.3 Comparison of embedding layer and one-hot-encoding success results 35
Figure 4.4 Comparison of embedding layer and one-hot-encoding time performances 36
Figure 4.5 Comparison of 1 and 2-Month Test Sets in Terms of F1 Scores . 39
Figure 4.6 Comparison of 1 and 2-Month Test Sets in Terms of Saturation Points 40
Figure 4.7 BRUS Algorithm 42
Figure 4.8 Comparison of BRUS and random oversampling with different fraud/legal ratios in terms of F1 Score 45
Figure 4.9 Comparison of BRUS and SMOTE with different fraud/legal ratios in terms of F1 Score 45
Figure 4.10 Comparison of BRUS and SMOTENC with different fraud/legal ratios in terms of F1 Score 46
Figure 4.11 Comparison of feature selection methods in terms of quantity-based F1 Score 47
Figure 4.12 Comparison of feature selection methods in terms of cost-based F1 Score 48
Figure 5.1 Fraud Percentages in Dollar Fee Ranges 51
Figure 5.2 Ratio of Fraud Transactions in Dollar Fee Ranges to Fraud Transactions in the Whole Dataset 51
Figure 5.3 Fraud Percentages in Normalized Fee Ranges 52
Figure 5.4 Ratio of Fraud Transactions in Normalized Fee Ranges to Fraud Transactions in the Whole Dataset 52
Figure 6.1 Profiling Algorithm 57
Figure 6.2 F1 Score Gains of Profile-Based Fraud Detection System Compared to Transactional Approach in Terms of Quantity . . . 65

Figure 6.3 F1 Score Gains of Profile-Based Fraud Detection System Compared to Transactional Approach in Terms of Cost 66
Figure 6.4 Architecture of the Developed Fraud Detection System 67
Figure 6.5 Architecture and configuration of the developed DNN 68
Figure 7.1 The initial fraud-detection performance of state-of-the-art learning algorithms and the impact of the feature-selection process in terms of quantity 72
Figure 7.2 The initial fraud-detection performance of state-of-the-art learning algorithms and the impact of the feature-selection process in terms of cost 73
Figure 7.3 Balanced random undersampling (BRUS) vs. BRUS + random oversampling (ROS) in terms of F1 Score 74
Figure 7.4 Balanced random undersampling (BRUS) vs. BRUS + SMOTENC in terms of F1 Score 74
Figure 7.5 The contribution of the proposed profiling modes for each state-of-the-art algorithm on a quantity basis. 77
Figure 7.6 The impact of feature-selection algorithms on the performance of initial DNN utilizing multi-modal profiling 78
Figure 7.7 The contribution of the proposed profiling modes for each state-of-the-art algorithm on a cost basis. 79
Figure 7.8 The impact of CHI^2 on the state-of-the-art algorithms utilizing multi-modal profiling 79
Figure 7.9 The impact of calibration of the number of epochs to the system success 80
Figure 7.10 The impact of calibration of the batch size to the system success 80
Figure 7.11 The impact of calibration of the learning rate to the system success 80
Figure 7.12 The impact of dropout on the DNN hidden layers on the system success 81
Figure 7.13 The impact of DNN layer formation to the system success . . . 82
Figure 7.14 Ordered F1 Score results of hyperparameter tuning experiments 82
Figure 7.15 Comparison of Ensemble Learning and Machine Learning Techniques 85
Figure 7.16 Higher scale fraud/legal ratio tests (quantity based) 88
Figure 7.17 Higher scale fraud/legal ratio tests (cost based) 89

LIST OF TABLES

Table 1.1 Available features in our dataset supported by a worldwide airline company. (P&D: Profiled and Discarded) 13
Table 2.1 Fraud Distributions by Category in Significant Data Fields 18
Table 4.1 Corrupted Row Recovering Examples 26
Table 4.2 Number of Features Formed with One-Hot Vectors According to Fraud / NonFraud Balance 31
Table 4.3 Performance Comparison of Train/Test Set Distribution Methods 38
Table 5.1 Quantity and Cost Based Transactional Results 54
Table 6.1 Two example credit card profiling scenarios to emphasize the difference in fraud detection capability between transactional and profile-based approaches. 59
Table 6.2 example PNR profiling scenarios to emphasize the difference in fraud detection capability between transactional and profile-based approaches. 61
Table 6.3 Nine independent record profiling scenarios to emphasize the difference in fraud detection capability between transactional and profile-based approaches. 63
Table 7.1 The details of our dataset and train/test setup. 71
Table 7.2 Transactional approach vs. profiled approach 83
Table 7.3 Time performances per transaction of the preprocessing stages of the developed fraud detection system in milliseconds 87
Table 7.4 Time performances per transaction of the classification stages of the developed fraud detection system in milliseconds 87
Table A.1 Embedding layer test results without feature selection (ML = MultiLayer , SL = SingleLayer, X = Embedding size per categorical field) 98
Table A.2 Embedding layer test results without feature selection (ML = MultiLayer , SL = SingleLayer, X = Embedding size per categorical field) 98
Table B.1 Random Distribution Results According to Fraud/Legitimate Balances 99

Table B.2 Chronological Distribution Results According to Fraud/Legitimate Balances 100
Table B.3 Monthly Sequential Distribution Results 100
Table B.4 Decision Tree - Cumulative Distribution Results (1 Month Test) . 101
Table B.5 Random Forest - Cumulative Distribution Results (1 Month Test) 101
Table B.6 SVM - Cumulative Distribution Results (1 Month Test) 101
Table B.7 MLP - Cumulative Distribution Results (1 Month Test) 102
Table B.8 Naive Bayes - Cumulative Distribution Results (1 Month Test) . . 102
Table B.9 Decision Tree - Cumulative Distribution Results (2 Month Test) . 102
Table B.10 Random Forest - Cumulative Distribution Results (2 Month Test) 103
Table B.11 SVM - Cumulative Distribution Results (2 Month Test) 103
Table B.12 MLP - Cumulative Distribution Results (2 Month Test) 103
Table B.13 Naive Bayes - Cumulative Distribution Results (2 Month Test) . . 104
Table C.1 Hybrid BRUS + Random Oversampling Results 105
Table C.2 Hybrid BRUS + SMOTE Results 106
Table C.3 Hybrid BRUS + SMOTENC Results 107
Table D.1 Decision Tree - CHI2 Feature Selection Results 108
Table D.2 Random Forest - CHI2 Feature Selection Results 108
Table D.3 SVM - CHI2 Feature Selection Results 109
Table D.4 MLP - CHI2 Feature Selection Results 109
Table D.5 Naive Bayes - CHI2 Feature Selection Results 109
Table D.6 Decision Tree - PCA Feature Selection Results 110
Table D.7 Random Forest - PCA Feature Selection Results 110
Table D.8 SVM - PCA Feature Selection Results 110
Table D.9 MLP - PCA Feature Selection Results 111
Table D.10 Naive Bayes - PCA Feature Selection Results 111
Table D.11 Decision Tree - AutoEncoder Feature Selection Results 111
Table D.12 Random Forest - AutoEncoder Feature Selection Results 112
Table D.13 SVM - AutoEncoder Feature Selection Results 112
Table D.14 MLP - AutoEncoder Feature Selection Results 112
Table D.15 Naive Bayes - AutoEncoder Feature Selection Results 113
Table E.1 Credit Card Profiling Test Results (Decision Tree) 114
Table E.2 Credit Card Profiling Test Results (Random Forest) 115
Table E.3 Credit Card Profiling Test Results (SVM) 115
Table E.4 Credit Card Profiling Test Results (MLP) 116
Table E.5 Credit Card Profiling Test Results (Naive Bayes) 116
Table E.6 PNR Profiling Test Results (Decision Tree) 117
Table E.7 PNR Profiling Test Results (Random Forest) 117
Table E.8 PNR Profiling Test Results (SVM) 118

Table E.9 PNR Profiling Test Results (MLP) 118
Table E.10 PNR Profiling Test Results (Naive Bayes) 119
Table E.11 Record Profiling Test Results (Decision Tree) 119
Table E.12 Record Profiling Test Results (Random Forest) 120
Table E.13 Record Profiling Test Results (SVM) 120
Table E.14 Record Profiling Test Results (MLP) 121
Table E.15 Record Profiling Test Results (Naive Bayes) 121
Table E.16 Only Profiling Mode Test Results (Decision Tree) 122
Table E.17 Only Profiling Mode Test Results (Random Forest) 122
Table E.18 Only Profiling Mode Test Results (SVM) 122
Table E.19 Only Profiling Mode Test Results (MLP) 122
Table E.20 Only Profiling Mode Test Results (Naive Bayes) 122
Table E.21 Multi-Modal Profiling Test Results (Decision Tree) 123
Table E.22 Multi-Modal Profiling Test Results (Random Forest) 123
Table E.23 Multi-Modal Profiling Test Results (SVM) 124
Table E.24 Multi-Modal Profiling Test Results (MLP) 124
Table E.25 Multi-Modal Profiling Test Results (Naive Bayes) 125
Table F.1 Experimental setup for tuning hyperparameters and network structure of DNN 126
Table F.1 *Experimental setup for tuning hyperparameters and network structure of DNN Cont.* 127
Table F.2 The experimental results of fine-tuning test setup for DNN 128
Table G.1 Ensemble Learning Results (Random Forest, SVM, Naive Bayes) 129
Table G.2 Ensemble Learning Results (Random Forest , MLP , Naive Bayes) 130
Table G.3 Ensemble Learning Results (Decision Tree , SVM , Naive Bayes) 130
Table G.4 Ensemble Learning Results (Decision Tree , MLP , Naive Bayes) 131
Table G.5 Ensemble Learning Results (SVM , MLP , Naive Bayes) 131
Table G.6 F1 Score Comparison of Machine Learning and Ensemble Learning (Quantity Based) 132
Table G.7 F1 Score Comparison of Machine Learning and Ensemble Learning (Cost Based) 132

ABSTRACT

Deep Learning-Based and Cost-Aware Fraud Detection System Using Multi-Modal Profiling Approach to Detect Fraud Attempts in Airline Ticket Sales

Mehmed Taha ARAS

Department of Computer Engineering
Doctor of Philosophy Thesis

Supervisor: Assoc. Prof. Mehmet Amaç GÜVENSAN

Today, in most systems where product and service sales occur digitally, cases of fraud attempts with stolen credit cards are rapidly becoming widespread. Although the datasets most commonly used in fraud detection systems in the literature belong to the banking sector, the aviation industry is one of the sectors most affected by fraud cases in terms of financial and customer satisfaction.

In this thesis study, a fraud detection outline has been created for the first time in the literature with state-of-the-art machine learning algorithms in the aviation field. Later, in order to avoid the correlation problem caused by a transactional dataset consisting of unconnected transactions, a new, multi-modal profiling mechanism based on deep learning was presented. This multi-modal profiling mechanism includes three different profiles that can contribute to fraud detection: credit card profile, PNR (Passenger Name Record) profile, and record profile.

The Balanced Random Undersampling (BRUS) algorithm was developed in this new system to eliminate the imbalanced dataset problem. Utilizing this algorithm, a mechanism that allows a temporally balanced selection among the majority class records from the dataset and prevents duplication in record selection has been revealed. As a result of using the BRUS algorithm together with the profiling mechanism, the information of the unselected records was preserved thanks to the profiled fields, despite undersampling.

Finally, a cost-based metric system and a cost-based measurement mechanism have been suggested so that the developed system can measure both the number of transactions and the transaction cost. Due to this cost-based measurement mechanism, both the financial success and the success of the fraud detection system in terms of customer satisfaction can be observed. A significant success was achieved by feeding the newly created profile-based dataset into a fine-tuned deep neural network by adjusting well-known hyperparameters related to aviation data.

Keywords: Fraud detection in aviation, profile-based systems, deep learning, cost-sensitive measurement, history-aware detection

ÖZET

Havayolu Bilet Satış İşlemleri İçin Derin Öğrenme Tabanlı Maliyet Bilinçli ve Çok Modlu Profilleme Yaklaşımını Benimseyen Sahtekarlık Tespit Sistemi

Mehmed Taha ARAS

Bilgisayar Mühendisliği Anabilim Dalı
Doktora Tezi

Danışman: Doç. Dr. Mehmet Amaç GÜVENSAN

Günümüzde ürün ve hizmet satışlarının gerçekleştiği çoğu dijital sistemde çalıntı kredi kartları ile teşebbüs edilen sahtekarlık vakaları hızla artmaktadır. Literatürde sahtekarlık tespit sistemlerinde en yaygın olarak kullanılan verisetleri bankacılık sektörüne ait olmasına rağmen, havacılık endüstrisi sahtekarlık vakalarından en çok etkilenen sektörlerden birisidir.

Bu tez çalışmasında, öncelikle havacılık alanında modern makine öğrenmesi algoritmaları ile literatürde ilk defa bir ana hat oluşturulmuştur. Daha sonrasında birbiriyle bağlantısı olmayan işlemlerden oluşan işlemsel verisetlerinin yol açtığı bağıntısızlık probleminin önüne geçebilmek için derin öğrenmeye dayalı yeni, çok modlu bir profilleme mekanizması sunulmuştur. Çok modlu profilleme mekanizmasında kredi kartı profili, PNR(Passenger Name Record) profili ve kayıt profili olmak üzere fraud tespitine katkı sağlayabilecek üç farklı profil yer almaktadır.

Oluşturulan bu yeni sistemde dengesiz veri seti probleminin ortadan kaldırılabilmesi için Dengeli Rastgele Aşağı Örnekleme (BRUS) algoritması geliştirilmiştir. Bu algoritma sayesinde çoğunluk kayıtlar arasında verisetinden dengeli şekilde seçim yapılabilen ve kayıt seçiminde duplikasyonun engellendiği bir mekanizma ortaya çıkarılmıştır. BRUS algoritmasının profilleme mekanizması ile birlikte kullanılmasıyla hem aşağı örnekleme yapılmış hem de profiller sayesinde seçilmeyen kayıtların bilgileri korunmuştur.

Son olarak ise geliştirilen sistemin hem işlem sayısına hem de maliyete göre ölçüm yapabilmesi için maliyet tabanlı bir metrik sistemi ve bu metrikleri kullanan maliyet tabanlı bir ölçüm mekanizması geliştirilmiştir. Maliyet tabanlı bu ölçüm mekanizması sayesinde geliştirilen sahtekarlık tespit sisteminin hem finansal başarısı hem de müşteri memnuniyeti anlamındaki başarısı gözlemlenebilmektedir. Yeni oluşturulan profil tabanlı verisetinin, havacılık verilerine ilişkin iyi bilinen hiperparametreler ayarlanarak ince ayar yapılan derin bir sinir ağına beslenmesiyle önemli bir başarı elde edilmiştir.

Anahtar Kelimeler: Havacılıkta fraud tespiti, profil tabanlı sistemler, derin öğrenme, maliyete hassas ölçüm, geçmişe duyarlı tespit

1
INTRODUCTION

In recent years, cases of fraud committed with stolen credit cards have increased on digital platforms where product sales and money transfers occur. This increase causes companies to experience serious revenue losses and customers to be dissatisfied. The aviation industry is one of the sectors where these losses are experienced the most. Due to the globally accepted civil aviation laws in aviation, airline companies are required to refund customers whose credit card is stolen in order to prevent them from being victimized. These repayments lead to quite severe losses. For this reason, it plays a negative role in corporate income and may even cause the closure of some loss-making flight stations. In addition, passengers victimized by false-positive decisions turn to other airlines or transportation options, causing severe revenue losses to airline companies. These situations, in addition to the losses suffered by airline companies, may also lead to a loss of prestige in our country due to the flag carrier airline companies.

Fraud detection mechanisms have been developed over time to prevent fraud attempts and revenue losses. The most commonly used fraud detection systems in the literature and the industry are rule-based systems [1, 2]. These detection systems generally work using in-house data or credit card monetary transaction information held by banks. Rule-based fraud detection systems are systems where business units in relevant companies use their expertise to create static rules to stop illegal purchasing attempts at the moment of transaction. These rules structurally include the conditions to decide which requests coming to the payment system should be rejected or accepted. However, these systems rely heavily on expert opinions both when defining detection rules and when deciding what action to take against suspicious transactions. Therefore, they are inadequate and error-prone against fraud attempts, which are rapidly increasing in number and type. In addition, these systems cannot go beyond predetermined rules or detect fraud attempts outside this framework. They offer reactive solutions instead of proactive ones and can cause significant financial losses when faced with new fraud methods.

Rule-based systems using static rules are inadequate against complex fraud patterns. Therefore, automated systems that use machine learning and deep learning instead of traditional rule-based fraud detection systems can provide more successful results [3, 4]. Fraud detection systems, especially those created using deep neural networks, can detect more complex cases. However, these methods alone are insufficient to develop an effective fraud detection system. In order to develop a successful fraud detection system, some key points must be taken into consideration, and frequently encountered problems must be solved [5].

One of the most important problems in the literature is that datasets consisting of independent transactions limit the success of fraud detection. Temporality and chronological order are of great importance in detecting fraud. Therefore, historical connections between transactions will enable holistic evaluation at the fraud detection stage. Information on a single transaction may not be sufficient for the decision-making process in fraud detection. Therefore, information about other related transactions that took place before a transaction will also enable the developed fraud detection system to have historical awareness. Although there are some different methods to design a system with historical awareness, one of the most effective solutions is to switch to a profile-based approach instead of a transactional approach. This change is accomplished by creating meaningful profiles by using the historical data linked to each profile, which can directly affect the detection method. In this study, we provide a unique profile-based fraud-detection system for suspicious airline ticket activity to distinguish between a fraudster and a regular airline customer based on their prior behavior. By incorporating discriminative time-line characteristics, we provide a variety of historical and in-record profiles in our system, which directly affects the success of fraud detection in the aviation sector. Thanks to these profiled fields, each transaction may be aware of the connected transactions that came before it. Furthermore, an analysis was conducted on the combinations of these created profiles to determine which combination was best for the aviation area.

From the standpoint of businesses involved in fraud cases, fraud is a financial issue. Thus, minimizing financial losses and averting future income declines is the primary goal of creating fraud-detection systems. Nonetheless, the majority of research [6, 7] computes the success rates using quantity for both traditional and automated detection methods. These measures cannot fully demonstrate the effectiveness of fraud-detection systems, which depend on the quantity of erroneous or genuine identified transactions. Because of this, we included cost-based measures in addition to quantity-based metrics in our study to assess the suggested profile-based design. This makes it possible to examine the degree to which loss reduction

and potential profit optimization may be accomplished by the created detection system. Maintaining the equilibrium between customer satisfaction and business profitability requires the detection system to generate results based on both cost and quantity.

Our thesis makes the following contributions to the literature:

- To the best of our knowledge, no research has been done specifically for the aviation sector in the literature that addresses fraud-detection issues and develops detection systems utilizing cutting-edge supervised learning methodologies. In this work, one of the biggest airline activity datasets with a specified data format and supervised machine learning (ML) methods were used to build a baseline for the first time in the aviation area. After creating the baseline and analyzing the ML findings, we utilized deep neural network (DNN) as our primary classification approach to improve the system's overall performance. To improve the DNN architecture's performance, we additionally tweaked its hyperparameters using aviation data.
- Our development of a multi-modal profile-based fraud-detection system was spurred by the analysis of passenger actions. This allowed us to eliminate the backward connection-restrictive structure of transactional information. This fraud-detection system explicitly designed for the aviation industry may create historical links by utilizing power from three distinct profiles and several statistical characteristics within these profiles. As a result, it dramatically increases the success rate of fraud detection, where historical background is of great importance.
- Although the consequences of fraud are essentially financial in nature, there aren't many studies in the literature that utilize cost-sensitive measurement metrics. Financial loss is a factor brought to light by the fact that airline companies are required by civil aviation laws to regulate customer grievances. In this study, we attempted to achieve a balanced success by measuring the effectiveness of the fraud-detection system not only in terms of quantity but also using the cost-sensitive metrics that we recommend.
- The imbalanced dataset problem, which arises most often in fraud detection, has been addressed by the balanced random undersampling (BRUS) technique. A mechanism in this method distributes the dataset equally and avoids duplication in the sample selection process. Our study also employed this technique as the initial stage in hybrid sampling testing.

1.1 Literature Review

It has been seen in the literature that different models of detection mechanisms are utilized to detect and prevent various fraud cases. As business areas and data contents change, these proposed detection systems have also changed. Still, most have not gone beyond being narrow-scope studies heavily dependent on data or business fields. Although fraud attempts can occur in almost every field where financial gains are made, the area where fraud attempts are most common is stolen credit card cases. These cases occur when the credit card or credit card information is captured. In addition, there are areas such as insurance fraud, credit abuse, embezzlement of corporate assets, tax fraud, and e-commerce fraud.

Fraud occurrences have been reported in many industries, particularly since online payments have become popular. In the banking industry, where the majority of research has been conducted, many strategies are advised for creating effective fraud-detection systems and resolving specific fraud-detection issues, including [8–12]. The incidence of fraud may be higher than in other industries since the virtual point-of-sale (POS) terminals owned by banks serve as the typical gateways for sales transactions. One of the industries where fraud detection studies are conducted is e-commerce, which sees a high volume of online payment transactions [13–15]. Fraud charges are common in the insurance industry since significant payouts may occur. Therefore, in an effort to proactively stop insurance abuse, several fraud-prevention studies, including those found in [16–18], have been conducted. There isn't much research on aviation fraud in the literature despite the aviation industry being one of the most frequently targeted by fraud cases. Airlines buy tools and services for fraud detection from some technology suppliers like Cybersource [19]. Unfortunately, the efficacy and sustainability of the fraud-detection systems now in use are limited due to their traditional rule-based design. The literature contains fraud-detection research that falls into three basic categories: systems that exploit history-aware detection approaches, modern automated systems, and conventional rule-based systems.

The majority of fraud-detection systems have a few significant issues that need to be fixed. The most critical issue is the imbalanced dataset problem. This problem is commonly solved using sampling strategies [20, 21]. Certain research proposes hybrid strategies that include many modifications to sampling procedures. Hanskunatai [22] worked on a three-phased hybrid sampling mechanism that applies undersampling and oversampling techniques as well as DBSCAN for clustering. On the basis of F1 Scores in tests conducted on various datasets, an increase of up to 69.91% for the SMOTE algorithm, up to 19.34% for the Tomek

Links method, and up to 59.93% for the non-sampling scenario was noted when this hybrid mechanism is applied. This study does not address the notion of temporality inherent in fraud detection; hence, undersampling may result in the loss of transactions in some time periods of the dataset. Furthermore, neither direct nor indirect safeguards were included to address the issue of linked transaction non-selection during the undersampling phase. Ultimately, it is incorrect to exploit the accuracy measure alone as a success criterion in the realm of fraud, where the imbalanced dataset problem arises. Instead, using the F1 Score or precision/recall measures will make assessing how well the built system performed easier.

The first systems created to stop fraud instances were rule-based, and many businesses still employ them today. These systems go through phases where the rules are created and tested, and choices are made by looking at circumstances in which the rules do not apply. All phases of the rule-based approach are completed manually. Because of this, rule-based systems exploit semi-automated classical methods that require a lot of manual labor and find it challenging to identify complicated patterns. Studies have been done to improve the success rate and decrease the amount of manual labor involved in the procedures of creating and managing rules. Garcia [23], for instance, utilized Web Ontology Language (OWL) and SemanticWeb Rule Language (SWRL) to construct a fraud-detection rule ontology. The authors of this work attempted to improve rule-based systems that were already in place by using a semantic approach. They utilized a dataset that included an existing rule-based system's anti-fraud rules. First, the relationships between fraud-detection rules were established. Conflicts were then discovered when these criteria were put to the test on an actual fraud dataset. These disclosed conflicts highlighted the system's shortcomings, which were fixed. This study revealed that removing the rules that cause the system to make poor decisions increases the accuracy of the results. Febriyanti [24] attempted to eliminate the detrimental impact of rigid rules on success in another rule-based fraud-detection study by creating dynamic rules as a consequence of business process analysis. This paper proposes a rule-based approach that can adapt to the ERP system's continuously changing event logs in order to detect fraud. The land-management procedure in a sugar company's ERP system is examined as a work topic. They made use of a dataset made up of business process event logs. The data model for these log records was chosen based on business operations. A method was devised to extract dynamic rules from business processes rather than producing static rules. When these automatically created dynamic rules were integrated with the preexisting static rules, the accuracy rose from 74% to 96%. Garcia [23] employs strategies in his work that will improve rule-based systems' performance,

but creating and managing these rules by hand is labor-intensive. Furthermore, the research's fraud detection was limited by specific rules, which made it challenging to identify intricate patterns.

The complexity of fraud detection is rising daily, mostly because of the prevalence of zero-day attack methods. The hallmarks of fraud are constantly changing and never stay the same. These factors make it necessary for fraud-detection systems to reinvent themselves and adjust to novel forms of fraud constantly. Rule-based systems, however, are unable to provide this flexibility and adaptability. It takes a lot of human labor to add new rules and rearrange old rules in response to evolving fraud tendencies. Furthermore, this procedure is prone to mistakes because it is mainly carried out by experts. Currently, highly adaptive automated fraud detection systems are preferred. With little user involvement, these systems can proactively guard against evolving fraud flows. Even though automated systems can have a variety of topologies, the majority of these systems exploit machine learning and deep learning techniques. In his study, Alarfaj [9] contrasted the Convolutional Neural Networks (CNN) fraud-detection performance with that of many machine learning techniques. The two-day credit card transactions that occurred in October 2018 made up the dataset. The CNN method was utilized to assess and contrast the well-known shallow learning algorithms, such as decision tree, k-NN, logistic regression, SVM, random forest, and XG Boost. The test findings demonstrated that the structure they constructed with CNN performed better in terms of accuracy. The F1 Score parameter is one of the most accurate success measures in the fraud detection sector. Accuracy is a misleading performance indicator due to the imbalanced datasets in the fraud detection area. Furthermore, given that the study only looks at credit card transactions for a two-day period, it's critical to remember that the nature of the fraud problem makes results for such brief timeframes potentially misleading because transactions in the fraud dataset may be connected to older transactions. Many fraud tendencies will remain hidden if the training set's temporal distance is limited to two days. In a similar vein, the outcomes of some machine learning algorithms were compared with the deep network architecture developed in Yu's study [25]. It's been said that using deep learning techniques yields more successful results. Nevertheless, this study had no cost-based measuring system, and the measurement outcomes were assessed using the accuracy metric. Furthermore, the issue of disregarding the linked history resulting from the nature of transactional databases remains unresolved. However, Chang [26] asserted that the most essential step in the fraud detection process is feature selection, as it influences the decision mechanism. They argued that artificial feature development approaches become time-consuming and inadequate due to the

continually changing nature of fraud characteristics and Internet banking scenarios. They suggested an automated feature-engineering technique based on deep feature synthesis and feature selection to stop this denial. Using this procedure, the original data table is divided into smaller parts in order to produce features. The process of developing features and choosing crucial traits for fraud detection is more efficient. This approach greatly minimizes human labor during the feature-engineering stage and does not require the input of business specialists. Overfitting issues, however, will surface if a balanced model with accurate auto-tuning cannot be produced during the feature-engineering stage. As a result, the generated features will only work with the training set. Furthermore, the deep feature synthesis approach could result in high running costs for big datasets.

One of the most crucial needs for fraud detection systems is historical record information, as fraud is a temporal and historical problem. As a result, the success attained with the transactional information alone is limited. The history awareness capacity in the fraud-detection system may be obtained by several strategies, such as utilizing algorithmic memory techniques or altering the dataset to incorporate historical relationships. Some research has made use of this concept. To improve fraud-detection systems, Olszewski et al. [27] attempted to map users using the self-organizing map (SOM) approach. In this investigation, matrices made out of records were utilized instead of vectors, as was the case with the SOM technique's original version. Consequently, these matrices capture the users' sequential actions. In the process of employing these matrices to identify fraud, a novel threshold method is suggested. In this study, users were simply grouped based on their actions. The objective is to identify fraudulent transactions by analyzing the deviation in the user's behavior. However, credit cards and other items with individual activity histories were disregarded. A novel method for profile-based fraud-detection systems that could adjust to shifting fraud tendencies was put out by Malekian [28]. In addition to the historical profile, the suggested method includes a temporary profile that allows for the perception of novel notions. Transactions are relayed into both profiles in real-time. It is acknowledged that new concepts appear, and the present model starts to become outdated at the point where the temporal profile and the historical profile diverge in their judgments. The fact that this study solely considered changes in cardholder behavior is one of its limitations. The system won't be able to identify any modifications if attempts are made using several credit cards. Behaviors of all data objects that might have an impact on the outcome need to be assessed in order to avoid this issue. Furthermore, according to Seyedhossein [29], a transaction-level fraud-detection system is unable to realize its full potential for success. Instead, they suggested a fraud-detection system that

relied on profiles created from daily credit card spending data, which was then utilized during decision-making. By examining the variations in daily expenditure amounts and the movements over time, they attempted to identify any trends. They tried to reduce the time that passed between realizing and identifying the fraud attempt. Nevertheless, only credit card-specific profiles and daily spending amount records were collected in this investigation. Although this study is among the first to exploit profiling procedures, it is neither multi-modal profiling nor does it take advantage of extensive profile features.

One method that utilizes Long Short-Term Memory (LSTM) to construct a deep network architecture where important data is stored in memory units is the consideration of previous transactions. A few research, such [30–33], have utilized LSTM to guarantee the relationship and continuity between transactions in the transactional dataset. However, unrelated transactions will also be considered because the memory units to be constructed in studies employing LSTM will have a global scope without any grouping or profiling applied to the data. As a result, the system won't be completely aware of history, and information will become more complicated in terms of historical connections. In addition, several statistical data fields associated with these profiles and other profiles that could influence the outcome would be absent from LSTM.

To the best of our knowledge, there hasn't been much discussion of the fraud-detection issue in the aviation sector, and no thorough findings have been provided thus far. This research is the first in the aviation industry to examine the issue of fraud detection, attempt to overcome obstacles and provide a thorough presentation of the findings. In order to mitigate the negative effects of random selection, we first provide the baseline in this study and then exploit the suggested BRUS method to choose the relevant occurrences from the whole dataset. To create a reliable fraud-detection model for aviation, we also take use of the oversampling approach and integrate it with the BRUS algorithm to get the ideal ratio between fraud and non-fraud cases. Next, we utilize the historical relationship between passenger actions to introduce novice-engineered traits that we received from the suggested profiles. Finally, we utilize cost-sensitive measurements in our evaluation of our mechanism's success in addition to traditional metrics.

1.2 Fraud Domains

The most common areas of fraud are categorized below as in Figure 1.1.

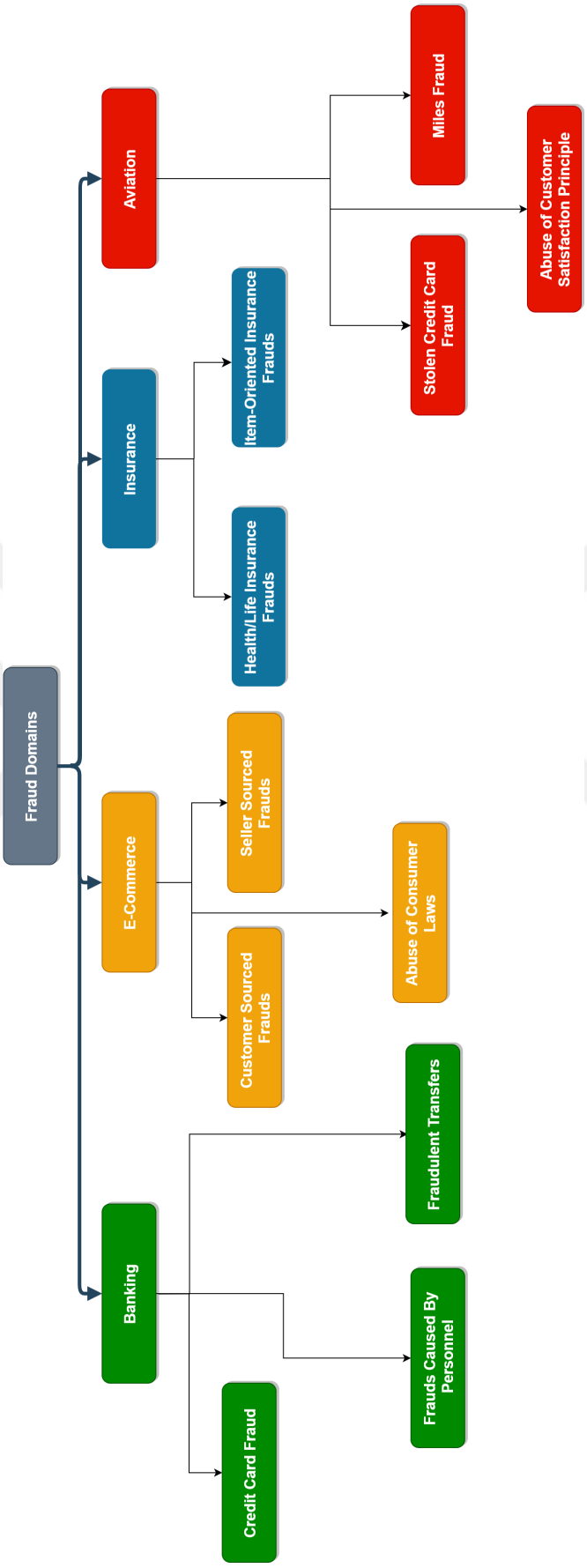


Figure 1.1 Fraud types according to their domains

1.2.1 Banking

The banking sector is the area where fraud attempts occur the most. Regardless of the field of fraud attempts made with stolen credit cards, they pass through the transaction traffic of banks and fall under their responsibilities. Fraud attempts made by fraudsters using stolen credit cards in online channels attempts to illegally transfer money from other customers' accounts to their own accounts, and employees' attempts to extort money from the company are among the fraud attempts frequently experienced in the banking field [8, 34].

Since bank employees are familiar with the processes, they can catch some of the bank's vulnerabilities and transfer money to their personal accounts. Although this situation is difficult to catch, fraud cases can be handled before they occur by processing personnel activities and transaction logs.

If credit or debit card information is stolen, these cards can be used for various purchases and money withdrawal/transfer transactions. Banks are responsible for detecting and blocking these fraud attempts before they succeed. If the transaction is an online purchase made with credit card information, the seller company may also be responsible for this monetary loss. Unless there is an apparent fault or negligence of the cardholder, the loss must be covered.

Money transfers can be made from customer accounts to fraudsters' accounts by capturing customer account information and personal information. The unusual characteristics of such transactions must be immediately detected and blocked. In addition, institutions such as BKM (Interbank Card Center) manage the transactions of all banks in that country from a single place and establishing a fraud detection system on this huge data will be much more comprehensive.

1.2.2 E-Commerce

The e-commerce sector is an environment where various products are sold over the internet, and a significant amount of stolen credit cards are used in these sales. Since the products sold in this sector do not have a specific area restriction, the data richness is greater, and it is possible to examine a wide range of transactions holistically [35]. In addition, as the e-commerce industry evolved from centralized sales to a model that allows personal sales, sellers began to victimize customers with various fraud attempts.

This fraud area generally represents the part of the e-commerce system that purchases products/services with stolen credit cards and customer information.

While fraud attempts were previously observed only by customers, since sellers can also be individuals, fraud attempts are also observed by sellers. While customers spend with stolen credit cards, sellers may resort to methods such as distortion of the products they sell, selling fake or used products, and fraud in return processes. Until fraud attempts are detected, sellers who close their virtual stores with the profits they have made so far open new stores with entirely different information. Even if legal processes are in place, those who suffer the most from such fraud cases are the companies that provide the sales transaction and the banks to which the card belongs. Therefore, in order to prevent these losses, e-commerce companies have their own fraud detection mechanisms.

1.2.3 Insurance

The insurance industry is one of the areas where fraudsters resort to various irregularities to obtain insurance money from the insurance company unfairly. By pretending that the conditions for insurance payment have been met even though they have not been met, insurance companies can incur significant losses [36, 37].

People-oriented or property-oriented insurance can be obtained at high costs. Fraud attempts are made against the insurance company by presenting these insurances as if the payment terms have been revealed. For example, in life insurance, it is possible to show the person as if he or she is dead, to present the vehicle as if it had an accident, to present the person as if they have suffered any severe disability, etc. Insurance companies may suffer heavy losses in such frauds by preparing false documents and evidence. Therefore, fraud detection systems are added to the information processing flows of insurance companies to try to detect abuses and fraud attempts.

1.2.4 Aviation

Tickets/services purchased from airlines with stolen credit cards oblige the airline company to compensate the victim customer. Airline companies resort to various methods and measures to prevent these revenue losses they face. The solutions implemented as precautions generally do not go beyond rule-based systems. However, due to the rapid expansion of the airline industry, the increase in the number of flights, and the shift of travel preferences to airlines, the volume of data processed is increasing very rapidly. Since rule-based systems can only survive with manual efforts, they have difficulty coping with these huge data volumes and become error-prone systems.

Unlike e-commerce and bank data types, rather than product and bank transaction-oriented data, the airline industry uses data focused on flights, passengers, payment, and customer movements (Passenger Name Record) (PNR). The fact that this data type is different from most other sectors that are subject to fraud attempts and that civil aviation laws protect customers in this sector makes fraud detection in the aviation field a unique study subject.

1.2.4.1 Fraud Types in Aviation

Fraud types in the aviation industry are classified according to where they originate and where they are reported. These types are listed as follows:

- **Indirect Fraud:** If the passenger traveling in Business is unsatisfied with the service, he/she can request the ticket price from the airline company. When abused, this right is indirectly considered a type of fraud.
- **Fraud:** It is a direct fraud process. The Occupational Safety Unit has determined this in advance. It is mainly due to stolen credit cards.
- **TC 40:** These are fraudulent transactions carried out by Türkiye İş Bankası.
- **Prosecutor's Letter:** Traveller names submitted through the prosecutor's office are recorded in the system.
- **Ethoca Notification:** It is a 3rd party company from which fake transaction pools are obtained.

1.2.4.2 Aviation Fraud Data Structure

Data structures used globally in the aviation field consist of 5 essential parts as follows. In addition, the grouping and usage status of the features in the dataset are stated in Table 1.1

- Flight Data
- Passenger Data
- Card Holder Data
- Payment Data
- Auxiliary Data

Table 1.1 Available features in our dataset supported by a worldwide airline company. (P&D: Profiled and Discarded)

Feature	Group	Type	Distinct Categories	Status
Request_Date	Flight Data	Date	N/A	P&D
Payment_Instrument	Payment Data	Categorical	5	Used
Office_Number	Auxiliary Data	Categorical	950	Used
Amount	Payment Data	Numerical	Continuous	Used
Currency	Payment Data	Categorical	16	Used
TDS	Payment Data	Categorical	2	Used
MD_Status	Payment Data	Categorical	10	Used
Installment	Payment Data	Categorical	36	Used
Payment_Service_Provider	Payment Data	Categorical	12	Used
SubChannel	Payment Data	Categorical	12	Used
Sale_Result_Code	Payment Data	Categorical	2	Used
Client_OS	Passenger Data	Categorical	54	Used
CardHolder_Name	Payment Data	Categorical	880107	P&D
CardHolder_Surname	Payment Data	Categorical	1057694	P&D
Email	Passenger Data	Categorical	5288442	Profiled
Phone_Number	Passenger Data	Categorical	6512043	P&D
FFP_Number	Passenger Data	Categorical	1180514	Used
Card_Number	Payment Data	Categorical	5773393	P&D
Card_Origin_Country	Payment Data	Categorical	188	Used
Card_Brand	Payment Data	Categorical	11	Used
Card_Program	Payment Data	Categorical	9	Used
Card_Type	Payment Data	Categorical	3	Used
Origin_Country	Flight Data	Categorical	157	Used
Destination_Country	Flight Data	Categorical	153	Used
Reservation_Code	Flight Data	Categorical	12589975	Used
Journey_Type	Flight Data	Categorical	4	Used
Flight_Type	Flight Data	Categorical	3	Used
Departure	Flight Data	Categorical	757	Used
Arrival	Flight Data	Categorical	761	Used
GMT	Flight Data	Categorical	33	Used
Flight_Date	Flight Data	Date	N/A	P&D
Flight_Time	Flight Data	Categorical	1088	P&D
Carrier	Flight Data	Categorical	140	Used
Cabin	Flight Data	Categorical	2	Used
Cabin_Class	Flight Data	Categorical	20	Used
Passenger_Name	Passenger Data	Categorical	1139421	P&D
Passenger_Surname	Passenger Data	Categorical	1114833	P&D
Client_Name	Auxiliary Data	Categorical	11	Used
Response_Value	Payment Data	Categorical	4	Used
Charged	N/A	Categorical	2	Class

The majority of these data fields are not suitable for direct use in a deep-learning model. Therefore, the dataset consisting of these fields will need to undergo some

transformation, pre-processing, and data-cleaning processes. For example, although the credit card number does not mean anything on its own, it can be utilized for the following purposes:

- Whether or not the bank received a "Declined" response in previous attempts for the same PNR
- Number of "Accept" and "Reject" numbers for this card's purchases in the specified time period
- Whether a chargeback has been issued for this card before due to fraudulent transactions

These processes will be discussed within the scope of data preprocessing and profiling in the future stages.

1.2.4.3 Current Rule-Based Systems in Aviation

These are fraud detection systems developed or purchased by airline companies in order to prevent fraud attempts that are frequently encountered in the aviation industry. Sales transactions made on Web, Mobile, or Call Center channels via Virtual POS are handled.

Basically, rule-based systems can make three different decisions by processing the rules registered in the database according to their priorities when detecting fraudulent transactions.

- A transaction is deemed reliable and accepted. These decisions are labeled "Accept". In accepted transactions, sales occur directly.
- A transaction can be detected as fraudulent and will be rejected. These transactions cannot proceed to the ticketing stage. The label for these decisions is "Reject". In rejected transactions, the sale is blocked directly.
- If any opinion on a transaction cannot be executed, that is, if it falls outside the scope of the rules in the database or is caught by the suspicion rules, the system labels "Review" for these transactions. For these transactions, sales are made by presenting a card. However, expert personnel in the transaction security department check these transactions individually and can manually block the suspicious transactions or remove the card presentation requirement by approving reliable transactions.

1.2.4.4 Fraud Rates in Aviation

As in all fraud detection literature, the rate of fraud transactions in the aviation field is very low. Therefore, the imbalanced dataset problem arises, making fraud detection very difficult. Figure 1.2 shows the rates of fraud transactions according to the sales channels of a major airline company.

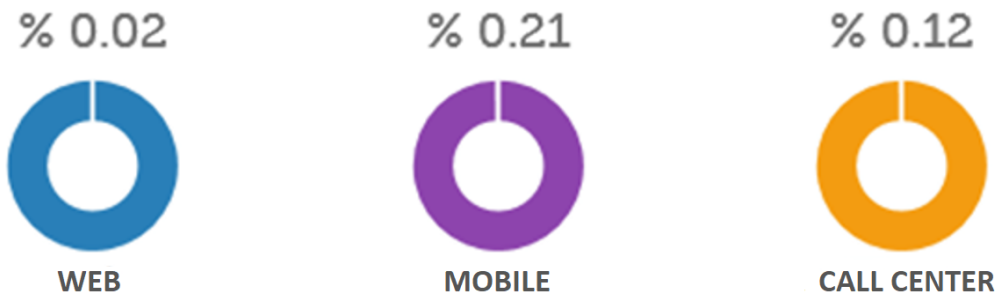


Figure 1.2 Fraud Rates in Aviation

1.3 Objective of the Thesis

The aim of this thesis is to design and develop a fraud detection system that can successfully detect fraud attempts on payment request data made via virtual POS in an airline company, by using deep learning techniques and producing up-to-date and valid solutions to fraud detection problems frequently encountered in the literature.

We aimed to identify the problems encountered in detecting fraud in the aviation industry and to provide up-to-date and valid solutions to eliminate these problems. In this regard, Imbalanced Dataset, Ineffective Random Instance Selection, Cost-Senstive Measurement, and Fraud Detection on Transactional Information problems were identified, and various solutions at different stages of the fraud detection system were developed against these problems.

1.4 Hypothesis

Fraud is a problem that involves temporal connections. Therefore, there are often links between fraud attempts. However, transactional information is mostly utilized in fraud detection studies in the literature and fraud detection systems developed in the sector. There is no connection between the records in transactional datasets; thus, the records have no awareness of the past. Our first hypothesis is that designing a profile-based fraud detection system will reveal the relationships between records. Thus, success rates will increase compared to the transactional approach.

In fraud, one of the areas where the imbalanced dataset problem occurs most

frequently, sampling methods are commonly utilized to solve this problem. However, the selection of legal records in undersampling methods and processing times of big datasets in oversampling methods may pose problems. Another hypothesis in our thesis is that developing a balanced selection algorithm that also prevents duplication when applying undersampling will demonstrate superior success compared to random downsampling. In addition, using the balanced sampling algorithm to be developed with the profile-based fraud detection system will indirectly reduce the problem of not selecting some of the relevant records.

Fraud is primarily a financial problem for companies. For this reason, the financial return to companies should be considered when discussing the success of a fraud detection system. However, since many studies only consider success based on the number of transactions, the fraud detection systems developed are not financially optimized. Our last hypothesis is that when both the number of transactions and cost-based measurement criteria are utilized in the solutions developed within the scope of our thesis, the resulting fraud detection system will have balanced success in both aspects.

2
DATASET

In our study, the labeled dataset consisting of payment transactions received via virtual POS of the fraud detection application called TKFraudFight belonging to the Turkish Airlines company was exploited. This dataset consists of 37,303,697 transactions consisting of 66 features for 15 months between October 2017 and December 2018. Since dates, times and chronological order are important in the field of fraud, the dataset was divided into monthly parts, preserving the chronological order.

This dataset contains 5512 fraudulent transactions, and the fraud rate is 0.016%. The fraud labels in this dataset are marked because the rule-based TKFraudFight system makes the wrong decision, and victimized customers complain about the relevant transactions to the bank.

The dataset has 18.29% missing values. This high rate is due to differences in in-house sales systems and the fact that customers are not required to fill in all fields during sales. While the rate of missing values is very high in some areas, there are no missing values in some areas, which supports this theory.

When fraud rates are examined according to the fields in the dataset, some of the risky areas that are thought to have an impact are shown in Table 2.1. In this table, the lines shown in green are the most reliable distributions, and the lines shown in orange are the most risky distributions. Only categories with the most significant fraud distribution are considered.

All fields in the raw dataset are shown below with their definitions:

1. **RequestDateTime:** The date and time when the sales request received via virtual POS reaches the TK Fraud Fight application.
2. **ID:** Unique ID of the incoming request

Table 2.1 Fraud Distributions by Category in Significant Data Fields

Flight Type			
Category	Fraud Percentage	Fraud Count	Legal Count
International	0.19%	4791	2480209
Domestic	0.02%	638	3380950
TDS			
Category	Fraud Percentage	Fraud Count	Legal Count
False	0.16%	5021	3155148
True	0.02%	489	2850120
Card Origin Country			
Category	Fraud Percentage	Fraud Count	Legal Count
EC	8.58%	120	1278
IS	4.34%	16	353
HK	3.17%	190	5806
GE	0.00%	0	13288
GR	0.00%	0	11751
PL	0.00%	0	6436
Payment Instrument			
Category	Fraud Percentage	Fraud Count	Legal Count
Amex	0.31%	848	276270
Miles	0.00%	0	120923
Payment Service Provider			
Category	Fraud Percentage	Fraud Count	Legal Count
Isbank_ABD	0.68%	1634	238803
Amex	0.50%	841	166429
Global_Collect	0.22%	16	7162
Finansbank	0.002%	4	190954
Akbank	0.00%	0	281659
Comarch	0.00%	0	120923
Currency			
Category	Fraud Percentage	Fraud Count	Legal Count
CNY	2.39%	140	5728
USD	0.43%	2891	664048
DKK	0.25%	44	17221
PTS	0.00%	4	120923
AUD	0.00%	0	3691

3. **RequestDate:** Field containing only the date of the sales request received via virtual POS

4. **TransactionType:** Type of incoming sales transaction (Sales, Pre-Sales, FraudCheck)

5. **PaymentInstrument:** Instrument of payment (Credit card, debit card, miles)

6. **ExecutionPhase:** It is the technical phase where the request takes place

(transactions that have passed through the payment gateway are taken into account)

7. **OfficeNumber:** Office code from which the sales request came
8. **CityCode:** City code of the office where the sales request comes from
9. **Amount:** Total amount of the requested sale
10. **Currency:** Currency information of the requested sale
11. **TDS:** Whether the sales transaction will go through the 3D Secure mechanism
12. **MDStatus:** If the sales transaction goes to 3D Secure, the status code that will be returned from there
13. **ClientIP:** Device IP sending the sales request
14. **Installment:** Number of installments requested in the sale
15. **PaymentServiceProvider:** The bank where the POS where the sales transaction will be made belongs
16. **SubChannel:** The channel through which sales occur in the airline company
17. **SaleResultCode:** Approval or rejection code returned from the airline's Payment Gateway application for various reasons
18. **BankResultMessage:** This is the response message returned from the bank regarding the sales transaction. Since the format of each bank is different, there are many different forms of response messages.
19. **ClientOS:** Operating system of the device sending the sales request
20. **ClientIPCountry:** Country of the client sending the sales request
21. **ClientIPCity:** City of the client who sent the sales request
22. **DeviceType:** Device type of the client sending the sales request
23. **CoverageApp:** Campaign information organized by the airline
24. **AgencyID:** This is the code of the agency that sent the sales request.
25. **CHName:** Cardholder name declared by the customer
26. **CHSurname:** Cardholder surname declared by the customer

27. **Email:** Email address declared by the cardholder
28. **StreetAddressA:** Address declared by the customer, Part 1
29. **StreetAddressB:** Address declared by the customer, Part 2
30. **City:** City information declared by the customer
31. **State:** State information declared by the customer
32. **CountryCode:** Code of the country declared by the customer
33. **PostalCode:** Customer declared zip code
34. **PhoneNumber:** Telephone number declared by the customer
35. **FFPNumber:** Customer's loyalty program number (if available)
36. **FFPProgramId:** The airline with which the customer is enrolled in the loyalty program
37. **FFPStatus:** The customer's status in the loyalty program
38. **FFPOperationType:** For which service the loyalty program is used in this transaction
39. **CardHolderName:** Name written on the card (This is the information returned from the bank)
40. **CardNumber:** Card number used in the sales transaction
41. **BankName:** The bank to which the credit card belongs
42. **CardOriginCountry:** Country of origin of the card
43. **CardBrand:** Brand of the card (Visa, Mastercard)
44. **CardProgram:** Program of the card (Maximum, World, Axess)
45. **CardType:** Type of card used (Credit, Debit)
46. **OriginCountry:** Country of Departure
47. **DestinationCountry:** Country of Destination
48. **ReservationCode:** Passenger Name Record (PNR) consisting of Flight and Passengers
49. **JourneyType:** Type of trip (One way, Round Trip, etc.)

50. **FlightType:** Type of Flight (Domestic, International)
51. **ThirdParty:** Information that the flight is operated by another airline
52. **Departure:** Departure Station
53. **Arrival:** Arrival Station
54. **GMT:** Time Zone
55. **FlightDate:** Flight Date
56. **FlightTime:** Flight Time
57. **Carrier:** Carrier airline company
58. **Cabin:** Cabin Class
59. **CabinClass:** Fare class of the purchased seat
60. **PName:** Passenger's name
61. **PSurname:** Passenger's surname
62. **Id_TKFF2_Score_Transaction:** A field in the DB that holds relationship data needs to be deleted because it does not make sense.
63. **ResponseValue:** Decision made by rule-based TKFraudFight application
64. **ClientName:** Subclients forwarding the sales transaction internally
65. **FraudRiskScore:** Fraud risk score given by TKFraudFight application for the transaction using some relational information
66. **Charged (Class):** Information about whether a transaction is marked as fraud by notifications from the bank

3
CHALLENGES

The majority of fraud-detection research in the literature deals with a few fundamental issues. The aviation industry faces each of the difficulties listed in this section. Large datasets and cost issues may be particularly damaging in the aviation industry, as transaction volumes have reached extremely high levels in recent years.

3.1 Imbalanced Dataset

In nearly every fraud-detection arena, non-fraud records are significantly more common than fraudulent ones. As a result, practically all fraud-detection investigations in the literature, such as [20, 38, 39], run into this issue. One of these sectors where fraud-detection methods suffer from a significant degree of dataset imbalance is aviation. Because of this issue, in order to achieve greater accuracy values, fraud-detection algorithms often make poor judgments (i.e., decisions that are not fraudulent). However, because of this feature, the fraud-detection system will perform poorly in detecting fraudulent transactions, as evidenced by poorer recall and F1 Scores.

In aviation, legitimate records outweigh fraudulent ones by a wide margin, as in practically every other field of labor. For instance, the fraud record ratio in our dataset is 0.0002. This feature results in a significant imbalance issue, making it impossible to differentiate between genuine and fraudulent situations. The suggested models in such systems often tend to optimize accuracy by making only reasonable judgments [40, 41].

Typically, sampling techniques are employed to address this problem.

- In the undersampling technique, all of the fraudulent records are selected, and a target fraud/legitimate balance is used to choose some of the non-fraudulent records. The domain, interrelated transaction density, and dataset structure

can all affect this ideal balance. Additionally, legitimate records can be chosen at random, uniformly, or using a unique method that makes a decision by cleverly taking into account the connections between transactions [42, 43].

- To get the desired fraud/legitimate balance, some synthetic fraud records are created while all legitimate records are maintained using the oversampling approach. Based on the content of fraud records, these synthetic fraud records may be entirely new records or copies of existing fraud records [44, 45].

- When oversampling techniques cannot be applied due to the size of the dataset making it impossible to replicate fresh records, hybrid sampling techniques are frequently utilized [22, 46, 47]. This strategy exploits the undersampling method to some extent before applying oversampling instead of directly oversampling the entire dataset. Stated differently, hybrid sampling methods involve a two-phased sampling approach.

3.2 Smart Instance Selection

This is a side issue of the imbalanced dataset issue that arises when the main problem is solved by undersampling the dataset. Undersampling is one of the most straightforward techniques to address the imbalanced dataset issue. On the other hand, a sampling strategy employing a totally random selection method may choose records that have no connection. This might have a negative impact on the fraud-detection procedure and lower success rates. Therefore, rather than selecting records randomly, a smart instance selection mechanism that utilizes a specific intelligent algorithm to choose legitimate data can achieve higher success rates. By taking into account the relationships and linkages between transactions, undersampling may be carried out. For example, it is possible to choose accurate records by looking up the relationships between customers, flights, routes, and PNRs.

3.3 Cost-Sensitive Measurement

Since fraud directly impacts the prospective earnings and losses of businesses, it is frequently a financial issue. As a result, companies that employ only the quantity-based success criterion to detect fraud may suffer financial losses. With a few exceptions [48–50], the majority of research [51–53] in the literature, however, focus on fraud-detection systems that measure performance by quantity. Fraudulent transactions cause the aviation sector to suffer significant financial losses. Thus,

minimizing fraud-related losses and averting future revenue declines must be the main goals of any fraud-detection system created in this field. This study took into account a cost-centric approach to performance measurement, with the primary goals being identifying fraudulent activity and reducing financial losses for airline firms.

Cost-sensitive measurement highlights the potential drawbacks of quantity-based evaluation. A fraud-detection system's identification approach based on quantity measures may result in revenue losses. The financial aspect of fraud detection is entirely undermined when the systems' effectiveness is evaluated using quantity-based measures. Therefore, in order for fraud-detection systems to observe and compare cost- and quantity-based techniques, they require cost-sensitive indicators and a cost-based perspective.

3.4 Fraud Detection on Transactional Information

Several published research on the subject of fraud detection have attempted to identify fraudulent activity in datasets made up of credit card transactions [54–56]. However, since fraud is a temporal issue that is constantly changing, it is also crucial to consider the previous connections of the transaction under consideration. When it comes to the fraud-detection phase, a significant amount of information on the relevant transaction may not be sufficient since datasets containing transactional records lack historical links. This may result in mistakenly detecting a trustworthy transaction or, conversely, in failing to identify a potentially fraudulent transaction. The dataset must be arranged chronologically both inside the training and test sets as well as between them in order to tackle this issue. Next, either the dataset should be made historically sensitive or an algorithmic retro-operative control/memory sub-system should be created.

Transactional data in the aviation sector is more intricately connected than in standard credit card transactions. In order to develop a history-aware fraud-detection system and get over the transactional information problem in the aviation fraud domain, some profiles may be developed by making use of these links. To the best of our knowledge, the aviation industry has not created any fraud-detection systems that make use of past data. In order to address this prevalent issue, we present in this paper a profile-based fraud-detection system composed of aviation-specific profiles.

4
DATA PREPARATION

4.1 Data Pre-Processing

4.1.1 Data Cleaning

This phase is applied once on the dataset, and a new dataset is created by subjecting the dirty data fields to cleaning processes. These cleaning steps have been applied because the existing raw dataset is not suitable for machine learning algorithms, there is a very high number of missing data, and some of the content consists of misleading information. 4369 records were deleted during the data cleaning phase, and the number of records after the dataset was cleaned was 37,299,328. 32 of the 66 data fields on the raw data were deleted, and the number of data fields was reduced to 34. All static data pre-processing steps, except field deletion, were carried out with the application coded in Java.

4.1.1.1 Fixing Incorrect Representation of RequestDateTime Field

The RequestDateTime field is written in the dataset sequentially, but since it is written according to the 12H format, AM and PM times are mixed together in the raw dataset. To fix this, Java code was written to take advantage of the sequential nature of the data, and PM times were changed to 24H format. This arrangement was made in the sections that should have passed from 11 to 12 o'clock. For example, since the 00:02 time information coming after 11:59 is incorrect; it is corrected to 12:02.

4.1.1.2 Recovering Corrupted Rows

Some rows in the dataset were finished earlier than they should have been, or the number of columns was missing/more than they should have been. Although the number of these lines is not large, most could be corrected thanks to the written Java function. Rows that could not be corrected were removed from the dataset.

While the large number of lines may be related to the unexpected token problem, extra new line signs may also appear. To solve this, incorrect separator characters were deleted, or if the incorrect line could be combined with the following lines, they were merged. The below examples can be found in Table 4.1 consisting of the feature array that is tried to be parsed in each line. In this example, it reached the 7th line by parsing and successfully repairing the faulty lines, but since the 7th line is ambiguous, this line is eliminated and moved to the following line.

Table 4.1 Corrupted Row Recovering Examples

Features						**Action**
F1	F2	F3	F4	F5		No Action Needed
F1	F2	F3	F4	F5	F?	Incorrect Token Problem Fixed
F1 F4	F2 F5	F3				New Line Problem Fixed
F1	F2	F3	F4	F5		No Action Needed
F1	F2	F3	F4	F5		No Action Needed
F1	F2	F3	F4			Problem can't be fixed because of ambiguity
F1	F2	F3	F4	F5		No Action Needed

4.1.1.3 Filling in Missing GMT and CityCode Information

Some lines in the dataset contain blank GMT information. However, since GMT information is linked to the station from which departure took place, first a "Station" - "GMT" map was created. With the help of this map, empty GMTs were attempted to be matched. GMTs that cannot be matched are recorded as "NOT AVAILABLE". The same relationship was established between "OfficeNumber" and "CityCode," and attempts were made to fill in the records where the CityCode field was empty.

4.1.1.4 Filling in Missing PaymentInstrument Information

The "PaymentInstrument" field is blank for transactions not made via credit or debit cards. Such transactions are generally mile transactions. Therefore, if the "PaymentServiceProvider" field of transactions whose "PaymentInstrument" field is empty is "COMARCH", the "PaymentInstrument" field has been updated to "MILES". If the PaymentServiceProvider field is empty or not "COMARCH", this is unexpected, and the PaymentInstrument field has been changed to "UNKNOWN".

4.1.1.5 Filling in Missing Currency Information

Again, since cash is not used in transactions made with miles, the Currency field appears as PTS. However, since this rule was determined later, the Currency

field is empty in old transactions. Therefore, in this case, the Currency field has been updated to PTS in all transactions whose PaymentServiceProvider field is “Comarch”.

4.1.1.6 Filling in Missing MDStatus Information

In transactions made using 3D, a code is returned showing the 3D status of the transaction. This code takes a value between 0-8. However, in non-3D operations, this field is also empty. Therefore, the MDStatus information of transactions that are not 3D and whose MDStatus field is empty has been changed to "NOT3D". If the MDStatus field is empty even though it is 3D, it is recorded as “UNKNOWN”.

4.1.1.7 Filling in Missing BankResultMessage Information

BankResultMessage is the response returned from the bank after the transaction is sent to the bank. Although the value of this field varies from bank to bank, it may sometimes appear empty. In this case, the SaleResultCode returned by THY’s PaymentGateway application is taken as the basis because this field never returns empty. If the SaleResultCode field is “approved” or “declined”, this value is also written to the BankResultCode field. If another value comes in, the BankResultCode field is corrected to “Not Available”.

Note: We did not remove the BankResultCode field from the dataset because it contains many more types of result codes than SaleResultCode. Only "Approved" and "Declined" values are included in the SaleResultCode field.

4.1.1.8 Correction of Incorrect Installment Information

The Installment field is the data field that represents how many installments the sale was made in. If this field is empty or "0", it means that the installment has not been made and it is updated as "1".

4.1.1.9 Correction of Unexpected Tokens

The “ character is the value delimiter, and the ";" character is used as a column separator. Therefore, since the presence of these characters in the values will disrupt the parsing operations, the " character in the values has been replaced with the ’ character. This process should be done when the dataset is first created; otherwise, since the values of the data fields cannot be known exactly, parse errors will no longer be correctable.

4.1.1.10 Standardization of Missing Values

Since data comes to the fraud detection application from many sources, the missing data representation may change. Missing values can be empty or with values such as “null” or “nu”. Since this would make it difficult to clean the data, all missing values were changed to empty strings.

4.1.1.11 Correction of Amount Notation

Because some parsers cannot detect fractional amount fields, the "," character is replaced with the "." character.

4.1.1.12 Data Fields Deleted from the Dataset

- **Primary Data Field Deletion:**

  This stage was carried out entirely in Python using the Pandas library. The following fields were deleted directly because they were not likely to be utilized in our study and would not be useful. 10 data fields were deleted in this process:

  - **RequestDate:** There is another RequestDateTime field that also contains time information.
  - **TransactionType:** Contains a single value.
  - **Id_Tkff2_Score_Transaction:** It was included in the dataset with an SQL error, it is a Foreing Key field that has nothing to do with our study.
  - **ThirdParty:** Contains a single value.
  - **AgencyId:** Although it should be 10% - 20% full, almost all of it is missing. Therefore, it is a misleading area.
  - **StreetAddressA:** It contains much dirty data.
  - **StreetAddressB:** It contains much dirty data.
  - **CoverageApp:** There is a very high rate of missing values, and it is stated that there is no field unused by the business unit.
  - **ClientName:** It is the client application for the institution where the transaction comes to the TKFraudFight application. Since there is no field the customer can choose from, there is no distinguishing feature for detecting fraud.

- **Secondary Data Field Deletion:**

The following fields cannot be utilized directly in our study or have misleading/incomplete content. Although these fields are deleted from the dataset, they can be used in future stages such as “Feature Aggregation”, “Feature Engineering” and “Re-Sampling”. Therefore, there is a possibility that they will be included in the dataset again. 22 data fields were deleted in this process:

- **ID:** This unique value can be used to represent records in algorithms to be written.
- **CityCode, ClientIP, BankResultMessage, ClientIPCountry, ClientIPCity, DeviceType:** These data fields have large missing values. However, whether meaningful information can be obtained about the device used will be examined.
- **Email, City, State, CountryCode, PostalCode, PhoneNumber:** The number of missing values and incorrect entries are high in these data fields. However, it will be examined whether meaningful information about the customer can be obtained.
- **FFPNumber, FFPProgramID, FFPStatus, FFPOperationType:** These areas have large amounts of missing data. However, it can be exploited to create a new field indicating whether the customer is included in the loyalty program.
- **BankName:** Expert opinion has stated that the fraud relationship is a weak area.
- **CardHolderName:** This area is very difficult to use because some of them come from the bank as masked.
- **Pname, Psurname:** Passenger names and surnames can be utilized to compare the names and surnames of the flying passenger and the credit card holder.
- **FraudRiskScore:** This value is a score produced by the TKFraudFight application with the help of a relational algorithm. Therefore, it is excluded from this study.

4.1.2 Data Transformation

All steps in this stage were performed in Python using Scikit Learn. It consists of operations performed during runtime in the tests performed. These operations do not permanently affect the dataset; the dynamically transformed dataset is included in the classification stages.

4.1.2.1 One-Hot-Encoding

Deep neural networks (DNNs) and machine learning algorithms cannot directly utilize categorical variables. There will be a contradiction in the label values even after numbers are assigned to each category using the label-encoding approach, making it suitable for structural processing. At this stage, one can apply the one-hot-encoding approach [57], which creates a binary data field for every category. As a result, a clear relationship can be drawn between the categories and the class. 1524 binary features were produced using the one-hot-encoding procedure on 28 categorical variables in the dataset. The most distinguishing categories in each encoded data field were chosen, with the feature-selection stage applied to many binary fields.

Categorical fields that undergo one-hot-encoding are as follows:

- PaymentInstrument
- OfficeNumber
- Currency
- TDS
- MDStatus
- Installment
- PaymentServiceProvider
- SubChannel
- SaleResultCode
- ClientOS
- CardOriginCountry
- CardBrand
- CardProgram
- CardType
- OriginCountry
- DestinationCountry

- JourneyType
- FlightType
- Departure
- Arrival
- GMT
- Carrier
- Cabin
- CabinClass
- ResponseValue

Table 4.2 Number of Features Formed with One-Hot Vectors According to Fraud / NonFraud Balance

Fraud / Non-Fraud Balance	Feature Count with One-hot Vectors
1:1	1228
1:2	1311
1:5	1524
1:10	1704

4.1.2.2 Embedding Layer

Apart from one-hot-encoding, the embedding layer is another method used to handle categorical data fields by machine learning and deep learning algorithms. The embedding layer serves as a crucial element within deep learning frameworks, enabling the transformation of categorical or discrete data into continuous vectors. These vectors, called embeddings, effectively encapsulate the connections and resemblances among various categories or classes. All categorical data fields in the dataset that will be exploited in the decision mechanism are passed through the embedding process, and a new dataset piece emerges. This piece is combined with existing numerical fields and the dataset is made ready for classification algorithms to process.

In our study, two different methods were utilized when applying embedding layers to categorical areas. The first of these methods is to obtain a single transformed categorical dataset by including all categorical fields in the embedding process at once. This method determines the total embedding size by multiplying a fixed

embedding size with the number of categorical data fields. After the embedding process is applied to the categorical sub-dataset, a new dataset with the size of this embedding size is created. The process is completed by combining this obtained dataset with numerical data fields. This embedding method is described in Figure 4.1. The second embedding layer method we utilize in our study is to create a new dataset by subjecting categorical fields to the embedding process separately. This method calculates the embedding size value for each categorical data field, as shown in Equation (4.1). Categorical fields are converted into 2-dimensional arrays with the embedding size calculated for them. All created arrays are combined to create a new categorical dataset. Afterward, the embedding process is completed by combining the created categorical dataset with the existing numerical fields. The process of the second embedding method is described in Figure 4.2.

$$embeddingSize = min(number_of_unique_categories/2, 50) \tag{4.1}$$

Although the embedding layer method is a more costly method than one-hot-encoding, it can be preferred because it performs a more compact transformation in categorical data fields with high category numbers. The one-hot-encoding method produces very large vectors for data fields containing high categories and becomes impractical. In these scenarios, the embedding layer can be one of the most essential parts of the developed system. As can be seen in the results in Table A.1 and Table A.2, the embedding layer method showed very close success to the best results obtained with the one-hot-encoding method based on F1 Score. Additionally, if evaluated in terms of recall, the embedding layer showed a higher success rate of up to 1.3% on quantity and up to 2% on cost basis. The embedding layer is one step ahead of one-hot-encoding regarding fraud detection rate. As a result, although the success of embedding layer and one-hot-encoding is seen to be close to each other in our study, the embedding layer method can be preferred over the one-hot-encoding method since there are categorical data fields with a high number of categories in the aviation dataset.

The success and time performances of the embedding layer and one-hot-encoding techniques were compared. The success comparison can be seen in Figure 4.3, and the comparison of time performances was presented in Figure 4.4. Although the overall success results of the embedding layer and one-hot-encoding are very close, the embedding layer has higher fraud detection success, and the one-hot-encoding method has lower false alarm performance. In addition, when the operating

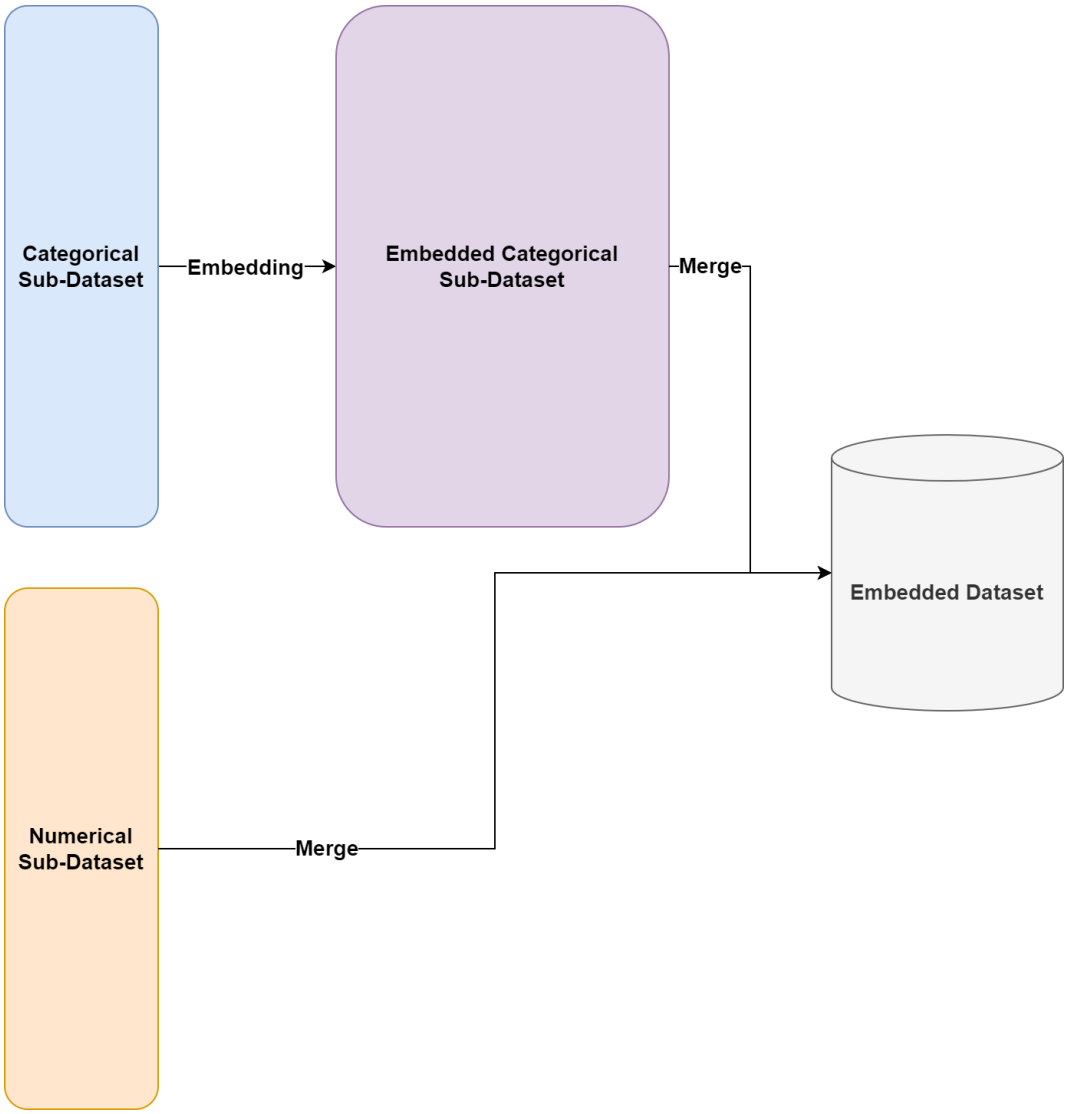


Figure 4.1 Single-Layer Embedding Process

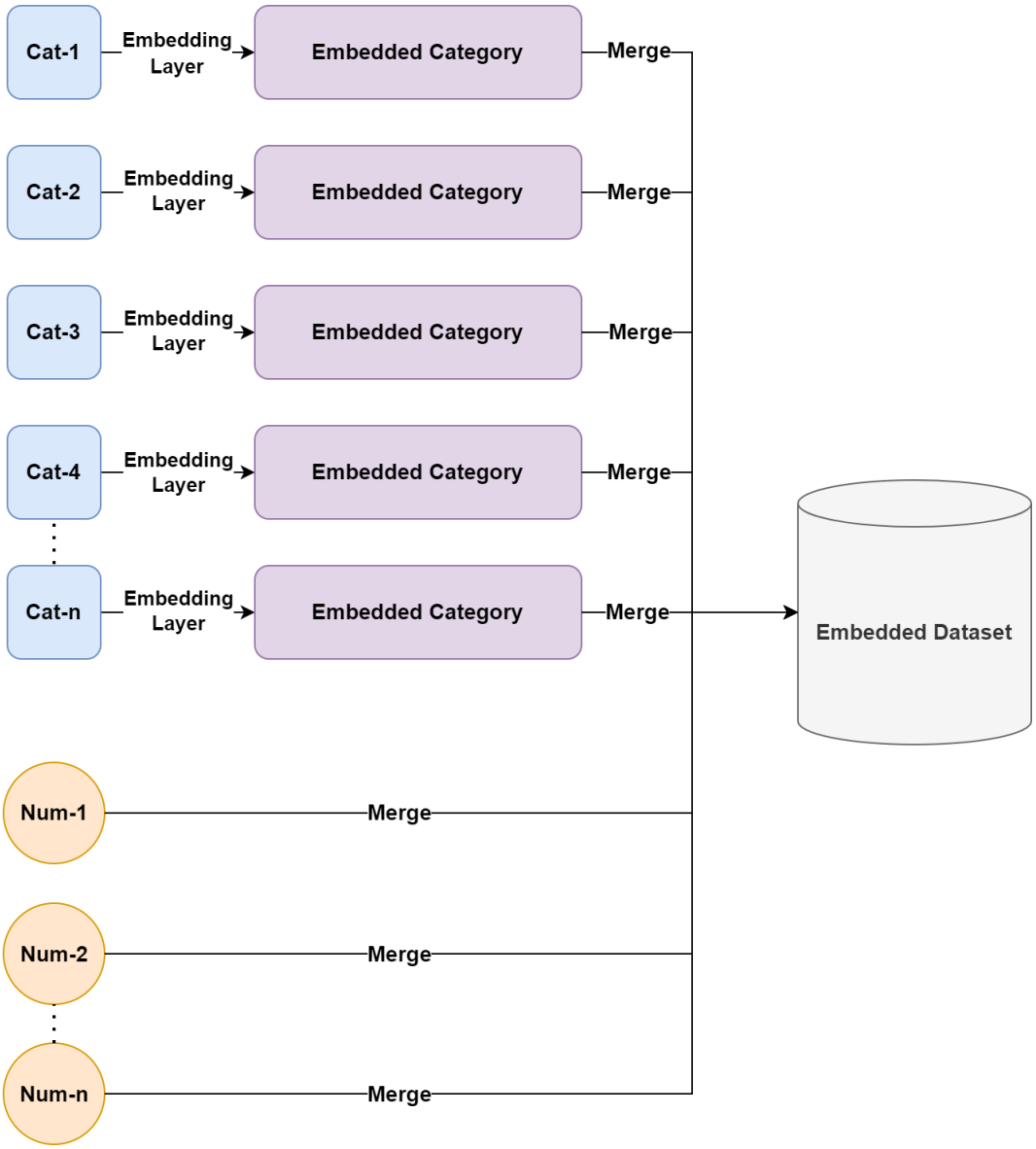


Figure 4.2 Multi-Layer Embedding Process

times are compared, both single-layer embedding and multi-layer embedding have higher operating times than one-hot-encoding. However, since the embedding layer method creates a smaller encoded dataset than one-hot-encoding, it can save processing times in the later stages.

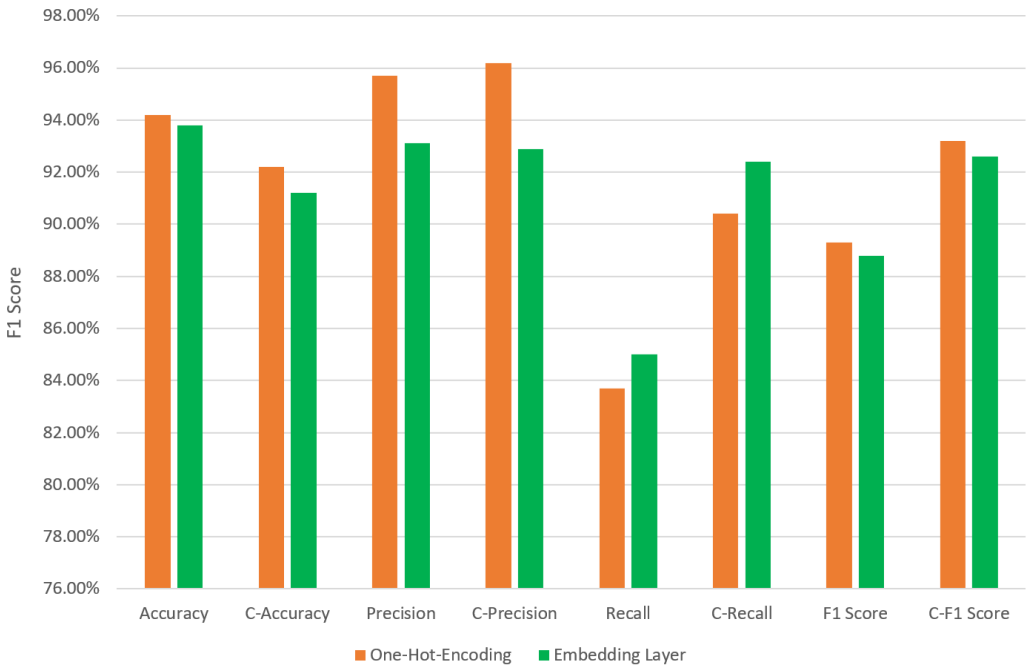


Figure 4.3 Comparison of embedding layer and one-hot-encoding success results

4.1.2.3 Min-Max Normalization

The "Amount" field, which is the only numerical field in the dataset, was subjected to min-max normalization. However, this field must undergo an additional exchange rate conversion process since there are amounts in different currencies. Otherwise, a great deal of misleading occurs between worthless and valuable currencies. For example, during the time period of this study, a flight worth 100 US Dollars corresponds to 4,210,500 Iranian Rials. Therefore, a significant deviation occurs. A single currency (USD) will be used as the basis in further stages, and all amounts will be converted to that unit. The min-max normalization formula utilized in this phase is presented in Equation (4.2).

$$\chi_{scaled} = \frac{\chi - \chi_{min}}{\chi_{max} - \chi_{min}} \tag{4.2}$$

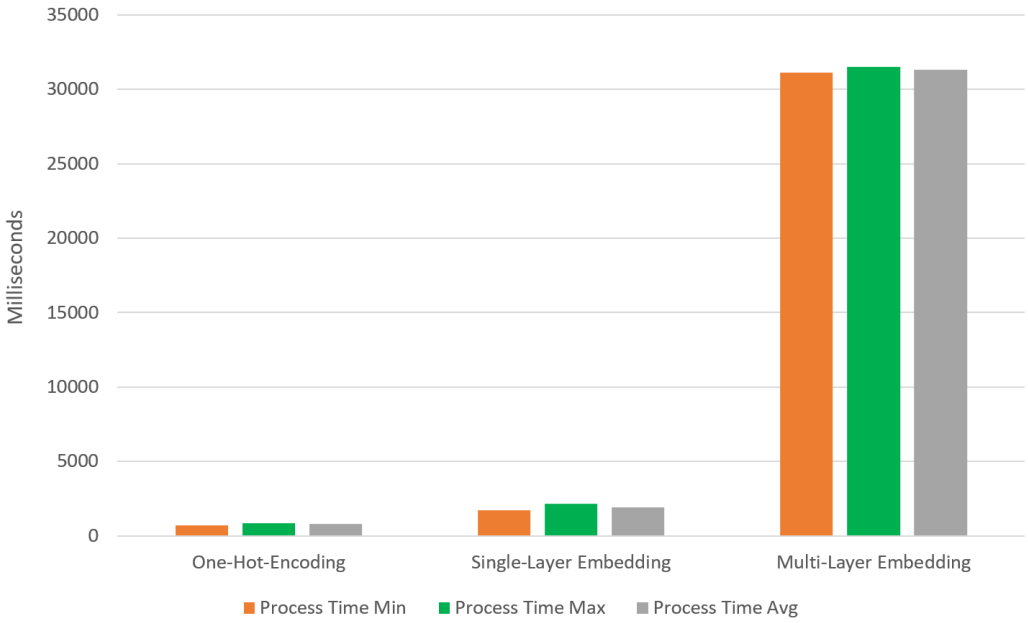


Figure 4.4 Comparison of embedding layer and one-hot-encoding time performances

4.2 Train and Test Set Distribution

Training and test sets were prepared using the dataset that went through the data pre-processing stages and took its final form. The ratio of the Training set was determined as 70% and the ratio of the test set was determined as 30%.

4.2.1 Random Distribution

This distribution method was implemented using the “train_test_split” method in the “model_selection” class in the Scikit Learn library. This method created training and test sets in a random order in the desired proportions by ensuring class balance on the entire dataset. Test results of randomly distributed train/test dataset were presented in Table B.1.

4.2.2 Chronological Distribution

This distribution method was implemented with the help of code written using the Pandas library. According to the "RequestDateTime" field, the first 70% of the data is divided into the training set and the next 30% into the test set. Test results of chronologically distributed train/test dataset were presented in Table B.2.

4.2.3 Monthly Sequential Distribution

It is the distribution type that is made after the dataset is divided on a monthly basis, with one month being the train set and the following month being the test set. This method was carried out to observe the relationship between consecutive months. However, it was observed that fraud patterns could not be detected because there was not enough temporal distance between the train and the test set. These tests were conducted at a fraud/legal ratio of 1:5, which we determined as optimal for success and closeness to reality. Test results of monthly sequential distributed train/test dataset were presented in Table B.3.

4.2.4 Comparison of Random, Chronological, and Monthly Sequential Distribution

In this experiment, we attempted to illustrate the significance of the chronological order of transactions. Applying the standard algorithm suite of our study, we examined randomly distributed, chronologically distributed, and monthly sequential distributed train/test sets. The train set for consecutive trials was the prior month, and the test set was the month after. Since fraud is a temporal problem, it is of great importance that the records in the dataset are chronologically ordered. For this reason, according to our theory, the success of tests performed on a randomly distributed train/test set will not be realistic. Table 4.3 provides specifics on the outcomes. Values highlighted in green represent the most successful results.

As our theory of temporality in the fraud problem stated, tests performed on a randomly distributed training/test set showed much higher success. We can see that across datasets generated chronologically and randomly, the F1 Score differs from 16% to 31%. However, as future fraud instances and zero-day attacks cannot be predicted, the findings of the random order dataset do not accurately reflect reality. Therefore, it is a known fact that maintaining the chronological sequence of train-test sets is the first rule of fraud detection; otherwise, a detection system may experience unexpectedly poor success rates in real-world situations. When the results of all distribution types are examined, it is observed that the precision successes are at similar levels. However, recall success was 2-3 times higher in random distribution. This shows that when the train test sets are randomly distributed, the fraud detection rate increases significantly because fraudulent transactions are recognized beforehand. On the contrary, no significant change was observed in false alarm rates.

Table 4.3 Performance Comparison of Train/Test Set Distribution Methods

Algorithm	Distribution Method	Accuracy	Precision	Recall	F1 Score
Decision Tree	Random	0.958	0.846	0.913	0.878
	Chronological	0.906	0.812	0.569	0.669
	Monthly Sequential	0.895	0.819	0.480	0.602
Random Forest	Random	0.972	0.922	0.906	0.914
	Chronological	0.900	0.908	0.450	0.602
	Monthly Sequential	0.892	0.895	0.401	0.602
SVM	Random	0.942	0.830	0.814	0.822
	Chronological	0.911	0.893	0.534	0.669
	Monthly Sequential	0.904	0.789	0.578	0.665
MLP	Random	0.958	0.825	0.942	0.880
	Chronological	0.909	0.869	0.534	0.662
	Monthly Sequential	0.899	0.837	0.489	0.615
Naive Bayes	Random	0.519	0.253	0.988	0.403
	Chronological	0.544	0.135	0.321	0.190
	Monthly Sequential	0.791	0.401	0.499	0.441

4.2.5 Monthly Cumulative Distribution

We utilized cumulative testing, in which the train set begins the previous month and gradually expands one month backward with each cumulation. The test set is the last month or the last two months. By doing this, we hoped to establish a timewise balance between the train and test sets and detect the saturation point at which F1 Score values are greatest [5].

In fraud detection systems, optimizing the ratio between the training set and the test set is of great importance. This balance to be established will determine the "Retraining" periods and the temporal distance between the training set and the test set. Therefore, in order to obtain this critical information in the system, cumulative experiments were carried out while creating the training set. In cumulative tests, the last 1 month and the last 2 months were used as the static test sets. On the contrary, the training set is dynamically grown backward by 1 month in each cumulation, starting from the previous month of the test set. These cumulative tests aim to find a saturation point in terms of F1 Score success in the temporal distance between the centers of the training and test sets. Thus, training/test balance can be achieved using the distance with the highest F1 Score values. The F1 Score results and saturation points of cumulative tests can be examined in Figure 4.5 and Figure 4.6, respectively. Cumulative test results based on algorithms are presented as 10 different test results in Table B.4, Table B.5, Table B.6, Table B.7, Table B.8, Table B.9, Table B.10, Table B.11, Table B.12 and Table B.13.

From this experiment, we can conclude that several important factors need to be followed in real fraud scenarios:

1. As the training set grows numerically and temporally, this does not indicate that success rates will constantly increase. Often, there is a saturation point in the temporal distance between the training and test sets. In addition, the more successful results of the experiments conducted with the 1-month test set also give clues about the "Retraining" periods in real scenarios. If a significant increase in success is achieved as the ratio of the test set to the training set decreases, it can be understood that the fraud evolution rate is high, and the "Retraining" period should be kept as short as possible.

2. After the detected saturation point, system success begins to decrease or fluctuate as algorithms begin to create patterns and fail to detect fraud cases. As mentioned in previous experiments, fraud is a problem that evolves in a flow. These saturation points can be considered as breaking points of fraud evolution flow for the relevant algorithms. The most successful algorithms in following this flow have been observed to be MLP and SVM, as they have lower deviations. However, in cases where the test set was 2 months old, SVM reached saturation with the 6-month training set and surpassed MLP, which reached saturation with the 13-month training set.

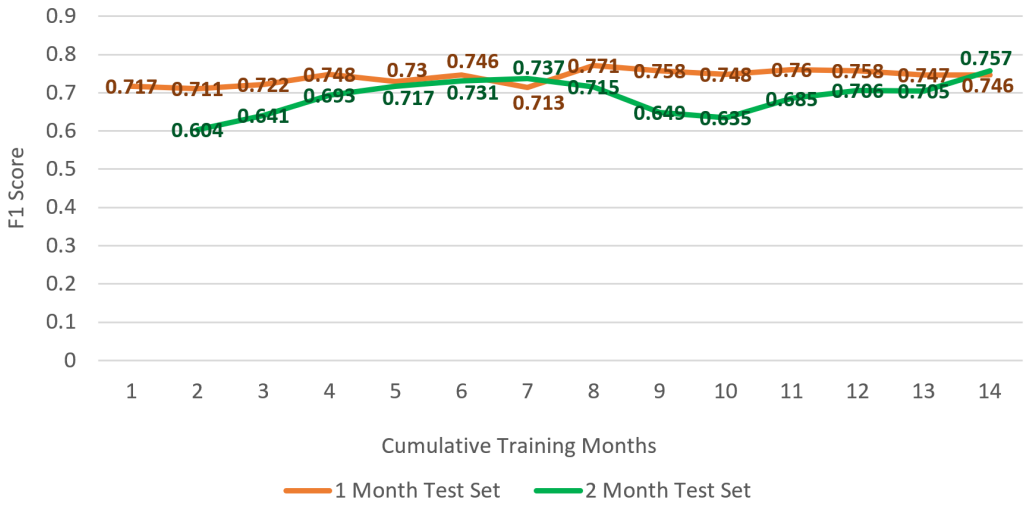


Figure 4.5 Comparison of 1 and 2-Month Test Sets in Terms of F1 Scores

The success rates obtained in cumulative tests showed that the saturation points for the 1-month test set of the SVM and MLP algorithms, which had the highest F1 Score success in almost all of the tests performed in general, were seen as the 8-month training set. Monthly datasets vary significantly in the number of transactions due to seasonal ticket sales differences. Therefore, when the saturation point was calculated in terms of the number of transactions, the test set's ratio was approximately 23.3%.

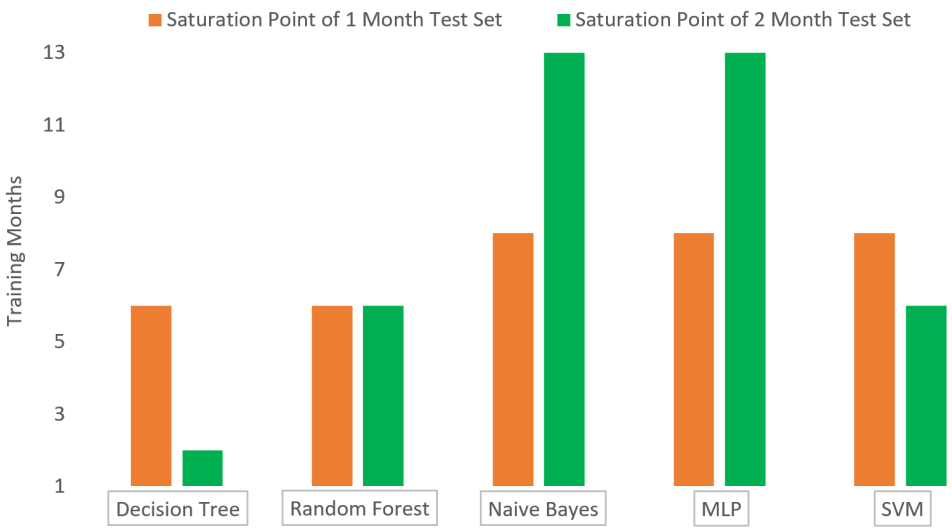


Figure 4.6 Comparison of 1 and 2-Month Test Sets in Terms of Saturation Points

4.3 Sampling

The dataset studied has the "Imbalanced Dataset" problem, as in most fraud datasets. This problem is caused by the fact that the number of non-fraud records in the dataset is much higher than the number of fraudulent records. To solve this problem in fraud studies, sampling methods were utilized to balance the fraud/non-fraud distribution. While some studies applying this method apply undersampling by reducing non-fraud records [3, 57–59], some studies apply oversampling by producing synthetic fraudulent records [60].

4.3.1 Balanced Random Undersampling (BRUS)

In this study, balanced random undersampling was performed to distribute homogeneously throughout the data with the help of an algorithm we wrote in Java. For this purpose, all fraud data was taken, and non-fraud records were randomly selected from each region of the data in proportion to that fraud data. Fraud/Legal ratios of 1:1, 1:2, 1:5, and 1:10 were determined, and downward sampling was performed at these rates for both the monthly and combined datasets. Considering that there are approximately 5500 fraud records, 5500 legal records were selected in a ratio of 1:1, 11000 in a ratio of 1:2, 27500 in a ratio of 1:5, and 55000 in a ratio of 1:10. Results of the first undersampling tests were presented in Table B.2. The 1:5 fraud/legal ratio has been determined as the optimal ratio in terms of success and closeness to reality and will be used as the default ratio in the later stages of the thesis.

The balanced random undersampling approach seeks to increase the

fraud/legitimate ratio by choosing between non-fraudulent records. However, It is not the same as the random undersampling techniques widely applied in library implementations in the literature. These characteristics are listed below:

- There is a guarantee that the dataset will have a uniform distribution of the data to be chosen from the majority class.
- The recently generated dataset is assured to be in chronological sequence.
- If some records have not been chosen yet, it is not possible to select others more than once.

The dataset has to be arranged in accordance with the transaction times in order to ensure these principles and the proper operation of the BRUS algorithm. First, the threshold to which the number of legal records should be lowered is determined by comparing the fraud/legal rate found in the dataset with the intended rate. The dataset is split into bins based on the ratio discovered in the previous stage in order to produce a balanced negative sample selection. A random negative sample is chosen for every new bin until the necessary fraud/legal ratio is obtained. Duplication is therefore avoided, and highly representative records with a balanced distribution across the sample are chosen. As a result, there will be less conditional variation between tests run on the same dataset. Algorithm in the Figure 4.7 presents the BRUS's essential phases, which we developed in our study and published in our article [61].

However, in random sample selection, it may be difficult to establish a pattern among the records in the new dataset, as some of the related records may be selected while others may be excluded from the sample. In addition, the exclusion of highly reliable records that may represent the "Legal" pattern from the sample may cause "False Alarm" (FP) rates to increase. Therefore, the profiling mechanism will be utilized in the following stages to support the BRUS algorithm. By using BRUS and profiled data fields together, a balanced undersampling process can be applied, and the loss of information in unselected records will be prevented.

The continuity of related records may be lost with undersampling approaches as most negative data are not chosen. Before sampling was applied, the complete dataset was profiled in order to solve this issue. As a result, a dataset has surfaced that is aware of prior connected records for both credit card and PNR as well as user profiles. Following the profiling phase, each transaction in the dataset is supplemented with statistical data from previous transactions. Through profiles, all

DETERMINATION OF BIN INTERVAL

$totalCount, fraudCount, binInterval \leftarrow 0, 0, 0$
for each $passengerActivityRecord \in dataset$ **do**
    if $passengerActivityRecord\ is\ fraud$ **then**
        $fraudCount \leftarrow fraudCount + 1$
    end if
    $totalCount \leftarrow totalCount + 1$
end for
$targetLegalCount \leftarrow fraudCount*(targetLegalRatio/targetFraudRatio)$
$binInterval \leftarrow totalCount/targetLegalCount$

NEGATIVE SAMPLE SELECTION

$counter, selectIndex \leftarrow 0, 0$
for each $passengerActivityRecord \in dataset$ **do**
    if $counter\ \%\ binInterval = 0$ **then**
        $selectIndex \leftarrow random.nextInt(downRatio)$
    end if
    if $counter\ \%\ binInterval = selectIndex$ **then**
        if $passengerActivityRecord\ is\ not\ fraud$ **then**
            $select(passengerActivityRecord)$
        else
            $selectIndex \leftarrow selectIndex + 1$
        end if
    end if
    $counter \leftarrow counter + 1$
end for

Figure 4.7 BRUS Algorithm

linked records that were not chosen were connected to the records that were selected using the BRUS algorithm.

4.3.2 Oversampling

Oversampling is a sampling technique typically used to avoid losing data in majority class records. Numerous oversampling methods exist, including SMOTE, ADASYN, and random oversampling. Nevertheless, if an oversampling approach is applied directly to the dataset, the total number of instances will exceed the original dataset. Because our dataset is very big, we did not choose to apply the direct oversampling strategy in our study.

4.3.3 Hybrid BRUS + Oversampling

The previous thesis follow-up steps applied the undersampling method to solve the imbalanced dataset problem. This thesis step aims to increase the representativeness

by taking more records from the dataset by adopting an approach in which undersampling and oversampling are utilized in a hybrid way.

In this study, firstly, undersampling was performed on the dataset at rates of 1:25 and 1:100. Afterwards, upsampling was applied to the minority classes of these sub-datasets obtained with fraud/legal ratios of 1:5, 1:2, and 1:1. For upsampling, "Random Upsampling", "SMOTE" and "SMOTENC" methods were selected and applied. The following sections include the results of these tests, as well as comparisons and conclusions drawn from the tests.

The following techniques are applied during the hybrid sampling mechanism's oversampling phase:

Random Oversampling: Using this technique, records from the minority class are copied and chosen at random to create synthetic records. A structure that can avoid imbalanced distribution and selection in the designed oversampling algorithm is critical to the detection system's performance. The results of Random Oversampling tests are given in Table C.1.

SMOTE: This well-known approach is employed to address the imbalanced dataset issue in several fraud-detection investigations [21, 62, 63]. To determine these records' k-nearest neighbors, minority class records are utilized. Next, a point is chosen between the selected record and the closest neighbor to create a synthetic record. It has been noted that the success rates are often quite low since this technique is not appropriate for processing categorical data fields, and our data collection has a large number of categorical data fields. The results of SMOTE tests are given in Table C.2.

SMOTENC: This is a SMOTE algorithm variant that also supports categorical data fields [64, 65]. When this approach is applied instead of the random oversampling technique, it has been noted that the low success rate in SMOTE is avoided, and more balanced results are produced. The results of SMOTENC tests are given in Table C.3.

Before the one-hot encoding phase, the sampling step was applied to minimize resource usage and shorten processing times for the SMOTE and SMOTENC algorithms. Label encoding was the only encoding utilized to prepare the dataset for sampling.

The test results of the hybrid method we designed by applying the BRUS and oversampling methods together were compared with the results we achieved by

applying only BRUS in Figure 4.8, Figure 4.9 and Figure 4.10. As a result of this comparison, when 1:5 fraud/legal ratio sampling results were investigated, some evaluations were made based on the F1 Score, and the following results were reached.

- In the hybrid method using random oversampling:
  - Significant decreases of 4-20% have been observed in F1 Score values in Decision Tree, Random Forest, and MLP algorithms.
  - As the amount of data used in the SVM algorithm increased, both precision and recall values increased steadily, and as a result, the F1 Score success rate increased up to 8.4%.
  - When the 1:100 dataset was exploited in the Naive Bayes algorithm, recall values increased significantly, and accordingly, the success in terms of F1 Score increased up to 28%. However, precision values could not accompany this increase in recall and remained at the same rate.
- Success rates in the hybrid method using SMOTE have generally been observed to be very low. When investigated, it was understood that the main reason was that the SMOTE algorithm was unsuitable for categorical data. The fact that our dataset contains a high percentage of categorical data fields caused the SMOTE tests to fail completely. To eliminate this problem, the SMOTENC algorithm, a variation of the SMOTE algorithm that can also handle categorical data, was utilized.
- In the hybrid method using SMOTENC:
  - A decrease in F1 Score was observed in Decision Tree and Random Forest algorithms. However, this decrease was not as high as the decrease experienced when Random Oversampling was applied. Decreases between 5-10% were observed in the Decision Tree and 7-14% in the Random Forest.
  - No significant increase or decrease was observed in MLP and SVM algorithms.
  - As a result of the significant increase in recall values and the slight increase in precision values in the Naive Bayes algorithm, a 28% increase in F1 Score was observed.
  - Although no big difference in success was observed in the changes between Random Oversampling and SMOTENC, it was observed that

with SMOTENC, more balanced changes occurred in all algorithms, and serious declines were prevented.

As a summary of our sampling experiments, the oversampling methods utilized were both costly and did not contribute significantly to the success of the BRUS algorithm. For this reason, the BRUS algorithm was determined as the optimal algorithm in terms of its contribution to operational cost and success.

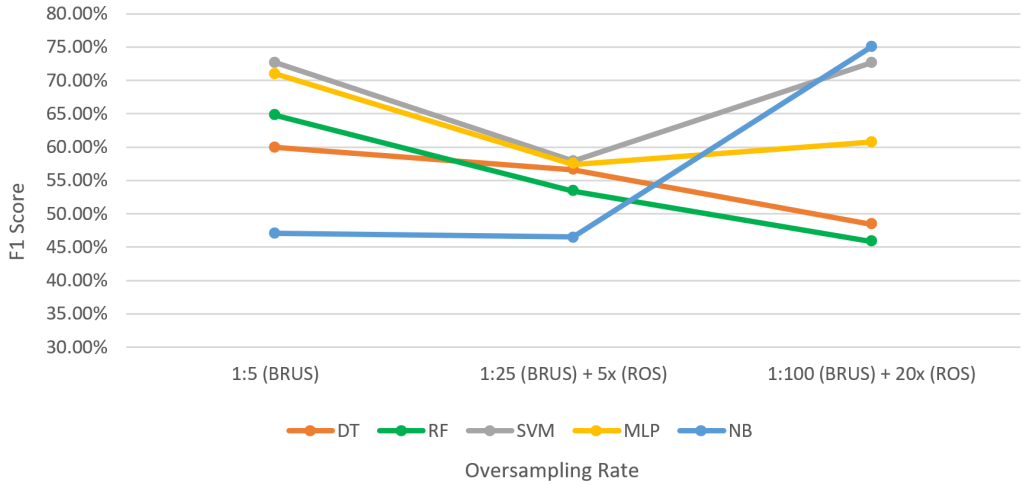


Figure 4.8 Comparison of BRUS and random oversampling with different fraud/legal ratios in terms of F1 Score

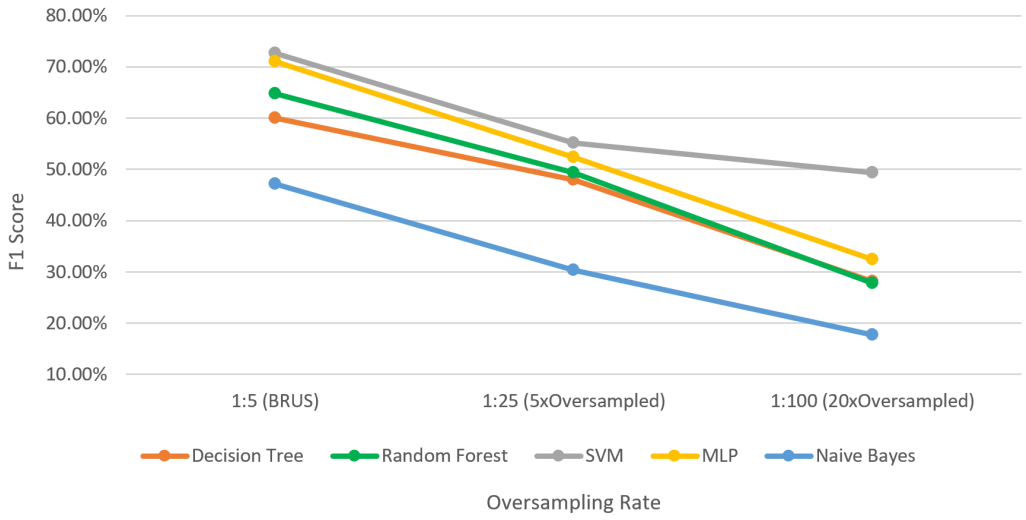


Figure 4.9 Comparison of BRUS and SMOTE with different fraud/legal ratios in terms of F1 Score

4.4 Feature Selection

We applied feature-selection methods on the accessible dataset to identify discriminative characteristics and save processing time. Three distinct methods

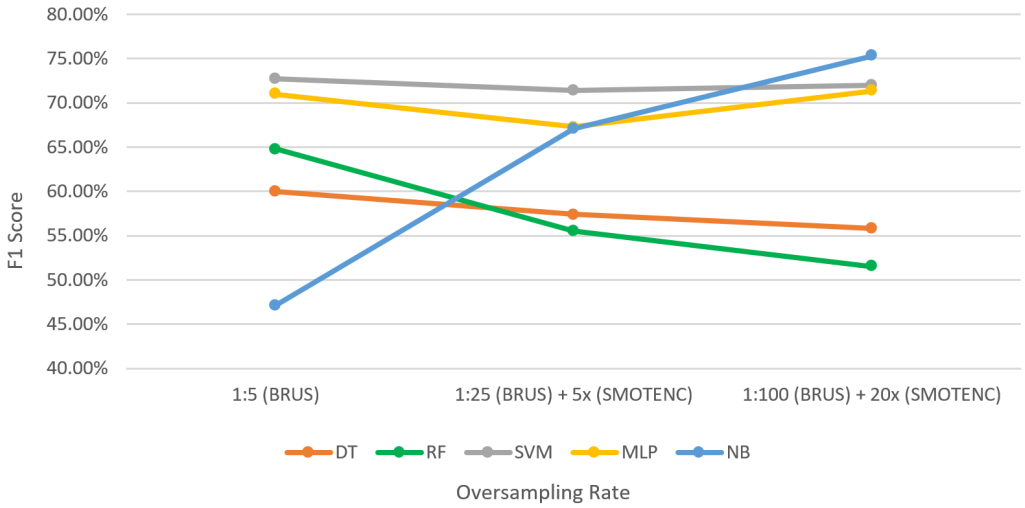


Figure 4.10 Comparison of BRUS and SMOTENC with different fraud/legal ratios in terms of F1 Score

were utilized in our investigation to pick features: CHI2, PCA, and AutoEncoder. These methods were applied to 1524 transactional binary features, 1 numerical transactional feature, and 21 extra numerical profiled features. Higher-resolution tests were conducted with less than 100 features since the required success scores could not be attained with more than 100 features. Following these tests, it was found that the tests conducted with feature numbers ranging from 20 to 75 produced the best results. The following feature-selection methods were then routinely applied in this range. In order to regulate how the findings alter outside of this range, we have also included some lower and higher characteristics in our trials.

Results of applied feature selection experiments with state-of-art machine learning algorithms were presented as CHI^2 results in Table D.1, Table D.2, Table D.3, Table D.4, Table D.5, PCA results in Table D.7, Table D.8, Table D.9, Table D.10 and AutoEncoder results in Table D.11, Table D.12, Table D.13, Table D.14, Table D.15. According to these results, it was observed that the most successful feature selection algorithms were CHI^2 and AutoEncoder methods. Although CHI^2 and AutoEncoder algorithms give similar results, CHI^2 was chosen as the default feature selection algorithm in the later stages of the thesis because it is more advantageous in terms of processing costs. Quantity-based and cost-based F1 Score success comparisons of the feature selection methods applied for the 55 features considered as the basis in our study are shown in Figure 4.11 and Figure 4.12, respectively. In the next stages, these feature selection methods were also applied in deep learning experiments, and the most successful results were observed.

In CHI2, PCA and AutoEncoder tests, the most successful machine learning

algorithms stood out as SVM and MLP. When feature selection methods were compared based on these algorithms, F1 Score successes were observed to be very close to each other. When Maximum and Average F1 Score values are calculated, conclusions can be seen below:

- For SVM algorithm:
  - CHI^2 has 77.8% max and 71.3% average F1 Score success
  - PCA has 74.8% max and 70.3% average F1 Score success
  - AutoEncoder has 77.7% max and 73.9% average F1 Score success
- For MLP Algorithm
  - CHI^2 has 71.4% max and 65.1% average F1 Score success
  - PCA has 73.7% max and 68.1% average F1 Score success
  - AutoEncoder has 73.2% max and 66.6% average F1 Score success

In addition, feature selection tests were performed again in the deep learning and profiling stages, and the results were observed. At this stage, the CHI2 algorithm, which gives similar results to AutoEncoder, gives much more successful results in the profile-based fraud detection system, as can be seen in the following stages.

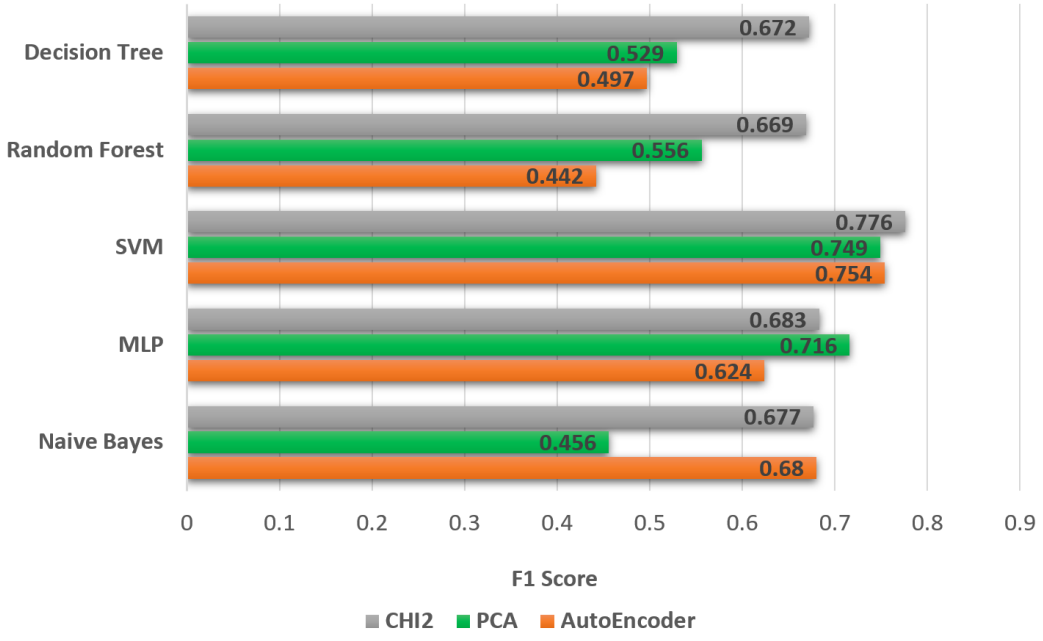


Figure 4.11 Comparison of feature selection methods in terms of quantity-based F1 Score

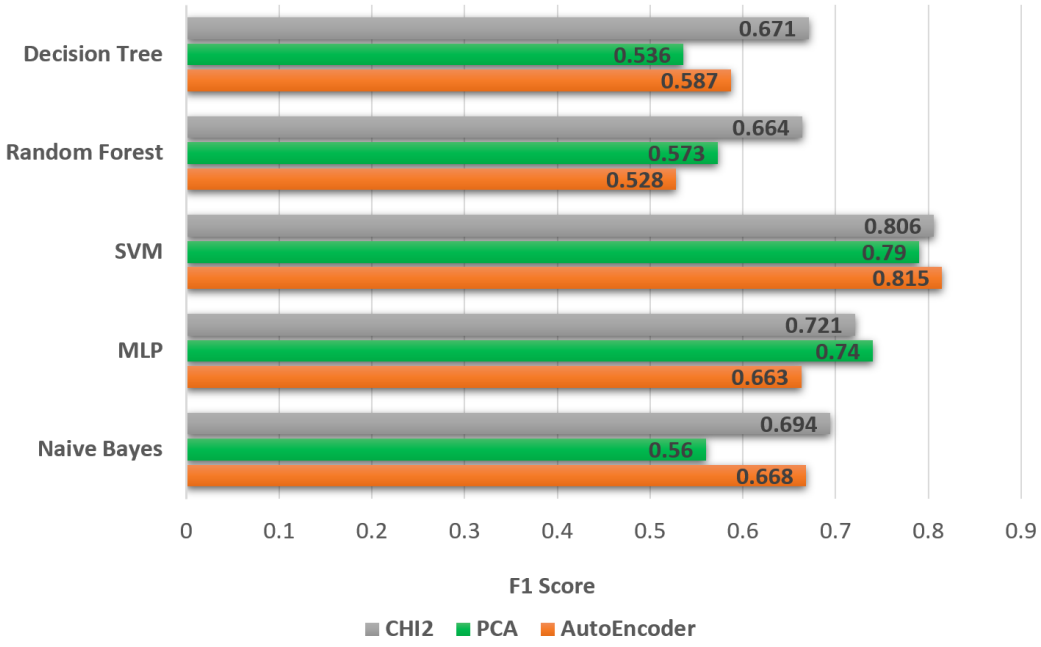


Figure 4.12 Comparison of feature selection methods in terms of cost-based F1 Score

5
COST-BASED MEASUREMENT SYSTEM

One of the main purposes of fraud systems is to prevent income losses resulting from illegal transactions. Therefore, the main expected function of these systems is to provide profitability optimization. However, in most studies, the success of fraud detection systems is calculated based on the number of transactions. Although metrics calculated based on the number of transactions reveal a successful system in detection results, they may block some legal transactions that can generate high income or allow a few fraudulent transactions that may cause very high financial losses [66–68].

In order to avoid this optimization error, a cost-based calculation must be made when considering the success of fraud detection systems. In addition, comparing cost-based success results with quantity-based results can reveal the monetary characteristics of fraud and legal transactions in the dataset. Thus, in addition to the transaction-based success criteria, cost-based success criteria were also applied in this study. Cost-based calculations are used to determine measures such as accuracy, precision, recall, and, most importantly, F1 Score, which are frequently employed in fraud detection. This work suggests four cost-based metrics: C-F1 Score, C-Accuracy, C-Precision, and C-Recall.

$$\begin{aligned} c &= currencyUSDRatio \times amount \\ cScore &= (c_i - min(c))/(max(c) - min(c)) \end{aligned} \tag{5.1}$$

First, the transaction amounts were converted using currency conversion rates to normalize them before these suggested metrics could be calculated. Afterward, the quantity column within the 0–1 interval underwent min–max normalization. Lastly, each transaction was included in the cost-based confusion matrix after being weighted with its c-score. Cost is now a factor in all previously determined measures using a cost-based confusion matrix. Equation (5.1) provides

the processes for obtaining the c-score, which are then utilized to generate a cost-sensitive confusion matrix.

5.1 Impact Analysis of Transaction Amount on Fraud Risk

While fraud is a financial problem from the perspective of companies, it is a source of profit from the perspective of those who commit fraud. Therefore, the main goal of fraudsters will be to maximize this profit. For this purpose, it can be predicted that fraudsters will try to make as much profit as possible from a single fraud transaction. If they choose the tickets they buy with stolen credit cards for as high a value as possible, they will increase their illegal earnings.

In line with these predictions, our study put forward the hypothesis that as the number of transactions made to purchase tickets increases, the risk of fraud will increase continuously or up to a point. All the data for the last three months were taken from the dataset to confirm this hypothesis, and histograms were created according to transaction amounts. By examining the obtained histograms, we tried to observe whether there was a relationship between the transaction amount and fraud risk.

In the pre-processing steps of the dataset, dollar conversions of transaction amounts in all currencies were already carried out. Therefore, first of all, the purchase amount scale in dollars was issued. This fee scale was divided into ten equal intervals, and the number of transactions falling within these intervals was calculated. Histograms were calculated based on both normalized values from 0 to 1 and their corresponding dollar values. Additionally, in this experiment, two different statistics were tried to be measured as follows:

- The percentage ratio of the number of fraudulent transactions in that range to the total number of transactions in that range.
- The percentage of fraudulent transactions in that range to the total number of fraudulent transactions in the dataset.

The fraud distribution histograms created and observed as mentioned above are presented in Figure 5.1, Figure 5.2, Figure 5.3 and Figure 5.4.

As observed from the histograms, the majority of fraudulent transactions occur in transactions with low fees. However, this is because there are many more low-priced purchases. When the percentages of frauds are examined, increasing the transaction

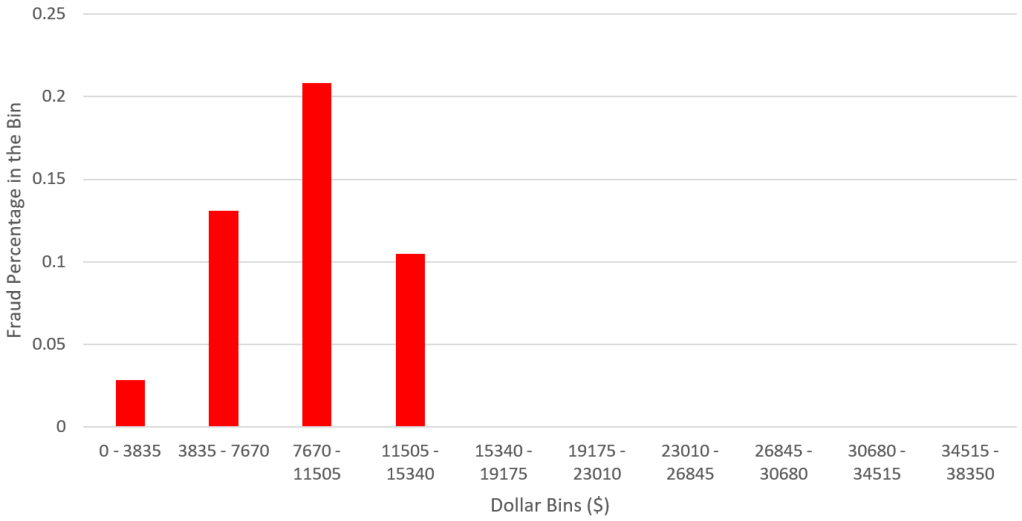


Figure 5.1 Fraud Percentages in Dollar Fee Ranges

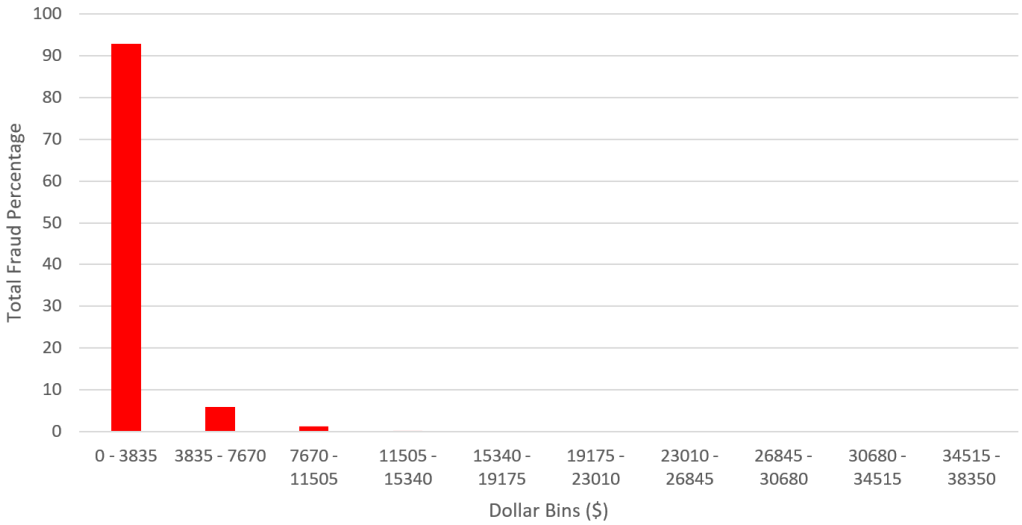


Figure 5.2 Ratio of Fraud Transactions in Dollar Fee Ranges to Fraud Transactions in the Whole Dataset

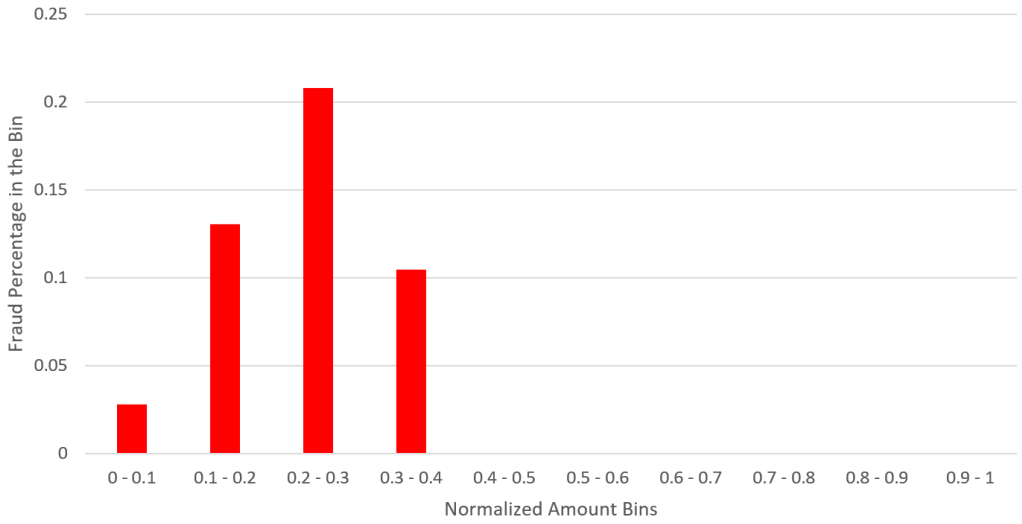


Figure 5.3 Fraud Percentages in Normalized Fee Ranges

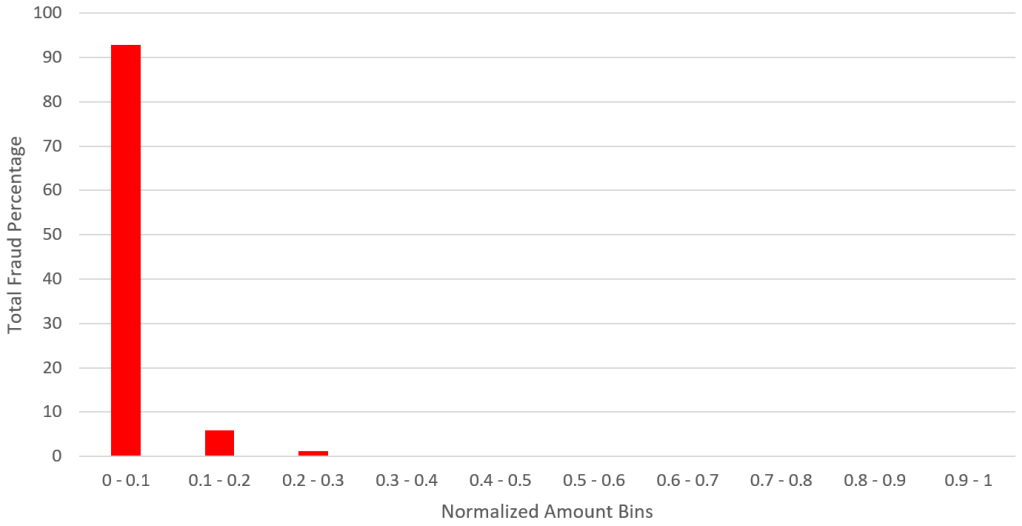


Figure 5.4 Ratio of Fraud Transactions in Normalized Fee Ranges to Fraud Transactions in the Whole Dataset

amount clearly increases the risk of fraud. This increase continued until a specific range and then started to decrease. When transaction fees increase, stricter rules are defined in the rule-based system, and the number of transactions at high fee levels is much less. As a result of these tests, it would be appropriate to develop a cost-based success measurement mechanism and plan a classification approach that also takes into account the fee level.

5.2 Currency Standardization

Sales transactions at the relevant airline are carried out in 16 different currencies. Since different currencies have different fraud risks, currency information is an essential and distinctive data field. However, since the value differences between currencies are very high, all transactions must be converted to a single currency. American Dollar (USD) was selected as the common currency, and the "Amount" fields for all transactions were converted to USD in accordance with the date they occurred.

5.3 Amount Normalization

After the currency standardization process was carried out, the amounts were normalized between 0 and 1 by applying the min-max normalization method to reduce the algorithms' calculation cost.

5.4 Cost-Based Confusion Matrix

After currency standardization and fee normalization procedures were carried out, a cost-based confusion matrix was created. In quantity-based measurement, when adding TP, TN, FP, and FN records, a standard increment of 1 is made per transaction. However, in cost-based measurement, when adding to the matrix, the standardized and normalized wage weight (c) of that transaction is added for each transaction. In this way, the confusion matrix is created entirely based on cost. The c values that make up the matrix are calculated for each process as in Equation (5.1).

5.5 Quantity and Cost-Based Success Comparison

This comparison was conducted to see the relationship between cost-based and quantity-based success rates in chronological distribution experiments performed on a dataset with a 1:5 fraud/legal ratio and 55 selected features. It has been observed that cost-based recall and accuracy rates can differ significantly from

transaction-based rates. This comparison experiment proved that traditional systems that focus on transaction-based success catch more fraud cases but prevent fewer financial losses. When Recall and C-Recall values are compared, it is seen that up to 12.5% of money loss can occur in systems that focus on transaction-based success. Therefore, cost-oriented performance should be considered in all commercial companies using fraud detection systems. If this comparison is evaluated from the perspective of companies, systems that focus on the number of transactions as a measure of success prioritize customer satisfaction. In contrast, systems that focus on the fee weights of transactions prioritize company profitability. Quantity-based and cost-based success rates are shown in Table 5.1. C-Accuracy, C-Precision, C-Recall, and C-F1 Score metrics are cost-based metrics.

Table 5.1 Quantity and Cost Based Transactional Results

Algorithm	**Accuracy**	**C-Accuracy**	**Precision**	**C-Precision**	**Recall**	**C-Recall**	**F1 Score**	**C-F1 Score**
Decision Tree	0.905	0.816	0.788	0.873	0.586	0.545	0.672	0.671
Random Forest	0.908	0.817	0.840	0.904	0.555	0.525	0.669	0.664
SVM	0.936	0.864	0.790	0.802	0.762	0.806	0.776	0.804
MLP	0.914	0.835	0.895	0.862	0.552	0.620	0.683	0.721
Naive Bayes	0.900	0.796	0.738	0.721	0.625	0.670	0.677	0.694
DNN	0.91	0.834	0.869	0.903	0.546	0.582	0.671	0.708

6
PROFILE-BASED FRAUD DETECTION

The fraud detection problem is a problem where chronological and historical data are of high importance, not only in the aviation field but also in other business fields. Therefore, having information about historical data in almost all fraud detection systems is critical in making the right decision. However, in most of the studies we examined in the literature, transactional datasets, that is, datasets that contain only information about the current transaction and consist of only self-aware records, were utilized. This situation does not coincide with the concept of being based on history, which is the most basic rule of fraud detection systems.

At this point, we carried out a profiling study in order to give historical representation ability to our transactional dataset and to reveal some business logic in the record in new areas. This study consists of three main parts, and new fields are added to our dataset in each part. Thanks to these added fields, the transactions in the dataset are now informed not only about themselves but also about the past transactions related to them. The detection system we designed on the dataset with new capabilities is called a profile-based fraud detection system.

In the profiling mechanism, new fields have been created to represent the history of the profiles involved in each transaction. The entire dataset was scanned chronologically to fill these fields with appropriate values. Credit card profiles were kept in memory during this scanning, and the current information in memory was used for each transaction. After each record was processed, the memory was updated, and the scanning continued. When the scanning process is completed, all new profiled fields have been successfully filled out, and the dataset becomes profile-based.

This study has made it possible to know potential passengers' history travels by incorporating them into the present transactional data and extracting them from historical statistics of certain airline customer profiles. Three fundamental profiles serve as the foundation for the suggested profile-based fraud-detection system for

aviation:

- Credit Card Profile
- PNR Profile
- Record Profile

The PNR profile and the credit card profile are distinct from one another while sharing many similarities, primarily in their respective application areas. While the PNR profile only includes the customer's tries, additions, and cancellations from the time they join the system and begin to request service until the conclusion of that service, the credit card profile's scope encompasses the entire dataset. The historical movement approach is advantageous for the credit card and PNR profiles generated in this study. New data fields are created in these profiles as the process moves forward. These fields are made up of statistical data computed by taking into account all pertinent historical transactions related to the profile value. First, records must be handled chronologically, just as real-world transaction flow, in order to make use of such historical movements. Therefore, each transaction's profile information only reflects that transaction's history. Each record in the dataset will thus have details about all of the prior transactions connected to them in addition to information about the record itself. On the other hand, the record profile uses a local strategy to make some concealed information inside the record visible. The purpose of this profile is to create new data fields that directly influence decision-making and to expose information that is buried inside the record. However, specific profiling modes are made to allow the profiles to be built and utilized in various combinations. It will be feasible to determine which profiles significantly impact the outcome by using various profiling modes. The methods section will provide a detailed explanation of these profiling modes.

We tried to demonstrate the usefulness of the suggested profiles in the ensuing subsections by providing several sample situations utilizing Table 6.1, Table 6.2 and Table 6.3 in our previous study on multi-modal profiling [61]. It is important to note that the "chargeback" column serves as the ground truth indicator for non-fraud and fraud cases. Additionally, all email addresses and passenger names presented in Table 6.2 and Table 6.3 are imaginary and generated by the authors solely to depict the situation.

6.1 Profile-Based Transformation

It is necessary to transform the dataset into a profile-based form in order to create a profile-based fraud-detection system. As a result, the dataset's transactional records will have historical data and be aware of the earlier connected records. Java was used to create the profile-based transformation module that was applied to the dataset. Since real-world circumstances must be followed during the profiling process, the dataset must be arranged chronologically. Since the credit card and PNR profiles are historical profiles, their conversion operations exploit maps containing statistical items. There is a statistical object and a map for every distinct profile value. From the first record to the last, the chronologically ordered dataset is scanned for each profile, and each record is subjected to a few standard procedures. The required profile is built and added to the dataset after the scanning cycle is finished. Algorithm in Figure 6.1, which we developed in our previous study [61], presents the fundamental phases of the profiling technique.

```
1:  for each passengerActivityRecord ∈ dataset do
2:      statistics_i ← profileMap[profilingObjectId_i]
3:      if statistics_i = null then
4:          initialize(statistics_i)
5:      else
6:          newStatistics ← recalculate(statistics_i)
7:      end if
8:      passengerActivityRecord[j] ← statistics_i[j]
9:      statistics_i ← newStatistics
10: end for
```

Figure 6.1 Profiling Algorithm

The targeted profile is established once the dataset has been scanned iteratively using these procedures for every entry. When the profiling scans are finished, the dataset is transformed into a profiled dataset since the newly created fields are integrated into the corresponding record throughout the scanning process. The dataset is prepared for sampling when the whole profiling procedure is finished.

6.2 Profiles

6.2.1 Credit Card Profiling

The historical usage of a particular credit card is the main focus of this profiling method. As shown in Table 6.1, it searches the complete history of the corresponding card. As a result, we gathered input about the card used to buy an airline ticket and determined its reliability. The following fields are taken from the

card's previous transactions as fundamental features.

- Number of chargeback transactions in history
- Number of accepted transactions in history
- Number of rejected transactions in history
- Number of transactions sent for manual investigation in history
- Number of domestic flights purchased in history
- Number of international flights purchased in history
- Number of distinct phone numbers used in history
- Number of distinct emails used in history
- Average amount spent to purchase ticket/service in history
- Maximum amount spent to purchase ticket/service in history

Two credit card profiling situations are shown in Table 6.1, showing the profiling map's current state at the time of each transaction. These examples show how, from a profile-based viewpoint, changes in spending patterns can be utilized to identify missed issues from a transactional perspective. Red regions indicate fraudulent transactions that cannot be discovered from the transactional viewpoint, whereas colorless regions reflect legitimate transactions. Green regions, on the other hand, represent fraudulent transactions that can be detected from the proposed perspective. Significant increases in spending habits were noted for the transactions shown in red. Nevertheless, the fraud-detection system is unable to identify these relative changes in transactional datasets. Being aware of the spending patterns of associated credit cards is one of the primary objectives of our credit card profile. A fraud-detection system will, therefore, be able to recognize the questionable discrepancies between recent transactions and historical data.

Fraudsters have utilized the stolen card in red-marked purchases. Some fraudsters attempt to buy expensive tickets in an effort to make the most money. As a result, the spending patterns associated with the card may abruptly alter once it is stolen. The profile-based method makes it easy to notice this change, even while the transactional technique is unable to capture it. When the transactional detection system does not correctly identify a red-marked transaction, a chargeback occurs. Subsequently, it is discovered that this card has been stolen, and despite

being fraudulent, the transactions indicated by the green checkmark under the red checkmark can be prevented. In conclusion, from the time the first fraudulent transaction happens until the chargeback feedback is received, the transactional method will fail to identify any fraudulent transactions involving a stolen card. Nevertheless, the profile-based method can detect all fraudulent transactions at the outset since it can consider transaction histories, statistical data, and modifications.

Table 6.1 Two example credit card profiling scenarios to emphasize the difference in fraud detection capability between transactional and profile-based approaches.

Credit Card Profile							
	Transactional Fields				**Profile Map**		
#	**Fraud Attempt**	**Approval Status**	**Chargeback**	**Amount**	**Approval Count**	**Chargeback Count**	**Average Expense Amount**
Scenario 1							
T1	No	Accept	No	$600	0	0	$0
T2	No	Accept	No	$300	1	0	$600
T3	No	Accept	No	$500	2	0	$450
T4	No	Accept	No	$600	3	0	$466.67
T5	Yes	Accept	Yes	$3000	4	0	$500
T6	Yes	Reject	No	$3500	5	1	$500
T7	Yes	Reject	No	$2500	5	1	$500
Scenario 2							
T1	No	Accept	No	$150	0	0	$0
T2	No	Accept	No	$200	1	0	$150
T3	No	Accept	No	$75	2	0	$175
T4	No	Accept	No	$125	3	0	$108.33
T5	No	Accept	No	$250	4	0	$112.5
T6	Yes	Accept	Yes	$2000	5	0	$140
T7	Yes	Reject	No	$2500	5	1	$140

6.2.2 PNR Profiling

PNR symbolizes a session that includes all of the user's transactions from when they first engage with the sales channel to the conclusion of their customer experience. Our goal with this profiling approach is to monitor a customer's transient behavior with respect to the relevant session. Our examination of fraud cases reveals that fraudsters often act differently from regular clients. A specific PNR number is utilized to engineer the following features.

- Number of chargeback transactions in history
- Number of accepted transactions in history

- Number of rejected transactions in history
- Number of transactions sent for manual investigation in history
- Number of domestic flights purchased in relevant PNR
- Number of international flights purchased in relevant PNR
- Number of distinct credit cards used in relevant PNR
- Number of distinct emails used in relevant PNR
- Number of distinct phone numbers used in relevant PNR

In Table 6.2, it is evident that many email addresses and phone numbers have been used in purchase attempts inside the same PNR for two distinct circumstances. In these situations, the transactional approach fails because it is unable to manage PNR transactions thoroughly. On the other hand, the profile-based method is quite good at spotting fake activity. Whereas the red regions represent fraudulent transactions that are missed by the transactional viewpoint, the green regions show where the suggested perspective may successfully identify fraudulent transactions. The fraud-detection system rejected the initial attempts due to the high risk associated with the provided phone number or email address. However, by approving the transaction on the red attempt, the system erred, and the fraud complaint resulted in a return of the cost. Visual examination of the transactions makes it quite evident that we are dealing with a fraudster attempting to go around the system. However, this suspicious behavior cannot be identified since there is no relationship between these transactions in the transactional dataset. A PNR profile was established and statistical data for each experiment conducted on the pertinent PNR was added to the records in order to guard against such vulnerabilities. A record is informed of all efforts in the PNR to which it belongs in this way.

Table 6.2 example PNR profiling scenarios to emphasize the difference in fraud detection capability between transactional and profile-based approaches.

PNR Profile											
	Transactional Fields					Profile Map				
#	Fraud	Approval	Chargeback	Email	Phone Number	Accept Count	Reject Count	Chargeback Count	Unique Email Count	Used Phone Numbers
Scenario 1										
T1	Yes	Reject	No	richard4355@big.co	0505	0	0	0	0	0
T2	Yes	Reject	No	andy77@anyco.com	0532	0	1	0	1	1
T3	Yes	Reject	No	john4@company.com	0555	0	2	0	2	2
T4	Yes	Reject	No	john5@company.com	0534	0	3	0	3	3
T5	Yes	Accept	Yes	safemail@safe.com	0535	0	4	0	4	4
T6	Yes	Reject	No	safemail@safe.com	0535	1	4	1	5	5
T7	Yes	Reject	No	gerard123@fake.com	0533	1	5	1	5	5
Scenario 2										
T1	Yes	Reject	No	suspicious1@mail.com	0531	0	0	0	0	0
T2	Yes	Reject	No	suspicious2@mail.com	0532	0	1	0	1	1
T3	Yes	Reject	No	suspicious3@mail.com	0533	0	2	0	2	2
T4	Yes	Accept	Yes	deceptive1@mail.com	0534	0	3	0	3	3
T5	Yes	Reject	No	deceptive1@mail.com	0534	1	3	1	4	4

6.2.3 Record Profiling

The second profile we created in our profiling study is Record Profiling. Although this profiling step is not based on history, only some business logic contained in the record and defined by business units are extracted into new areas. The new fields created at this stage are as follows.

- The number of minutes remaining until the flight at the time the transaction arrives in the payment system
- Whether the credit card holder's surname matches the surname of any passenger

Table 6.3 presents nine distinct record profiling scenarios that illustrate the process of identifying potentially dangerous trends by enhancing the significance of the date and name/surname alignment information in the record. Cases that defy natural law can be identified by comparing the cardholder and passenger data

and figuring out how long the remaining time is until the flight. Red regions indicate fraudulent transactions that cannot be discovered from the transactional viewpoint, whereas colorless regions represent valid transactions. Green regions, on the other hand, reflect fraudulent transactions that can be detected from the proposed perspective. The provided and chronological information appears appropriate in legal transactions. Nonetheless, some instances and discrepancies in fraudulent transactions might put the record in danger. To begin with, fraudulent transactions happen just before takeoff. In order to leave no time for the manual investigation of suspicious transactions, this occurrence may be utilized as a fraud tactic. Furthermore, in fraudulent transactions, there is a discrepancy between the surnames of the cardholder and any passengers. This could happen intrinsically when a ticket is bought for someone not from the family, but it also happens with every effort at fraud. These are two situations that are easily detectable through manual reviews. Nevertheless, the system can find it challenging to identify them effectively since relevant information is scattered over several locations.

Table 6.3 Nine independent record profiling scenarios to emphasize the difference in fraud detection capability between transactional and profile-based approaches.

Record Profile								
			Transactional Fields				Profile Map	
#	Fraud	Approval	Transaction Date	Flight Date	Credit Card Owner	Passengers	Remaining Minutes To Flight	Passenger Surname Matched
T1	No	Accept	21/12/2022 11:30	28/12/2022 19:00	Anthony Kent	Julia Kent	10,530	Yes
T2	No	Accept	15/02/2023 10:42	28/06/2023 19:00	Thomas Johnson	Madelyn Johnson	191,718	Yes
T3	No	Accept	06/08/2022 22:24	02/09/2022 06:00	Bill Morris	Bill Morris, Alice Morris	37,896	Yes
T4	Yes	Accept	28/12/2022 21:30	28/12/2022 23:20	Cris Jackman	Tony Stark, Andrei Fox	110	No
T5	No	Accept	10/02/2023 14:22	02/03/2023 07:15	Leonie Larson	Rosanna Larson	28,373	Yes
T6	No	Accept	28/09/2022 08:56	14/11/2022 10:30	Marianne Wallace	Danielle Wallace	67,774	Yes
T7	No	Accept	07/05/2023 11:43	25/05/2023 14:55	Findlay Abbott	Findlay Abbott, Valentina Abbott	25,728	Yes
T8	No	Accept	25/12/2022 22:24	01/01/2023 04:30	Jensen Erickson	Gavin Rose, Lukas Erickson	9006	Yes
T9	Yes	Accept	11/07/2023 02:38	11/07/2023 04:15	Laura Holland	Milan Molina, Adrian Brennan	97	No

6.3 Profiling Modes

Profiling modes are the set of guidelines that demonstrate how the created profiles are utilized in our investigation. By utilizing the pre-existing transactional information and the newly added data fields to the profiles, we developed different combinations of new datasets. We ran our basic tests using these datasets and identified the effective profiling modes. The following profiling modes were utilized for the performance analysis.

- Credit Card Profiling Mode: Transactional Information + Credit Card Profile
- PNR Profiling Mode: Transactional Information + PNR Profile
- Record Profiling Mode: Transactional Information + Record Profile
- Only Profiling Mode: All Profiles without Transactional Information
- Multi-Modal Profiling: Transactional Information + All Profiles

Profile-based deep learning tests were carried out using the multi-modal profiling method, which was determined as the most successful. In our thesis, tests were carried out with 55 features, the default number of features, and the results are presented in Table F.2.

The F1 Score gains provided by the profile-based fraud detection system compared to the transactional approach are shown in Figure 6.2 on a quantity basis and in Figure 6.3 on a cost basis. As clearly seen from these F1 Score gains, the most successful profiling method has emerged as Multi-Modal Profiling. It achieved a gain of 18% on a quantity basis and 21.7% on a cost basis. Unlike other classification algorithms, the SVM algorithm could not exhibit the expected performance with the profiling approach.

6.3.1 Credit Card Profiling Mode

This profiling mode is created by adding profiled credit card data fields to the transactional information. This new dataset contains both transactional fields and statistical credit card profile fields. The results of the tests performed with the Credit Card Profiling Mode are stated in Table E.1, Table E.2, Table E.3, Table E.4 and Table E.5.

6.3.2 PNR Profiling Mode

It is a profiling mode created by adding profiled PNR data fields to the transactional information. This new dataset contains both transactional fields and statistical PNR profile fields. The results of the tests performed with the PNR Profiling Mode are stated in Table E.6, Table E.7, Table E.8, Table E.9 and Table E.10.

6.3.3 Record Profiling Mode

In this profiling mode, meaningful data fields that may affect the fraud decision are extracted from the record. These fields obtained from the record are added to

the transactional information, and a new dataset is created. The results of the tests performed with the Record Profiling Mode are stated in Table E.11, Table E.12, Table E.13, Table E.14 and Table E.15.

6.3.4 Only Profiling Mode

All three credit card, PNR, and registration profiles are included in this profiling mode, but the transactional information is not utilized. A new dataset is created using only the profiled fields. The results of the tests performed with the Only Profiling Mode are stated in Table E.16, Table E.17, Table E.18, Table E.19 and Table E.20.

6.3.5 Multi-Modal Profiling

In this profiling mode, credit card, PNR, and record profiles are added to the transactional information to obtain a dataset consisting of all data fields. This dataset includes both transactional information and profiled data fields. The results of the tests performed with the Multi-Modal Profiling are stated in Table E.21, Table E.22, Table E.23, Table E.24 and Table E.25.

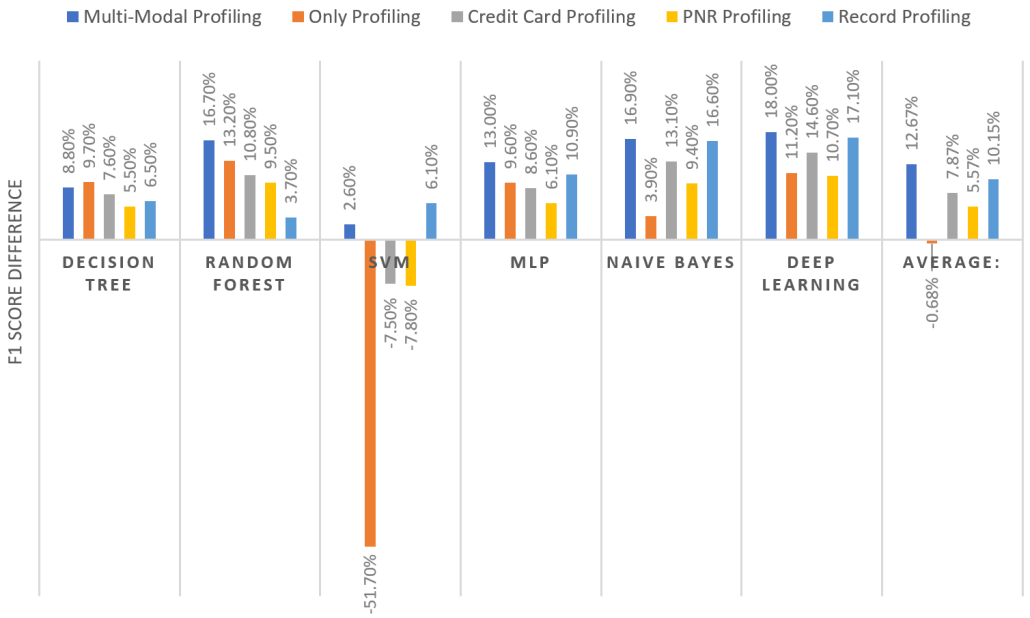


Figure 6.2 F1 Score Gains of Profile-Based Fraud Detection System Compared to Transactional Approach in Terms of Quantity

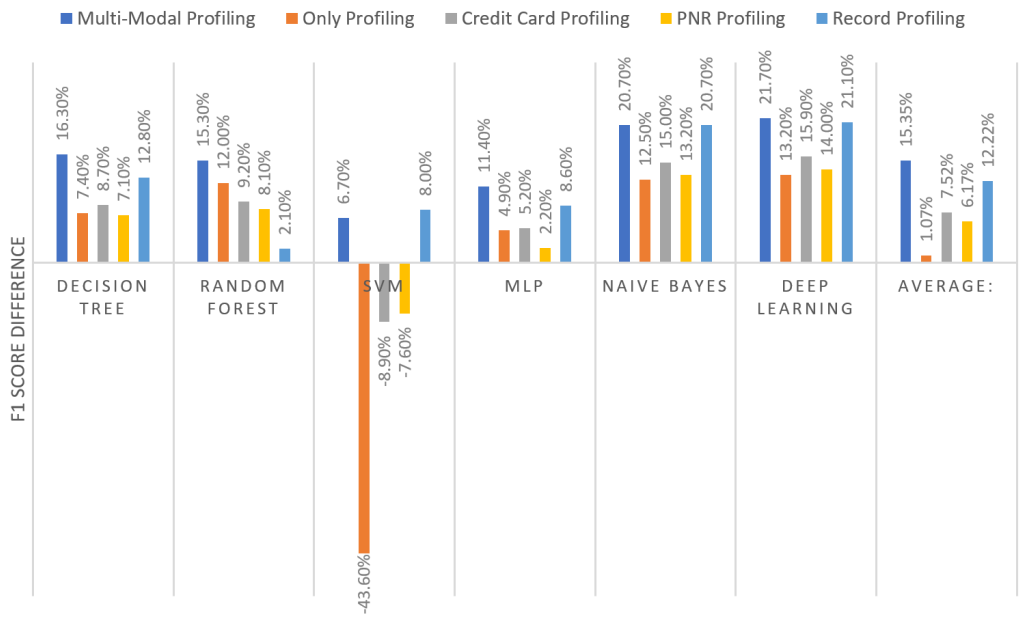


Figure 6.3 F1 Score Gains of Profile-Based Fraud Detection System Compared to Transactional Approach in Terms of Cost

6.4 Architecture of the Developed Fraud Detection System

The architecture of the developed fraud detection system is shown in Figure 6.4. In this architecture, the components that compose the fraud detection system are separated according to their functions and placed in order of data flow. The architecture and configuration of the deep neural network, which showed the highest success in terms of both quantity and cost in our study, is shown in Figure 6.5.

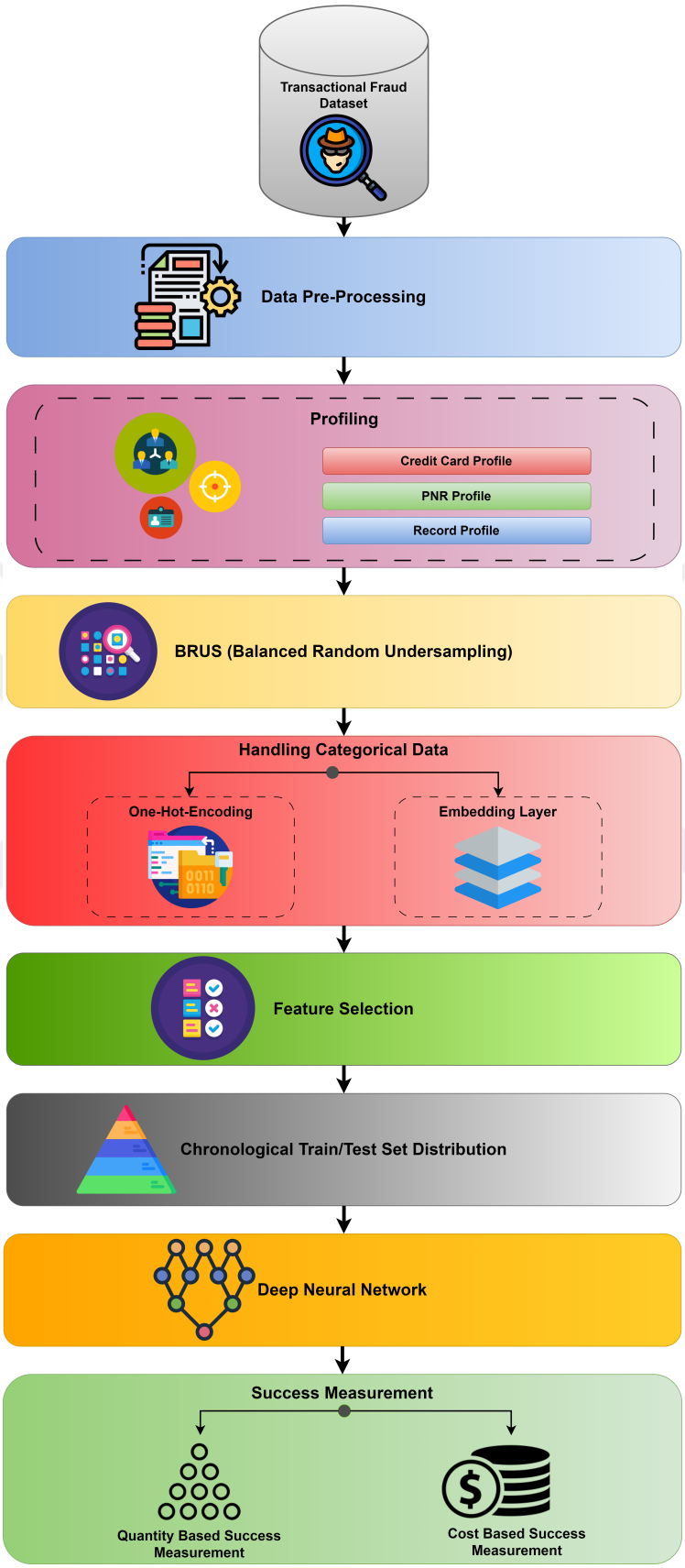


Figure 6.4 Architecture of the Developed Fraud Detection System

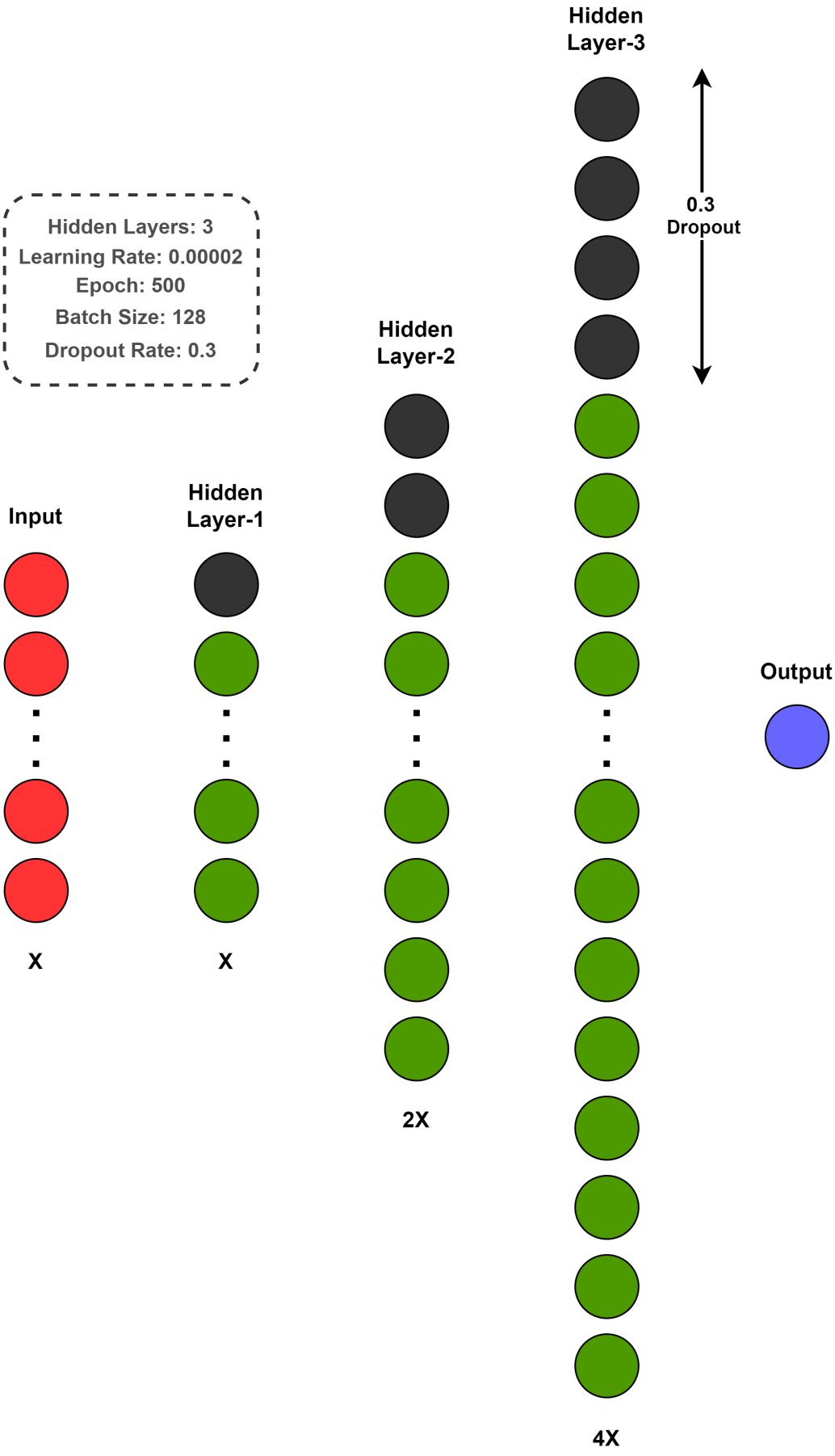


Figure 6.5 Architecture and configuration of the developed DNN

7
RESULTS AND DISCUSSION

In this section, we begin by providing a comprehensive overview of our experimental dataset. Subsequently, we present the preliminary findings obtained from employing cutting-edge learning algorithms, highlighting the significant impact of feature selection and sampling techniques on enhancing transactional information in terms of both quantity and cost. Furthermore, we delve into the successful implementation of the profiling mechanism that we introduced in our thesis study. Lastly, we present the Deep Neural Network (DNN) test results utilizing a combination of hyperparameters, showcasing the notable performance enhancement achieved by our proposed system. Throughout these experiments, we meticulously measure the state-of-the-art performance metrics, including accuracy, precision, recall, and F1 Score. Notably, we place particular emphasis on the F1 Score as it proves to be the most equitable metric for addressing imbalanced dataset problems. For a comprehensive understanding of the experimental setup, please refer to Table 7.1, which outlines the dataset configurations employed in these experiments.

Since the fraud-detection problem involves an imbalanced dataset, relying solely on the accuracy metric can be misleading when assessing true success. Imbalanced datasets typically consist of a large number of negative (legal) samples, leading detection systems to err on the side of caution and make negative decisions for all transactions. Consequently, accuracy results may appear remarkably high but fail to reflect reality accurately. To address this issue, it is crucial to evaluate precision and recall results in conjunction with each other, rather than relying solely on accuracy, in fraud-detection studies. To mitigate the disproportionate impact of precision and recall, the F1 Score combines both metrics and is employed as the primary measure for such problems. Additionally, the precision metric provides insights into the rate at which legitimate passengers are erroneously blocked, while the recall metric indicates the extent to which fraudulent transactions are detected. The F1 Score is a comprehensive indicator of fraud detection and false alarm prevention effectiveness.

Consequently, the F1 Score serves as the fundamental criterion for success in this study.

Our airline fraud-detection system utilized five cutting-edge machine learning algorithms in this phase. The machine learning algorithms listed below were selected from among the techniques that are commonly employed and contrasted in fraud-detection research in the literature [9, 59, 69–72].

- Decision Tree
- Random Forest
- Support Vector Machines
- Multilayer Perceptron
- Naive Bayes

Deep learning methods, which are the main subject of the thesis, were applied to be applied on the transactional dataset, which was subjected to various data processing, transformation, and logical operations mentioned before and reduced to a 1:5 ratio with the BRUS algorithm using the Keras library in Python language. The primary purpose of these experiments is to create deep network architectures suitable for the dataset studied and to improve the success of fraud detection by calibrating the hyperparameters of these networks to achieve maximum success without a significant increase in the false alarm rate.

Unless stated otherwise, all classification steps in our thesis were carried out in a 70% training and 30% test set chronological distribution, with a class ratio of 1:5 obtained with the BRUS algorithm, which is the most successful fraud/legal ratio, and 55 features, which is the most successful number of features.

7.1 Performance Improvements on Transactional Dataset

In our previous publication [5], we conducted various tests and enhancements to attain optimal success on the transactional dataset. Throughout these experiments, we explored different feature selection mechanisms, measurement metrics, and train/test set organization methods to enhance fraud detection performance in the aviation industry. In this stage, we examined different sampling techniques using the airline ticket sales dataset, including our proposed BRUS algorithm and deep learning approaches. The outcomes are presented in this subsection. It has been

Table 7.1 The details of our dataset and train/test setup.

Dataset Configuration		Value
	Fraud/Legitimate Ratio	1:5
	Dataset Time Range	15 Months
BRUS	Number of Fraud Number of Legitimate	5512 27,578
Oversampling × 5	Number of Fraud Number of Legitimate	27,560 137,890
Oversampling × 20	Number of Fraud Number of Legitimate	11,240 551,560
#Features	Transactional dataset CC Profile PNR Profile Record Profile	41 10 9 2
Distribution	Type Train/Test Ratio	Chronological 70/30

observed that there are significant limitations in improving the effectiveness of a transaction-based fraud detection system to prevent financial loss for airline companies and ensure a satisfactory sales experience for customers. Despite extensive efforts in improvement and optimization, the precision, recall, and particularly the F1 Score metrics did not reach satisfactory levels in terms of both transaction volume and cost considerations.

The success rates obtained after implementing the feature-selection process can be found in Figure 7.1. Additionally, Figure 7.2 presents the newly proposed cost-based measurement metric results. These figures initially display the test results using shallow and deep learning algorithms after performing only the necessary cleaning operations on the dataset. Upon examining these figures, it is evident that the decision tree algorithm did not experience any significant change after the feature selection was applied. On the other hand, the naive Bayes algorithm exhibited remarkable performance improvement following the feature selection. The primary reason for the unexpectedly poor performance of Naive Bayes before the pre-processing step can be attributed to the correlation between the provided features. The performance of this basic algorithm was significantly enhanced by both the feature selection process and the inclusion of engineered features. Except for the decision tree algorithm, a noticeable increase ranging

from 2.1% to 48.7% was observed in the quantity-based F1 Score. The cost-based evaluation demonstrates that the quantity-cost success balance is maintained, with the support vector machine (SVM) algorithm emerging as the most successful. Based on these observations, it becomes evident that the feature-selection process plays a substantial role in addressing the fraud-detection problem within the aviation domain. Even the initially poor success results obtained using the naive Bayes algorithm improved after the application of feature selection, reaching the same level of success as other algorithms. To further investigate the impact of the feature-selection process, tests were conducted using the CHI^2 and PCA algorithms. Among the classification algorithms, SVM proved to be the most successful, while CHI2 emerged as the most effective feature-selection method. On average, the highest success across all algorithms was achieved with 55 features. Furthermore, a significant increase of up to 48.7% was observed in the Naive Bayes test results. Detailed test results can be found in our previous research [5].

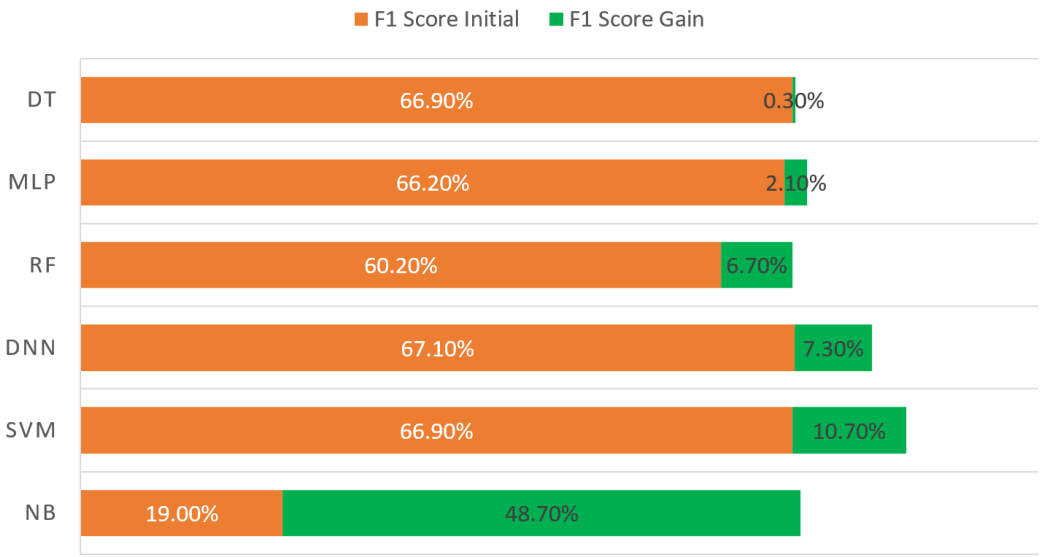


Figure 7.1 The initial fraud-detection performance of state-of-the-art learning algorithms and the impact of the feature-selection process in terms of quantity

Sampling is an essential technique for enhancing the effectiveness of resolving imbalanced dataset issues. In our previous sampling experiments [5], we exclusively utilized the random undersampling approach and identified a 1:5 fraud/legal ratio as the most successful configuration based on the test outcomes. The results of our implementation of hybrid oversampling methods to tackle imbalanced datasets, along with their comparison to the BRUS algorithm proposed in our study using the F1 Score, are presented in Figure 7.3 and Figure 7.4. The hybrid oversampling algorithms were employed in conjunction with the BRUS algorithm. Initially, we obtained unsatisfactory results as the SMOTE

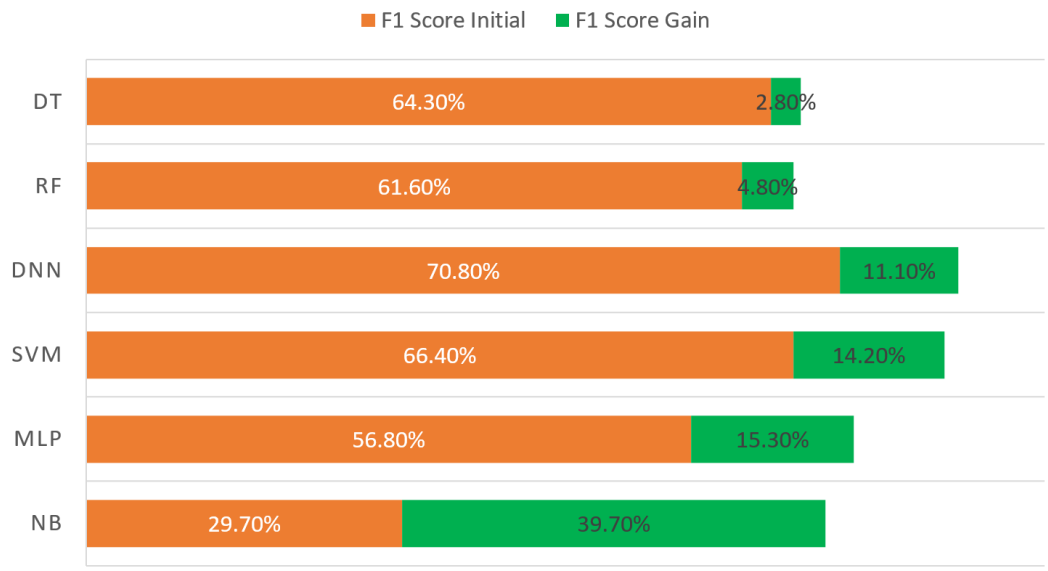


Figure 7.2 The initial fraud-detection performance of state-of-the-art learning algorithms and the impact of the feature-selection process in terms of cost

algorithm proved unsuitable for handling categorical fields. Consequently, the SMOTE algorithm was excluded, and its results are not included in this section. A comparison between the SMOTENC and Random Oversampling algorithms revealed that SMOTENC exhibited a higher success rate of 20.6% and 10.5% for the Naive Bayes algorithm and MLP algorithm, respectively. However, overall, oversampling, being a more resource-intensive sampling method, did not demonstrate a significant improvement compared to the BRUS algorithm. Therefore, in the subsequent stages of our study, we proceeded with the BRUS algorithm.

In summary, SVM demonstrates superior performance compared to other algorithms when evaluated based on quantity. Conversely, DNN has achieved remarkable success in mitigating financial losses. Additionally, our observations indicate that the F1 Score for both quantity-based and cost-based measurements reaches a threshold of 80% in aviation's transaction-based fraud detection.

7.2 Profile-Based Fraud Detection Results

A profile-based fraud detection method was devised to overcome the restrictions placed by the transactional information and reveal historical passenger behavior patterns connected to previous transactions. This method indirectly improves fraud detection's overall effectiveness. In this subsection, we present the test findings of the suggested aviation profiles and explain how they add to the current transactional perspective.

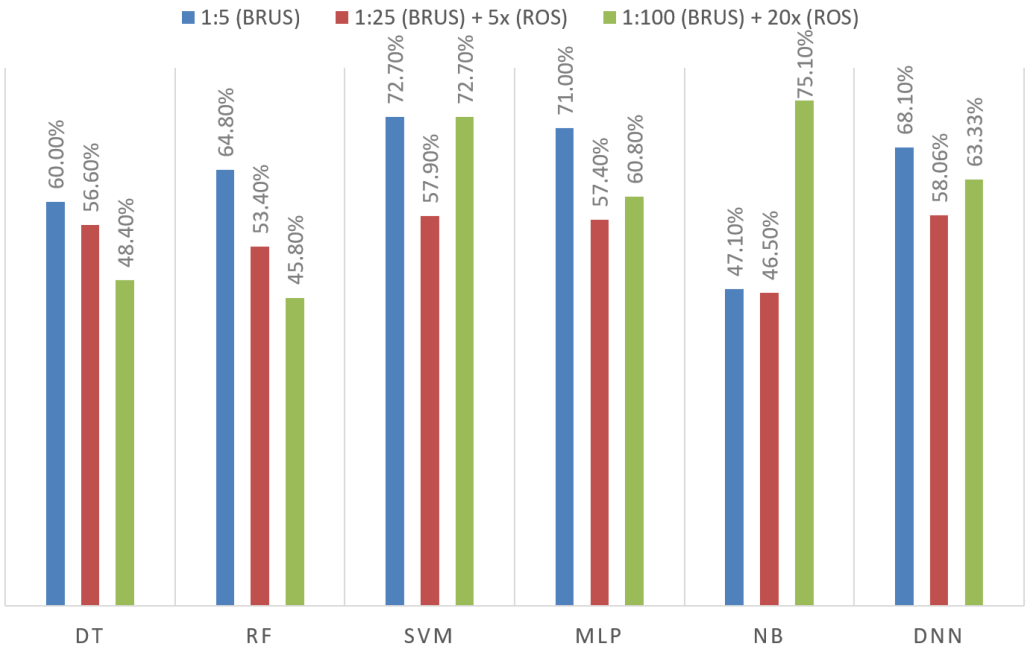


Figure 7.3 Balanced random undersampling (BRUS) vs. BRUS + random oversampling (ROS) in terms of F1 Score

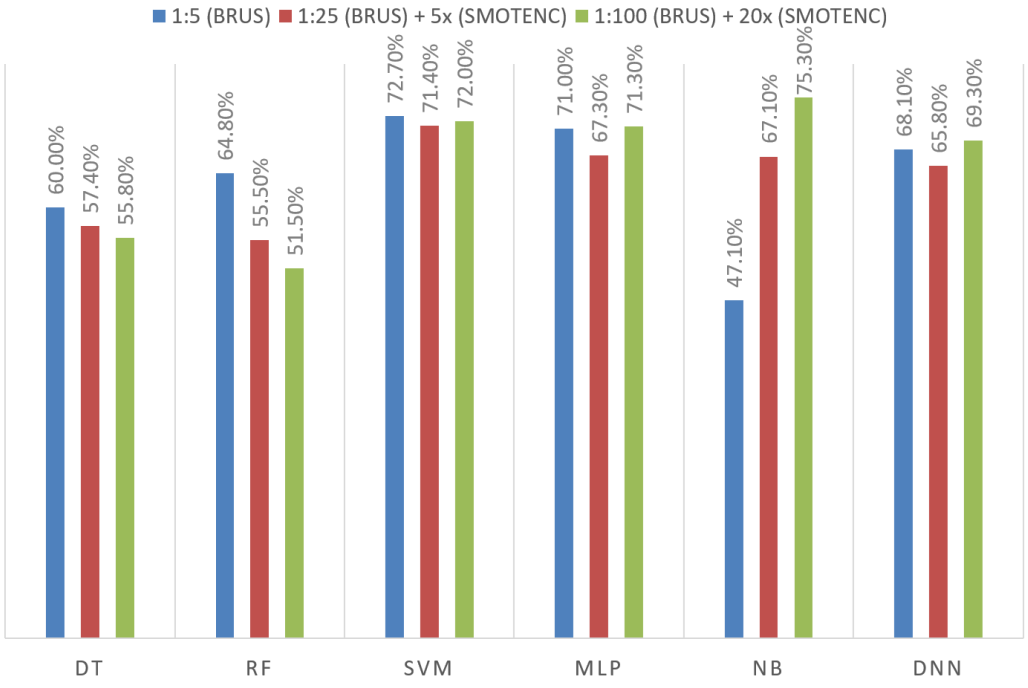


Figure 7.4 Balanced random undersampling (BRUS) vs. BRUS + SMOTENC in terms of F1 Score

In the previous stages, deep learning tests were carried out at the entry level on the profile-based dataset, and the results were observed. At this stage, a more systematic and expanded experimental set was created. While creating this experimental set, different systematic combinations were exploited on the architecture and hyperparameters of the deep neural network. The created experimental set is seen in detail in Table F.1, and the results of these tuning tests are presented in Table F.2.

Multi-modal profiling has emerged as the most effective profiling mode overall, as seen in Figure 7.5 and Figure 7.6. According to our research, the transactional fraud-detection system's maximum F1 Score success was achieved by the DNN at 81.9% in terms of cost and by the SVM algorithm at 77.8% in terms of quantity. On the quantity and cost bases, however, the Naive Bayes algorithm with multi-modal profiling performed with a success rate of 88.2% and 92.2%, respectively. The F1 Score success rates for the applied methods increased by 7 points to 17 points, which motivated us to publish our profiling methodology in the literature. Although each profile had a role in this increase, record, and credit card profiling had a significant impact. However, we also found that using simply profiled data fields behind the multi-modal profiling technique and excluding transactional information is not an option for an effective fraud detection system. Comprehensive testing was carried out employing the multi-modal profiling mode after it was shown to be the most effective one in terms of both quantity and cost basis.

In light of the newly introduced profiles, we decided to rerun the feature-selection tests by incorporating the Autoencoder alongside the existing features, as it complements the DNN. Surprisingly, contrary to our expectations, the CHI^2 algorithm still outperformed the Autoencoder, yielding a 2.4% higher success rate, and it also outperformed the PCA by 7.7% as can be seen in Figure 7.7. Consequently, we selected the CHI^2 algorithm for the feature-selection process in the upcoming DNN tuning tests. Subsequently, we conducted experiments using the new feature set derived from both transactions and profiles, as depicted in Figure 7.8. Among the algorithms tested, the naive Bayes algorithm exhibited the highest success rate, achieving an F1 Score of 88.2%. On the other hand, the random forest, DNN, and MLP algorithms achieved success rates of 86.6%, 85.3%, and 84.4%, respectively. The promising results obtained from the DNN motivated us to investigate the hyperparameters further and enhance the initial outcomes. To this end, we created an experimental set and performed various tests. The details of this experimental set, along with comprehensive results encompassing accuracy, precision, and recall, can be found in Table F.2. Experiment 45 yielded the highest F1 Score values in terms of both quantity and cost. The quantity-based success rate reached 89.3%, while the cost-based success rate reached 93.2%. Notably,

the DNN outperformed other algorithms by leveraging multi-modal profiling and hyperparameter adjustment. A thorough analysis of the results revealed that the network architecture and learning rate contributed most significantly. It is worth emphasizing that fine-tuning the hyperparameters is an essential step for fraud detection systems.

During the tuning tests, our objective was to assess the influence of hyperparameter calibration on the system's overall success. To achieve this, we systematically altered one hyperparameter at a time while keeping all other test parameters constant and subsequently analyzed the outcomes. Through this examination, we aimed to determine whether hyperparameters and architectural features directly impact the results. In Figure 7.9, we investigated the impact of varying the number of epochs between 50 and 500, then carefully examined the corresponding test results. However, our observations revealed that solely modifying the epochs did not directly affect the success. It was found that the success rate could be improved by adjusting the epoch numbers based on the learning rate value.

In the calibration tests for batch size, the outcomes were evaluated by progressively increasing the batch size by a factor of 2, ranging from 32 to 512 in each trial. Figure 7.10 illustrates that despite some fluctuations, the F1 Score decreased 2.9% in terms of quantity and 4% in terms of cost due to the augmentation of the batch size. This set of calibration tests demonstrates that superior performance can be attained with a smaller batch size. Nevertheless, striking an optimal balance between batch size and training time is crucial to ensure the training cost remains within an acceptable range.

Various factors, including the batch size and epoch number, influence the impact of the learning rate hyperparameter on the outcome. Several calibration tests were conducted to assess this effect while maintaining the other variables constant. The findings of these tests are illustrated in Figure 7.11. Notably, as the learning rate decreased from 0.01 to 0.00001, there was a significant improvement in the F1 Score, with an increase of 5.4% in terms of quantity and 8.8% in terms of cost. These results emphasize the crucial significance of intricate and comprehensive patterns in the realm of aviation fraud detection.

In our study, we developed a DNN to investigate the impact of dropout on the hidden layers. To evaluate the effectiveness of dropout, we compared the performance of the DNN with and without dropout in Figure 7.12. The results of this comparison revealed that incorporating dropout in our fraud detection system had a positive influence on its success. Specifically, we observed an improvement of 3.2% in the

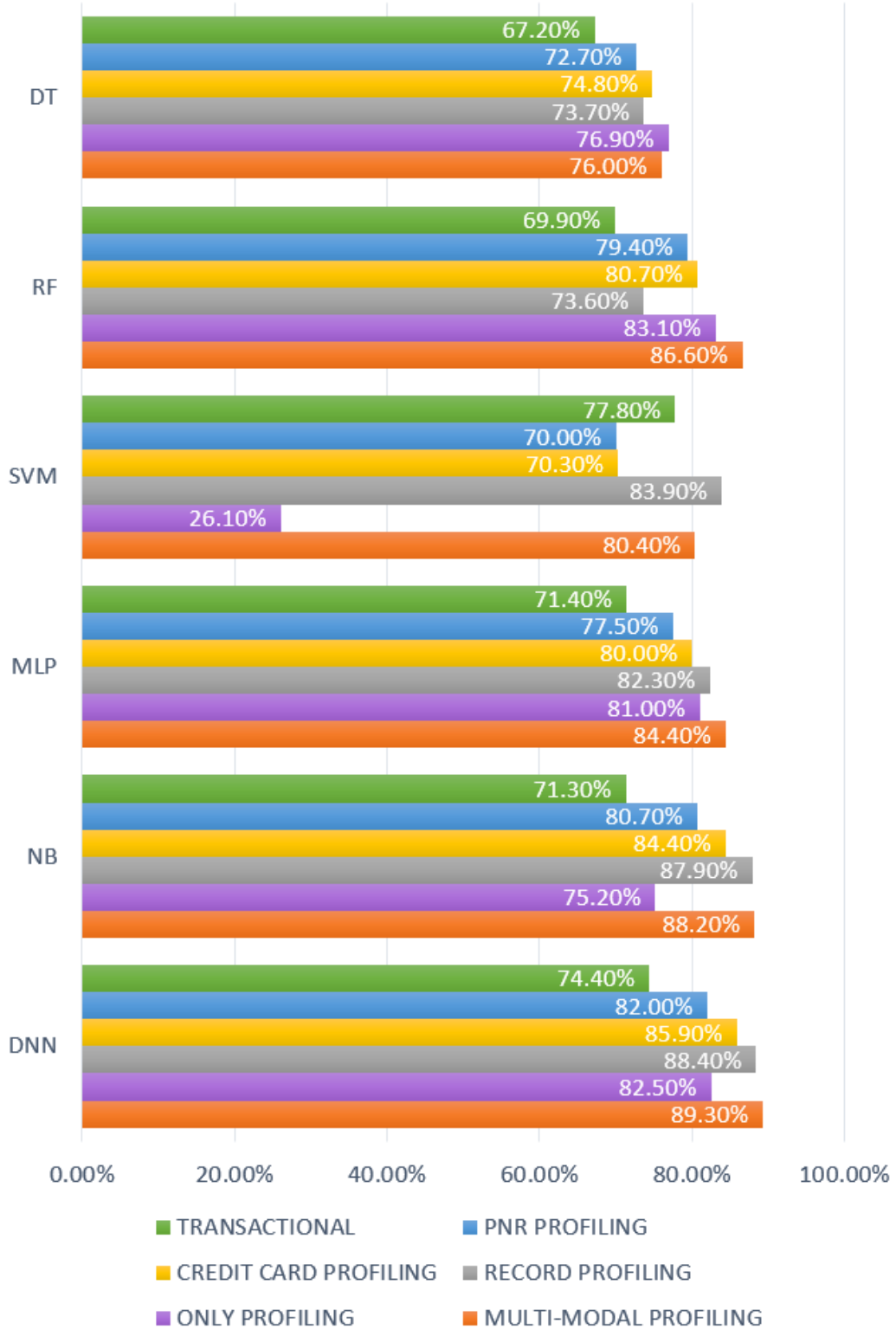


Figure 7.5 The contribution of the proposed profiling modes for each state-of-the-art algorithm on a quantity basis.

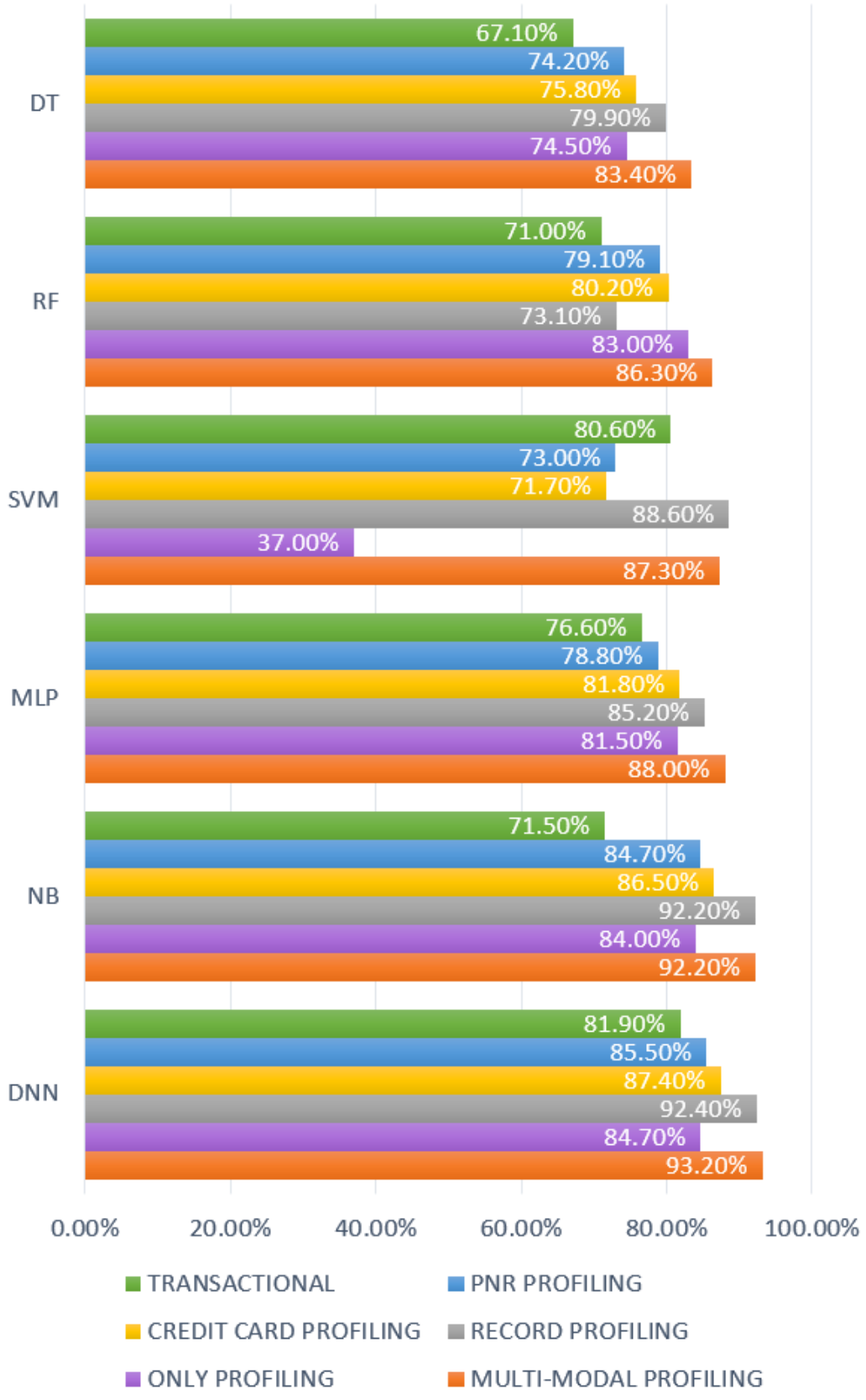


Figure 7.6 The impact of feature-selection algorithms on the performance of initial DNN utilizing multi-modal profiling

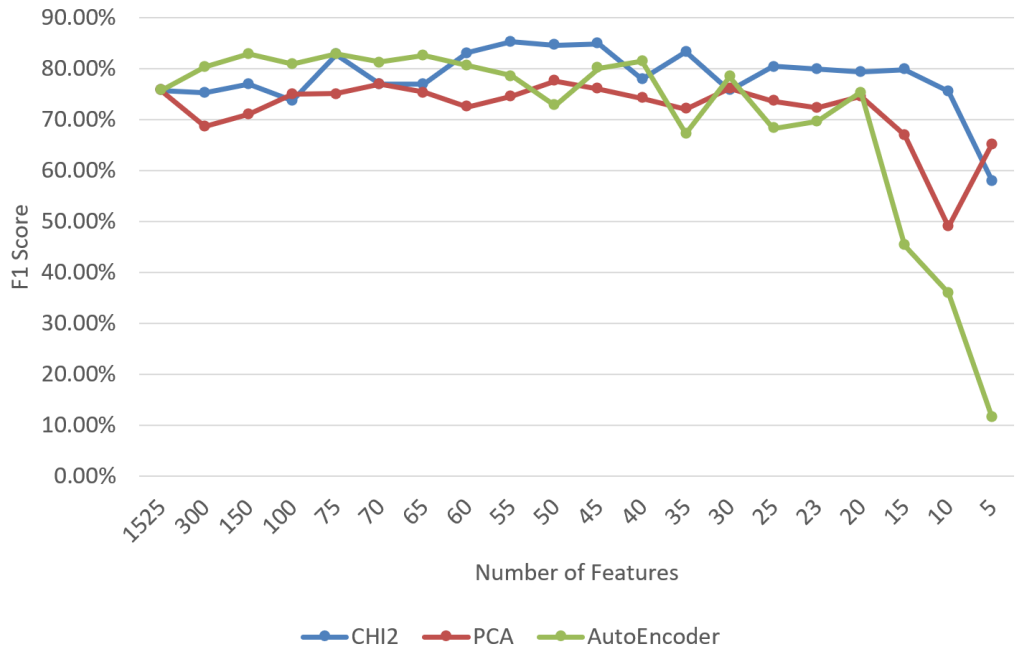


Figure 7.7 The contribution of the proposed profiling modes for each state-of-the-art algorithm on a cost basis.

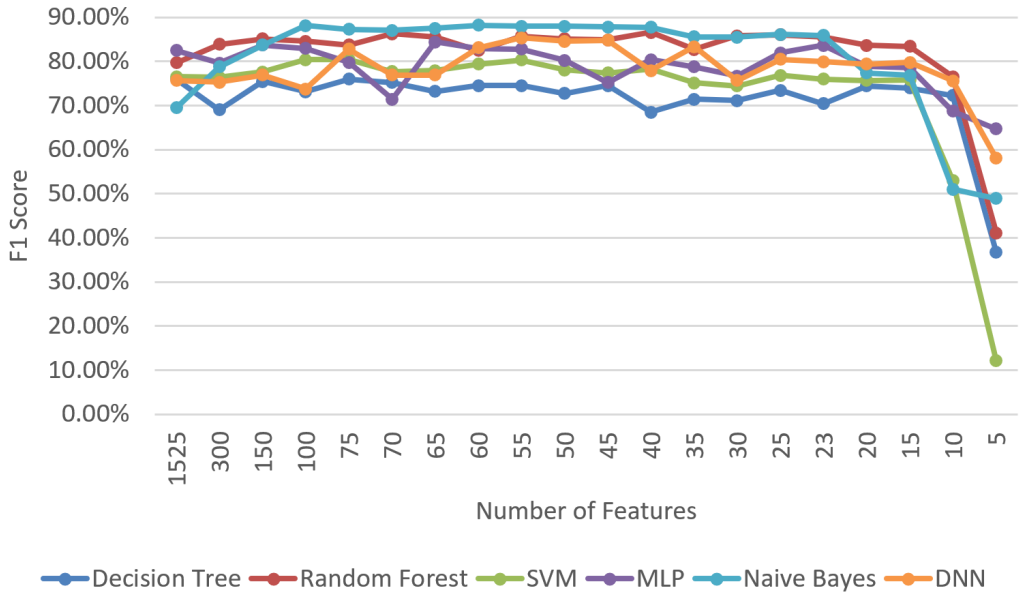


Figure 7.8 The impact of CHI^2 on the state-of-the-art algorithms utilizing multi-modal profiling

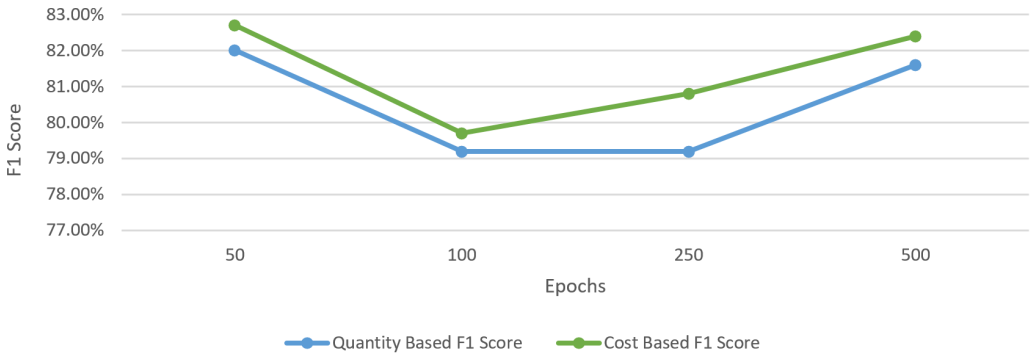


Figure 7.9 The impact of calibration of the number of epochs to the system success

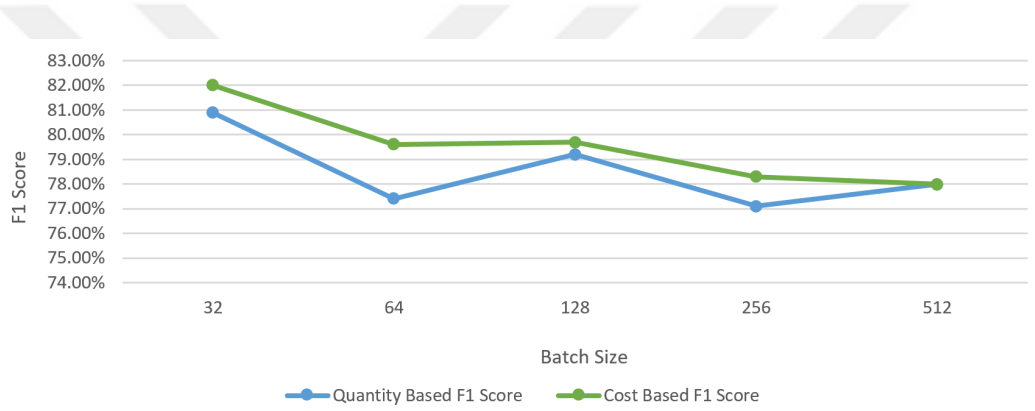


Figure 7.10 The impact of calibration of the batch size to the system success

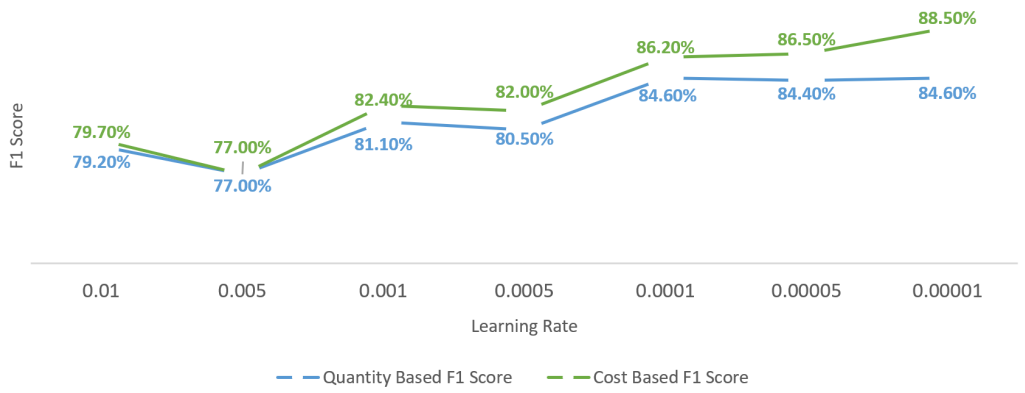


Figure 7.11 The impact of calibration of the learning rate to the system success

F1 Score in terms of quantity and 3.5% in terms of cost. These findings highlight the significance of dropout in enhancing the performance of the DNN for fraud detection purposes.

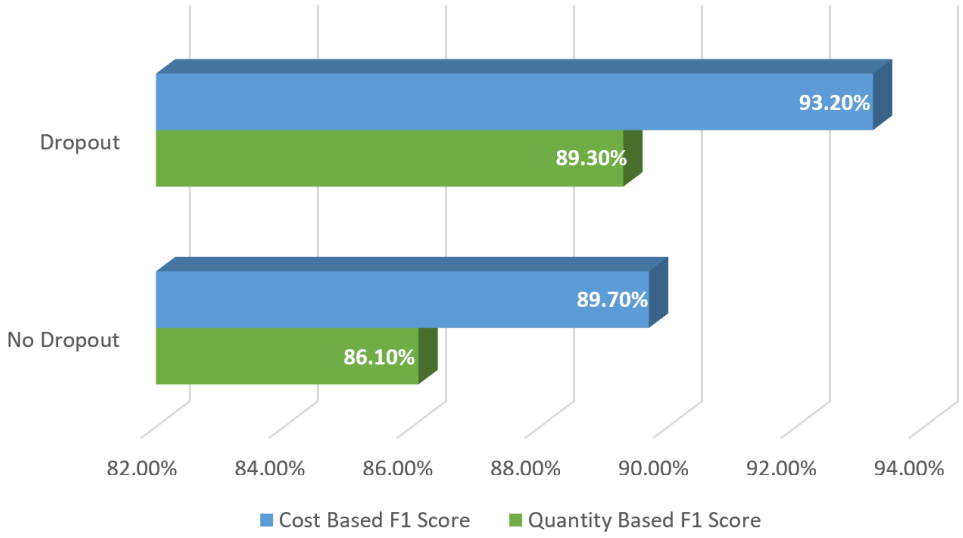


Figure 7.12 The impact of dropout on the DNN hidden layers on the system success

Various configurations of DNN layers are employed in the calibration tests to investigate the influence of DNN formation on the detection performance. The outcomes of these tests are presented in Figure 7.13, where they are arranged in ascending order based on the F1 Scores determined by quantity. The findings unequivocally demonstrate that DNN architectures with a greater number of hidden layers yield higher success rates. Upon scrutinizing the neuron arrangements within these hidden layers, it becomes apparent that success rates are elevated when there is an augmentation or a consistent number of neurons during layer transitions, as opposed to other neuron formations.

The findings from the hyperparameter tuning experiments revealed a noteworthy enhancement in the success rate, with a significant 14.6% increase based on the F1 Score. By examining Figure 7.14, which presents the test results arranged in ascending order, it becomes evident that the tuning phase conducted on the DNN plays a crucial role in the effectiveness of the developed fraud detection system.

According to the findings presented in Table 7.2, it has been demonstrated that the profile-based fraud detection system outperforms the transaction-based fraud detection system. The test results indicate that the profile-based detection systems achieved F1 Scores of approximately 90% and higher for both quantity-based and cost-based measurements. Notably, the most significant improvements were

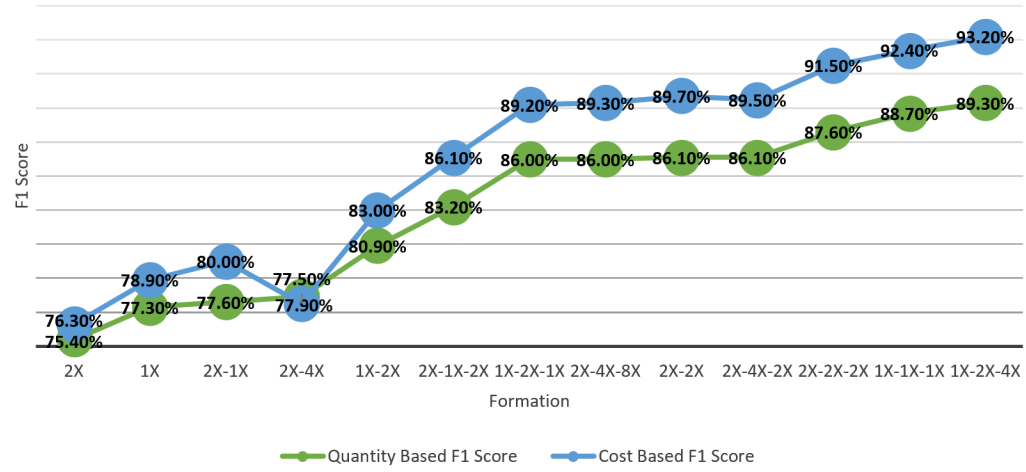


Figure 7.13 The impact of DNN layer formation to the system success

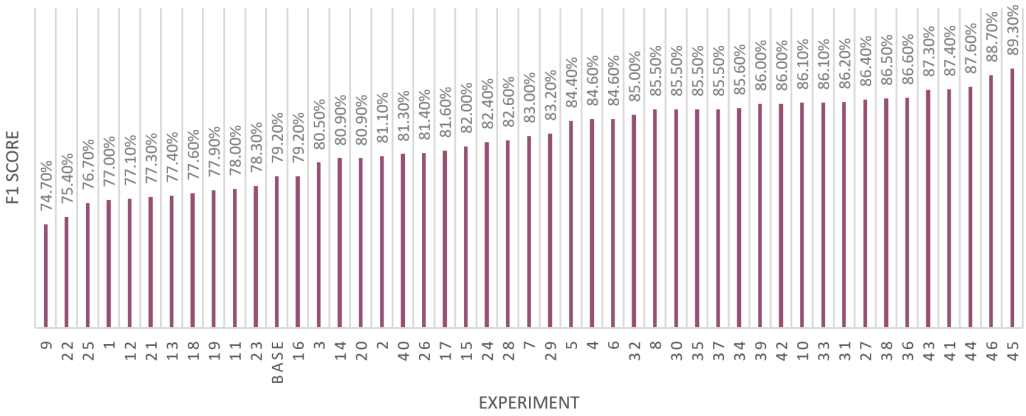


Figure 7.14 Ordered F1 Score results of hyperparameter tuning experiments

observed when utilizing deep neural networks and adjusting the hyperparameters. After deploying the DNN with the new hyperparameters, improvements were observed of 18% and 21.7% on a quantity basis and a cost basis, respectively.

Table 7.2 Transactional approach vs. profiled approach

Quantity-Based					
Approach	**Best Algorithm**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Transactional Approach—Initial	DNN	91.00%	86.90%	54.60%	67.10%
Transactional Approach—Improved	SVM	93.60%	79.00%	76.20%	77.60%
Profiled Approach—Initial	NB	93.30%	90.60%	85.90%	88.20%
Profiled Approach—Improved	DNN	94.20%	95.70%	83.70%	89.30%
Cost-Based					
Approach	**Best Algorithm**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Transactional Approach—Initial	DNN	83.40%	90.30%	58.20%	70.80%
Transactional Approach—Improved	DNN	88.10%	86%	78.10%	81.90%
Profiled Approach—Initial	NB	91.90%	90.70%	93.80%	92.20%
Profiled Approach—Improved	DNN	92.20%	96.20%	90.40%	93.20%

7.3 Ensemble Learning Results on Profiling Approach

At this stage, a new decision mechanism has been created using machine learning algorithms, which are applied as standard in our thesis study and show successful results. This mechanism includes the three most successful algorithms in different combinations. After all the algorithms have decided on the transaction, the final decision is determined by an equally weighted voting method.

While creating the ensemble learning combinations, the most successful single algorithms were taken as the basis. The 5 different ensemble learning combinations

applied in our study are as follows:

- Random Forest, SVM, Naive Bayes
- Random Forest, MLP, Naive Bayes
- Decision Tree, SVM, Naive Bayes
- Decision Tree, MLP, Naive Bayes
- SVM, MLP, Naive Bayes

These ensemble learning combinations we created in our study were tested using multi-modal profiling, the most successful profiling mode. The results of 5 different ensemble learning combinations can be seen in Table G.1, Table G.2, Table G.3, Table G.4 and Table G.5. Comparison of the F1 Score results of the performed ensemble learning tests and existing machine learning algorithms based on quantity is given in Table G.6, and the comparison based on cost is given in Table G.7. According to this comparison, ensemble learning showed almost the same success as the Naive Bayes algorithm, which is the machine learning algorithm with the best results based on the F1 Score. Ensemble learning slightly improved according to machine learning algorithms when examined on a cost basis.

As can be seen in Figure 7.15, although the ensemble learning approaches applied to the profiled dataset showed high success, they could not surpass the results of the most successful algorithm in it. The most successful ensemble learning combination emerged by equally weighting the decisions made by Random Forest, SVM, and Naive Bayes algorithms. F1 Score success of 88.2% on a quantity basis and 92.2% on a cost basis was achieved. While similar results were observed on a quantity basis, there was a slight improvement on a cost basis.

7.4 Fraud Detection Characteristics of the Developed System

The most successful test results performed with the fraud detection system developed in our thesis were examined based on all success metrics, and the system characteristics and some important points were revealed as follows.

- When the accuracy metrics are examined, the accuracy value is 94.2% on a quantity basis and 92.2% on a cost basis. Obtaining very different values based on quantity and cost in our study's previous stages revealed an



Figure 7.15 Comparison of Ensemble Learning and Machine Learning Techniques

imbalance between customer satisfaction and financial success. The fact that the difference between these values was observed to be very low at the last point shows that the developed system makes balanced decisions based on quantity and cost.

- When the results of the precision metrics are examined, it is seen that success was achieved with 92.2% on a quantity basis and 95.7% on a cost basis. This shows that the system both victimizes fewer passengers and prevents the company's potential loss of profit to a large extent.

- When the values of the recall metrics are examined, it is seen that the success rate increases to 83.7% on a quantity basis and 90.4% on a cost basis. This shows that the developed fraud detection system can detect a significant part of fraud transactions, so it has a high fraud detection rate. In addition, considering the higher cost-based success in terms of recall, it can be inferred that high-cost fraud transactions were successfully caught, and the company was saved from further financial losses.

- Considering the F1 Score, which is a balanced average of precision and recall metrics, it was observed that the success rate increased to 89.3% on a quantity basis and 93.2% on a cost basis. This shows that the system's precision, which means potential profit and customer satisfaction, and recall success, which means fraud detection rate, have reached balanced high rates. The relatively higher success rate in terms of cost means that when fraud and legal transactions are evaluated within themselves, transactions with high amounts are detected more accurately.

- **Fraud Detection Rate:** Since fraud is primarily a financial problem, this rate

is generally the primary targeted success type in fraud detection studies. It shows how much of the fraud attempts are caught and is measured by the recall metric. In our study, the algorithm with the highest fraud detection rate was observed to be Naive Bayes, with a success rate of 85.9% on a quantity basis and 93.8% on a cost basis. The lowest success in this regard belongs to the Decision Tree algorithm, with 68.2% on a quantity basis and 63.8% on a cost basis.

- **Preventing False Alarm Cases:** Another important success criterion of fraud detection systems is to prevent customer dissatisfaction by reducing false alarm cases. The precision metric expresses this success. The higher the success of precision, the lower the number of legal passengers who are mistakenly prevented from flying. In our experiments, the algorithm with the highest quantity-based precision success was Random Forest with 95.9%. In this context, the Decision Tree algorithm had the lowest success with 88.1%. Moreover, regarding cost, DNN has the highest precision with 96.2% success, and Decision Tree has the lowest success with 89.4%.
- **Harmonic Fraud Detection Success:** The success of a fraud detection system cannot be measured by precision and recall alone. Due to the imbalanced dataset problem, the accuracy metric can also be misleading at this point. Therefore, our study used the F1 Score metric to measure harmonic fraud detection success. Overall, the most successful method was DNN, which reached F1 Score values of 89.3% on a quantity basis and 93.2% on a cost basis. The least successful algorithm in a holistic sense was the Decision Tree, with 76.9% on a quantity basis and 74.5% on a cost basis. In addition, Decision Tree and Random Forest algorithms showed lower cost-based performance and appeared financially unsuccessful.

7.5 Time Performances of the Developed System

This section includes the time performances of the developed solutions and components. The minimum, maximum, and average time spent per transaction in the cleaning, profiling, embedding layer, one-hot-encoding, and feature selection stages are given in Table 7.3. Training and prediction times spent per transaction in the classification stage are given in Table 7.4

Although the profiling step seems to take the most time overall, the profiling time per transaction is very short since the entire dataset is profiled. Data preparation stages such as cleaning, one-hot-encoding, embedding layer, and feature selection

work on the sampled dataset. These stages similarly have short run times per transaction.

Due to their structure, machine learning algorithms have much lower training and prediction times than deep learning algorithms. The MLP algorithm exhibited much higher training and prediction times than other machine learning algorithms. The SVM algorithm typically exhibits a slower training time, but in our study, the scikit-learn-intelex library developed for Intel processors was used, and the running times of the SVM algorithm were accelerated up to 600 times.

The decision time of the developed deep neural network model for a transaction was measured as 0.049 milliseconds. This period was quite successful for the development of a deep neural network that can be used in a real-time fraud detection mechanism.

Table 7.3 Time performances per transaction of the preprocessing stages of the developed fraud detection system in milliseconds

System Phase	Min	Max	Average
Cleaning	0.013	0.016	0.014
Profiling	0.041	0.052	0.047
One-Hot-Encoding	0.070	0.088	0.079
Single-Layer Embedding	0.175	0.216	0.192
Multi-Layer Embedding	3.121	3.160	3.140
Feature Selection	0.005	0.047	0.025

Table 7.4 Time performances per transaction of the classification stages of the developed fraud detection system in milliseconds

System Phase	Min	Max	Average
DT Training	0.003	0.013	0.008
DT Prediction	0.006	0.03	0.019
RF Training	0.001	0.003	0.002
RF Prediction	0.001	0.008	0.005
SVM Training	0.019	0.024	0.021
SVM Prediction	0.047	0.058	0.052
MLP Training	1.218	2.182	1.594
MLP Prediction	2.819	5.065	3.699
NB Training	0.001	0.001	0.001
NB Prediction	0.002	0.005	0.003
DL Training	1.967	21.400	5.311
DL Prediction	0.040	0.061	0.049

7.6 High Scale Fraud/Legal Ratio Experiments

Using the profile-based deep learning configuration that gave the most successful results in our study, we performed tests that approached the real fraud/legal ratio. Based on the 1:5 fraud/legal ratio, which we determined as default in the initial stages, we utilized this ratio as a basis with tests performed at 1:10, 1:25, 1:50, 1:100, 1:250, and 1:500 fraud/legal ratios. As can be seen in Figure 7.16 and Figure 7.17, when we tested the model trained with a 1:5 fraud/legal ratio with higher negative sample rates, a decreasing trend in precision success was observed. Precision, or false alarm prevention success, decreased by up to 25.5% on a quantity basis and up to 25.3% on a cost basis. Recall success showed slight fluctuations, decreasing by 1.3% on a quantity basis and 2.2% on a cost basis. A decrease of up to 13.5% in quantity and up to 14.6% in cost was observed in the harmonic success of the fraud detection system, that is, the F1 Score values. These tests have shown that the fraud/legal ratio to be chosen is of great importance in balancing the success and reality of the developed system.

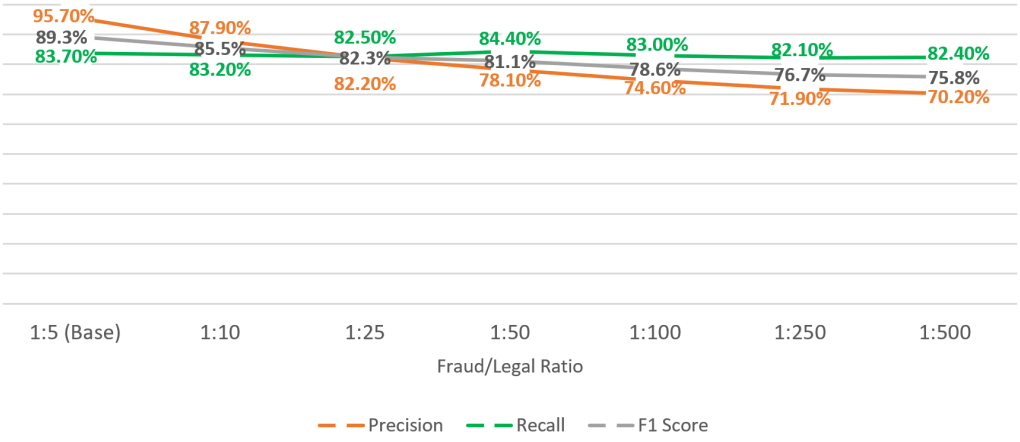


Figure 7.16 Higher scale fraud/legal ratio tests (quantity based)

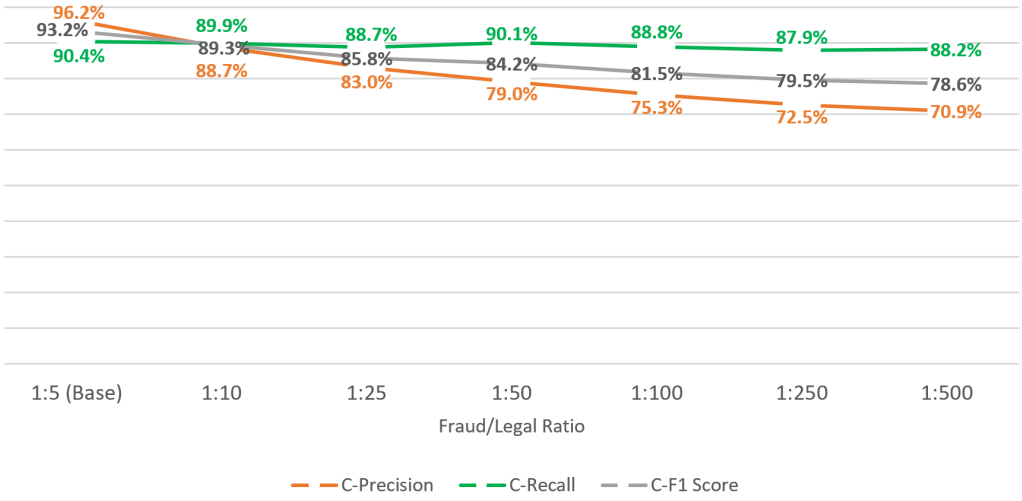


Figure 7.17 Higher scale fraud/legal ratio tests (cost based)

8
CONCLUSION

Airline companies have faced significant challenges in the past ten years due to fraudulent activities. While previous studies have focused on detecting fraud in the banking sector, particularly credit card fraud, this study stands out as the first comprehensive investigation into fraudulent activities within the aviation industry. Our research began by establishing a baseline for the success rates of existing state-of-the-art learning techniques. Subsequently, we implemented various preprocessing techniques on transactional airline sales data to enhance the effectiveness of these techniques. Through this process, we identified the limitations of transactional information in detecting suspicious airline ticket activities. To overcome these limitations, we recognized the importance of utilizing a passenger's historical background and consequently developed profiles based on credit card, PNR, and record activities. By combining this profiling mechanism with a deep learning approach, our proposed fraud system achieved a remarkable 33.1% improvement in the F1 Score. Moreover, this system also demonstrated the ability to reduce financial losses by up to 31.6%. The amount field is an essential component in our study's cost-sensitive measurement and multi-modal profiling mechanisms. It plays a crucial role in calculating cost-based success and generating various statistical profile fields. Our analysis revealed that certain engineered features, including the number of chargebacks, e-mail addresses, phone numbers, and domestic and international flights, are particularly significant in credit card and PNR profiles. Furthermore, we found that information about minutes to flight time and the match status of passenger surname and cardholder surname, obtained from record profiles, are key features for detecting fraudulent transactions in the aviation industry. We also observed that fine-tuning the architecture of the DNN specifically for aviation data can lead to a substantial improvement of 14.6% in terms of the F1 Score, compared to randomly chosen hyperparameters or those derived from existing studies in the banking sector. Finally, when the DNN model created in our study was compared with machine learning algorithms regarding F1 Score success, it showed higher success between 1.3% and 17.5% on a quantity basis and

between 1.1% and 10.5% on a cost basis. Moving forward, we plan to explore loss functions that are compatible with imbalanced data and potentially design new ones that consider cost-based gain. Additionally, our proposed multi-modal profiling mechanism can be applied to other sectors with high data diversity, such as e-commerce and insurance. However, in industries like banking, where the data structure is not suitable for multiple profiles, a single-modal profiling mechanism based on credit card activities will need to be established instead of a multi-modal approach.

REFERENCES

[1] Y. Kou, C.-T. Lu, S. Sirwongwattana, Y.-P. Huang, "Survey of fraud detection techniques," in *IEEE International Conference on Networking, Sensing and Control, 2004*, IEEE, vol. 2, 2004, pp. 749–754.

[2] K. Supraja, S. J. Saritha, "Robust fuzzy rule based technique to detect frauds in vehicle insurance," in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, IEEE, 2017, pp. 3734–3739.

[3] J. O. Awoyemi, A. O. Adetunmbi, S. A. Oluwadare, "Credit card fraud detection using machine learning techniques: A comparative analysis," in *2017 international conference on computing networking and informatics (ICCNI)*, IEEE, 2017, pp. 1–9.

[4] X. Kewei, B. Peng, Y. Jiang, T. Lu, "A hybrid deep learning model for online fraud detection," in *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, IEEE, 2021, pp. 431–434.

[5] M. T. Aras, M. A. Güvensan, "Challenges and key points for fraud detection in aviation," in *2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, IEEE, 2021, pp. 1–6.

[6] P. Tomar, S. Shrivastava, U. Thakar, "Ensemble learning based credit card fraud detection system," in *2021 5th Conference on Information and Communication Technology (CICT)*, IEEE, 2021, pp. 1–5.

[7] V. Jain, M. Agrawal, A. Kumar, "Performance analysis of machine learning algorithms in credit cards fraud detection," in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*, IEEE, 2020, pp. 86–88.

[8] A. El Bouchti, A. Chakroun, H. Abbar, C. Okar, "Fraud detection in banking using deep reinforcement learning," in *2017 Seventh International Conference on Innovative Computing Technology (INTECH)*, IEEE, 2017, pp. 58–63.

[9] F. K. Alarfaj, I. Malik, H. U. Khan, N. Almusallam, M. Ramzan, M. Ahmed, "Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms," *IEEE Access*, vol. 10, pp. 39 700–39 715, 2022.

[10] D. Sarma, W. Alam, I. Saha, M. N. Alam, M. J. Alam, S. Hossain, "Bank fraud detection using community detection algorithm," in *2020 second international conference on inventive research in computing applications (ICIRCA)*, IEEE, 2020, pp. 642–646.

[11] Y. Singh, K. Singh, V. S. Chauhan, "Fraud detection techniques for credit card transactions," in *2022 3rd International Conference on Intelligent Engineering and Management (ICIEM)*, IEEE, 2022, pp. 821–824.

[12] N. Boutaher, A. Elomri, N. Abghour, K. Moussaid, M. Rida, "A review of credit card fraud detection using machine learning techniques," in *2020 5th International Conference on cloud computing and artificial intelligence: technologies and applications (CloudTech)*, IEEE, 2020, pp. 1–5.

[13] G. Zhang *et al.*, "Efraudcom: An e-commerce fraud detection system via competitive graph neural networks," *ACM Transactions on Information Systems (TOIS)*, vol. 40, no. 3, pp. 1–29, 2022.

[14] S. Carta, G. Fenu, D. R. Recupero, R. Saia, "Fraud detection for e-commerce transactions by employing a prudential multiple consensus model," *Journal of Information Security and Applications*, vol. 46, pp. 13–22, 2019.

[15] J. Wang, J. Liu, Z. Xia, P. Chen, X. Li, X. Chen, "Semi-supervised labeling model based on gaussian mixture in the context of e-commerce price fraud," in *2022 4th International Conference on Robotics and Computer Vision (ICRCV)*, IEEE, 2022, pp. 300–304.

[16] B. K. Sethi, D. Singh, P. K. Sarangi, "Medical insurance fraud detection based on block chain and deep learning approach," in *2022 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON)*, IEEE, vol. 2, 2022, pp. 103–106.

[17] N. S. Patil, S. Kamanavalli, S. Hiregoudar, S. Jadhav, S. Kanakraddi, N. D. Hiremath, "Vehicle insurance fraud detection system using robotic process automation and machine learning," in *2021 International Conference on Intelligent Technologies (CONIT)*, IEEE, 2021, pp. 1–5.

[18] S. Vyas, S. Serasiya, "Fraud detection in insurance claim system: A review," in *2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, IEEE, 2022, pp. 922–927.

[19] *Cybersource corp. fraud and risk management*. [Online]. Available: `https://www.cybersource.com/en/solutions/fraudand-risk-management.html` (visited on 06/12/2023).

[20] T. C. Tran, T. K. Dang, "Machine learning for prediction of imbalanced data: Credit fraud detection," in *2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, IEEE, 2021, pp. 1–7.

[21] R. Qaddoura, M. M. Biltawi, "Improving fraud detection in an imbalanced class distribution using different oversampling techniques," in *2022 International Engineering Conference on Electrical, Energy, and Artificial Intelligence (EICEEAI)*, IEEE, 2022, pp. 1–5.

[22] A. Hanskunatai, "A new hybrid sampling approach for classification of imbalanced datasets," in *2018 3rd International Conference on Computer and Communication Systems (ICCCS)*, IEEE, 2018, pp. 67–71.

[23] M. del Mar Roldán-García, J. García-Nieto, J. F. Aldana-Montes, "Enhancing semantic consistency in anti-fraud rule-based expert systems," *Expert Systems with Applications*, vol. 90, pp. 332–343, 2017.

[24] K. D. Febriyanti, R. Sarno, Y. A. Effendi, "Fraud detection on event logs using fuzzy association rule learning," in *2017 11th International Conference on Information & Communication Technology and System (ICTS)*, IEEE, 2017, pp. 149–154.

[25] X. Yu, X. Li, Y. Dong, R. Zheng, "A deep neural network algorithm for detecting credit card fraud," in *2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, IEEE, 2020, pp. 181–183.

[26] S. Chang, C. Wang, C. Wang, "Automated feature engineering for fraud prediction in online credit loan services," in *2022 13th Asian Control Conference (ASCC)*, IEEE, 2022, pp. 738–743.

[27] D. Olszewski, "Fraud detection using self-organizing map visualizing the user profiles," *Knowledge-Based Systems*, vol. 70, pp. 324–334, 2014.

[28] D. Malekian, M. R. Hashemi, "An adaptive profile based fraud detection framework for handling concept drift," in *2013 10th International ISC Conference on Information Security and Cryptology (ISCISC)*, IEEE, 2013, pp. 1–6.

[29] L. Seyedhossein, M. R. Hashemi, "Mining information from credit card time series for timelier fraud detection," in *2010 5th International Symposium on Telecommunications*, IEEE, 2010, pp. 619–624.

[30] J. Forough, S. Momtazi, "Ensemble of deep sequential models for credit card fraud detection," *Applied Soft Computing*, vol. 99, p. 106 883, 2021.

[31] H. Najadat, O. Altiti, A. A. Aqouleh, M. Younes, "Credit card fraud detection based on machine and deep learning," in *2020 11th International Conference on Information and Communication Systems (ICICS)*, IEEE, 2020, pp. 204–208.

[32] T. T. Nguyen, H. Tahir, M. Abdelrazek, A. Babar, "Deep learning methods for credit card fraud detection," *arXiv preprint arXiv:2012.03754*, 2020.

[33] I. Mekterović, M. Karan, D. Pintar, L. Brkić, "Credit card fraud detection in card-not-present transactions: Where to invest?" *Applied Sciences*, vol. 11, no. 15, p. 6766, 2021.

[34] S. N. John, C. Anele, O. O. Kennedy, F. Olajide, C. G. Kennedy, "Realtime fraud detection in the banking sector using data mining techniques/algorithm," in *2016 international conference on computational science and computational intelligence (CSCI)*, IEEE, 2016, pp. 1186–1191.

[35] H. Weng *et al.*, "Online e-commerce fraud: A large-scale detection and analysis," in *2018 ieee 34th international conference on data engineering (icde)*, IEEE, 2018, pp. 1435–1440.

[36] R. Roy, K. T. George, "Detecting insurance claims fraud using machine learning techniques," in *2017 international conference on circuit, power and computing technologies (ICCPCT)*, IEEE, 2017, pp. 1–6.

[37] Y. Wang, W. Xu, "Leveraging deep learning with lda-based text analytics to detect automobile insurance fraud," *Decision Support Systems*, vol. 105, pp. 87–95, 2018.

[38] Z. Zhang, S. Huang, "Credit card fraud detection via deep learning method using data balance tools," in *2020 international conference on computer science and management technology (ICCSMT)*, IEEE, 2020, pp. 133–137.

[39] T. Baabdullah, A. Alzahrani, D. B. Rawat, "On the comparative study of prediction accuracy for credit card fraud detection wwith imbalanced classifications," in *2020 Spring Simulation Conference (SpringSim)*, IEEE, 2020, pp. 1–12.

[40] M. Zareapoor, P. Shamsolmoali, *et al.*, "Application of credit card fraud detection: Based on bagging ensemble classifier," *Procedia computer science*, vol. 48, no. 2015, pp. 679–685, 2015.

[41] M. S. Mahmud, P. Meesad, S. Sodsee, "An evaluation of computational intelligence in credit card fraud detection," in *2016 International Computer Science and Engineering Conference (ICSEC)*, IEEE, 2016, pp. 1–6.

[42] B. Krawczyk, C. Bellinger, R. Corizzo, N. Japkowicz, "Undersampling with support vectors for multi-class imbalanced data classification," in *2021 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2021, pp. 1–7.

[43] M. Bach, A. Werner, M. Palt, "The proposal of undersampling method for learning from imbalanced datasets," *Procedia Computer Science*, vol. 159, pp. 125–134, 2019.

[44] T. M. Alam *et al.*, "An investigation of credit card default prediction in the imbalanced datasets," *IEEE Access*, vol. 8, pp. 201 173–201 198, 2020.

[45] C. Liu *et al.*, "Constrained oversampling: An oversampling approach to reduce noise generation in imbalanced datasets with class overlapping," *IEEE Access*, vol. 10, pp. 91 452–91 465, 2020.

[46] H. Shi, Q. Gao, S. Ji, Y. Liu, "A hybrid sampling method based on safe screening for imbalanced datasets with sparse structure," in *2018 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2018, pp. 1–8.

[47] M. A. Alamri, M. A. Ykhlef, "A machine learning-based framework for detecting credit card anomalies and fraud," in *2023 27th International Conference on Information Technology (IT)*, IEEE, 2023, pp. 1–7.

[48] F. Ghobadi, M. Rohani, "Cost sensitive modeling of credit card fraud using neural network strategy," in *2016 2nd international conference of signal processing and intelligent systems (ICSPIS)*, IEEE, 2016, pp. 1–5.

[49] P. Pant, P. Srivastava, "Cost-sensitive model evaluation approach for financial fraud detection system," in *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, IEEE, 2021, pp. 1606–1611.

[50] A. C. Bahnsen, A. Stojanovic, D. Aouada, B. Ottersten, "Cost sensitive credit card fraud detection using bayes minimum risk," in *2013 12th international conference on machine learning and applications*, IEEE, vol. 1, 2013, pp. 333–338.

[51] V. Jain, H. Kavitha, S. M. Kumar, "Credit card fraud detection web application using streamlit and machine learning," in *2022 IEEE International Conference on Data Science and Information System (ICDSIS)*, IEEE, 2022, pp. 1–5.

[52] A. Singh, A. Singh, A. Aggarwal, A. Chauhan, "Design and implementation of different machine learning algorithms for credit card fraud detection," in *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, IEEE, 2022, pp. 1–6.

[53] M. V. Krishna, J. Praveenchandar, "Comparative analysis of credit card fraud detection using logistic regression with random forest towards an increase in accuracy of prediction," in *2022 International Conference on Edge Computing and Applications (ICECAA)*, IEEE, 2022, pp. 1097–1101.

[54] Y. Xie, G. Liu, C. Yan, C. Jiang, M. Zhou, "Time-aware attention-based gated network for credit card fraud detection by extracting transactional behaviors," *IEEE Transactions on Computational Social Systems*, 2022.

[55] C. Lawrencia, W. Ce, "Fraud detection decision support system for indonesian financial institution," in *2019 International Conference on Information Management and Technology (ICIMTech)*, IEEE, vol. 1, 2019, pp. 389–394.

[56] D. S. Nijwala, S. Maurya, M. Thapliyal, R. Verma, "Extreme gradient boost classifier based credit card fraud detection model," in *2023 International Conference on Device Intelligence, Computing and Communication Technologies,(DICCT)*, IEEE, 2023, pp. 500–504.

[57] B. Can, A. G. Yavuz, E. M. Karsligil, M. A. Guvensan, "A closer look into the characteristics of fraudulent card transactions," *IEEE Access*, vol. 8, pp. 166 095–166 109, 2020.

[58] G. E. Melo-Acosta, F. Duitama-Munoz, J. D. Arias-Londoño, "Fraud detection in big data using supervised and semi-supervised learning techniques," in *2017 IEEE Colombian conference on communications and computing (COLCOM)*, IEEE, 2017, pp. 1–6.

[59] J. A. Gómez, J. Arévalo, R. Paredes, J. Nin, "End-to-end neural network architecture for fraud scoring in card payments," *Pattern Recognition Letters*, vol. 105, pp. 175–181, 2018.

[60] Y. Lu, *Deep neural networks and fraud detection*, 2017.

[61] M. T. Aras, M. A. Guvensan, "A multi-modal profiling fraud-detection system for capturing suspicious airline ticket activities," *Applied Sciences*, vol. 13, no. 24, p. 13 121, 2023.

[62] J. Riffi, M. A. Mahraz, A. El Yahyaouy, H. Tairi, *et al.*, "Credit card fraud detection based on multilayer perceptron and extreme learning machine architectures," in *2020 International Conference on Intelligent Systems and Computer Vision (ISCV)*, IEEE, 2020, pp. 1–5.

[63] D. Varmedja, M. Karanovic, S. Sladojevic, M. Arsenovic, A. Anderla, "Credit card fraud detection-machine learning methods," in *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*, IEEE, 2019, pp. 1–5.

[64] Y. Su, X. Zhu, B. Dong, Y. Zhang, X. W. Wu, "Medfrodetect: Medicare fraud detection with extremely imbalanced class distributions," in *The thirty-third international FLAIRS conference (FLAIRS-32)*, 2020.

[65] A. Langevin, T. Cody, S. Adams, P. Beling, "Synthetic data augmentation of imbalanced datasets with generative adversarial networks under varying distributional assumptions: A case study in credit card fraud detection," *J. Oper. Res. Soc*, pp. 1–28, 2021.

[66] R. Reshma, R. Santhosh, N. Mekala, *et al.*, "An analytical approach to fraudulent credit card transaction detection using various machine learning algorithms," in *2023 Second International Conference on Electronics and Renewable Systems (ICEARS)*, IEEE, 2023, pp. 1400–1404.

[67] M. L. Gambo, A. Zainal, M. N. Kassim, "A convolutional neural network model for credit card fraud detection," in *2022 International Conference on Data Science and Its Applications (ICoDSA)*, IEEE, 2022, pp. 198–202.

[68] S. Bonkoungou, N. R. Roy, N. H. A.-E. Ako, U. Batra, "Credit card fraud detection using ml: A survey," in *2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*, IEEE, 2023, pp. 732–738.

[69] S. Mittal, S. Tyagi, "Performance evaluation of machine learning algorithms for credit card fraud detection," in *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, IEEE, 2019, pp. 320–324.

[70] A. H. Nadim, I. M. Sayem, A. Mutsuddy, M. S. Chowdhury, "Analysis of machine learning techniques for credit card fraud detection," in *2019 International Conference on Machine Learning and Data Engineering (iCMLDE)*, IEEE, 2019, pp. 42–47.

[71] S. Shirgave, C. Awati, R. More, S. Patil, "A review on credit card fraud detection using machine learning," *International Journal of Scientific & technology research*, vol. 8, no. 10, pp. 1217–1220, 2019.

[72] J. Singla *et al.*, "A survey of deep learning based online transactions fraud detection systems," in *2020 International Conference on Intelligent Engineering and Management (ICIEM)*, IEEE, 2020, pp. 130–136.

A EMBEDDING LAYER EXPERIMENTS

Table A.1 Embedding layer test results without feature selection (ML = MultiLayer , SL = SingleLayer, X = Embedding size per categorical field)

Embedding Method	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
ML	0.9	0.854	0.93	0.935	0.711	0.768	0.806	0.843
SL(10X)	0.912	0.882	0.927	0.943	0.757	0.819	0.834	0.877
SL(20X)	0.907	0.865	0.912	0.916	0.753	0.811	0.825	0.86
SL(30X)	0.936	0.917	0.937	0.936	0.838	0.916	0.884	0.928
SL(40X)	0.915	0.881	0.908	0.926	0.786	0.834	0.842	0.878
SL(50X)	0.903	0.869	0.91	0.932	0.738	0.802	0.815	0.862

Table A.2 Embedding layer test results without feature selection (ML = MultiLayer , SL = SingleLayer, X = Embedding size per categorical field)

Embedding Method	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
ML	0.868	0.777	0.946	0.957	0.58	0.59	0.719	0.73
SL(10X)	0.938	0.912	0.931	0.929	0.85	0.924	0.888	0.926
SL(20X)	0.901	0.839	0.961	0.954	0.687	0.721	0.801	0.821
SL(30X)	0.919	0.874	0.953	0.962	0.76	0.784	0.846	0.864
SL(40X)	0.916	0.85	0.938	0.945	0.761	0.751	0.84	0.837
SL(50X)	0.919	0.859	0.953	0.96	0.758	0.757	0.844	0.846

B
TRAIN TEST DISTRIBUTION EXPERIMENTS

Table B.1 Random Distribution Results According to Fraud/Legitimate Balances

Algorithm	Fraud/Legitimate Balance	Accuracy	Precision	Recall	F1 Score
Decision Tree	1:1	0.932	0.911	0.959	0.934
	1:2	0.931	0.876	0.929	0.902
	1:5	0.958	0.846	0.913	0.878
	1:10	0.973	0.826	0.897	0.860
Random Forest	1:1	0.938	0.923	0.957	0.940
	1:2	0,956	0,922	0,952	0,937
	1:5	0,972	0,922	0,906	0,914
	1:10	0,98	0,939	0,841	0,888
SVM	1:1	0,914	0,898	0,936	0,917
	1:2	0,925	0,879	0,905	0,892
	1:5	0,942	0,83	0,814	0,822
	1:10	0,957	0,834	0,661	0,737
MLP	1:1	0,94	0,921	0,962	0,941
	1:2	0,942	0,886	0,954	0,919
	1:5	0,958	0,825	0,942	0,880
	1:10	0,979	0,900	0,866	0,883
Naive Bayes	1:1	0,757	0,682	0,972	0,801
	1:2	0,634	0,483	0,982	0,648
	1:5	0,519	0,253	0,988	0,403
	1:10	0,475	0,148	0,985	0,257

Table B.2 Chronological Distribution Results According to Fraud/Legitimate Balances

Algorithm	Fraud/Legitimate Balance	Accuracy	Precision	Recall	F1 Score
Decision Tree	1:1	0,831	0,856	0,796	0,825
	1:2	0,858	0,837	0,712	0,770
	1:5	0,906	0,812	0,569	0,669
	1:10	0,938	0,771	0,452	0,570
Random Forest	1:1	0,878	0,905	0,846	0,874
	1:2	0,853	0,904	0,627	0,74
	1:5	0,9	0,908	0,45	0,602
	1:10	0,943	0,953	0,393	0,556
SVM	1:1	0,862	0,881	0,837	0,858
	1:2	0,876	0,921	0,689	0,788
	1:5	0,911	0,893	0,534	0,669
	1:10	0,941	0,877	0,41	0,558
MLP	1:1	0,844	0,906	0,769	0,832
	1:2	0,843	0,874	0,62	0,725
	1:5	0,909	0,869	0,534	0,662
	1:10	0,938	0,789	0,43	0,557
Naive Bayes	1:1	0,844	0,906	0,769	0,832
	1:2	0,554	0,337	0,347	0,342
	1:5	0,544	0,135	0,321	0,19
	1:10	0,572	0,071	0,305	0,115

Table B.3 Monthly Sequential Distribution Results

Algorithm	Train-Test Months	Accuracy	Precision	Recall	F1 Score
Decision Tree	2018.08 - 2018.09	0,906	0,808	0,574	0,671
	2018.09 - 2018.10	0,898	0,844	0,476	0,609
	2018.10 - 2018.11	0,887	0,848	0,394	0,538
	2018.11 - 2018.12	0,89	0,774	0,475	0,589
Random Forest	2018.08 - 2018.09	0,893	0,872	0,422	0,569
	2018.09 - 2018.10	0,876	0,864	0,307	0,453
	2018.10 - 2018.11	0,883	0,924	0,324	0,48
	2018.11 - 2018.12	0,917	0,92	0,55	0,689
SVM	2018.08 - 2018.09	0,905	0,756	0,632	0,688
	2018.09 - 2018.10	0,886	0,756	0,469	0,579
	2018.10 - 2018.11	0,918	0,834	0,637	0,722
	2018.11 - 2018.12	0,906	0,808	0,574	0,671
MLP	2018.08 - 2018.09	0,904	0,812	0,552	0,657
	2018.09 - 2018.10	0,883	0,815	0,388	0,526
	2018.10 - 2018.11	0,908	0,862	0,534	0,659
	2018.11 - 2018.12	0,9	0,857	0,482	0,617
Naive Bayes	2018.08 - 2018.09	0,772	0,362	0,484	0,414
	2018.09 - 2018.10	0,773	0,331	0,354	0,342
	2018.10 - 2018.11	0,778	0,395	0,624	0,484
	2018.11 - 2018.12	0,839	0,516	0,533	0,524

Table B.4 Decision Tree - Cumulative Distribution Results (1 Month Test)

Cumulation Months	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
1	0.912	0.914	0.809	0.792	0.621	0.612	0.702	0.691
2	0.909	0.912	0.831	0.785	0.57	0.582	0.676	0.668
3	0.924	0.923	0.84	0.851	0.674	0.646	0.748	0.735
4	0.926	0.928	0.833	0.776	0.691	0.703	0.755	0.737
5	0.908	0.899	0.821	0.81	0.569	0.566	0.672	0.667
6	0.929	0.928	0.826	0.835	0.73	0.724	0.775	0.775
7	0.906	0.911	0.83	0.861	0.548	0.541	0.661	0.665
8	0.924	0.918	0.828	0.863	0.685	0.665	0.75	0.751
9	0.905	0.9	0.837	0.826	0.534	0.512	0.652	0.633
10	0.926	0.922	0.83	0.811	0.694	0.674	0.756	0.736
11	0.904	0.906	0.802	0.827	0.562	0.601	0.661	0.696
12	0.906	0.891	0.796	0.747	0.581	0.536	0.672	0.624
13	0.909	0.903	0.812	0.817	0.593	0.565	0.686	0.668
14	0.923	0.928	0.823	0.778	0.684	0.66	0.747	0.714

Table B.5 Random Forest - Cumulative Distribution Results (1 Month Test)

Cumulation Months	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
1	0.922	0.919	0.841	0.803	0.656	0.631	0.737	0.707
2	0.917	0.921	0.864	0.823	0.596	0.613	0.706	0.702
3	0.922	0.918	0.831	0.818	0.67	0.644	0.742	0.721
4	0.922	0.924	0.817	0.753	0.687	0.706	0.746	0.729
5	0.912	0.906	0.852	0.849	0.569	0.571	0.682	0.683
6	0.928	0.925	0.836	0.839	0.707	0.693	0.766	0.759
7	0.914	0.917	0.848	0.89	0.586	0.56	0.693	0.688
8	0.927	0.922	0.831	0.867	0.706	0.69	0.764	0.769
9	0.916	0.907	0.858	0.829	0.597	0.563	0.704	0.671
10	0.927	0.927	0.821	0.816	0.719	0.706	0.766	0.757
11	0.918	0.918	0.835	0.852	0.634	0.656	0.721	0.741
12	0.918	0.909	0.822	0.807	0.646	0.61	0.724	0.695
13	0.92	0.914	0.835	0.852	0.65	0.608	0.731	0.709
14	0.926	0.933	0.827	0.805	0.702	0.668	0.76	0.73

Table B.6 SVM - Cumulative Distribution Results (1 Month Test)

Cumulation Months	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
1	0.929	0.926	0.848	0.802	0.7	0.697	0.767	0.746
2	0.928	0.928	0.847	0.795	0.694	0.706	0.763	0.748
3	0.928	0.92	0.844	0.831	0.698	0.648	0.764	0.728
4	0.928	0.927	0.846	0.774	0.695	0.696	0.763	0.733
5	0.922	0.921	0.829	0.836	0.671	0.689	0.741	0.756
6	0.926	0.919	0.833	0.828	0.695	0.664	0.758	0.737
7	0.93	0.93	0.839	0.856	0.714	0.682	0.771	0.759
8	0.934	0.933	0.844	0.869	0.738	0.755	0.787	0.755
9	0.931	0.937	0.844	0.867	0.716	0.737	0.774	0.797
10	0.934	0.929	0.847	0.841	0.738	0.692	0.789	0.759
11	0.931	0.937	0.846	0.867	0.72	0.765	0.778	0.765
12	0.93	0.926	0.846	0.856	0.705	0.675	0.769	0.755
13	0.933	0.942	0.851	0.879	0.724	0.769	0.782	0.82
14	0.934	0.938	0.852	0.815	0.726	0.703	0.784	0.755

Table B.7 MLP - Cumulative Distribution Results (1 Month Test)

Cumulation Months	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
1	0.916	0.912	0.815	0.772	0.639	0.62	0.717	0.688
2	0.917	0.913	0.844	0.755	0.614	0.627	0.711	0.685
3	0.917	0.912	0.817	0.806	0.646	0.616	0.722	0.699
4	0.924	0.924	0.833	0.761	0.679	0.687	0.748	0.722
5	0.919	0.914	0.816	0.805	0.661	0.682	0.73	0.739
6	0.922	0.917	0.819	0.818	0.684	0.661	0.746	0.731
7	0.917	0.915	0.843	0.861	0.618	0.568	0.713	0.685
8	0.928	0.925	0.814	0.859	0.732	0.715	0.771	0.78
9	0.928	0.927	0.853	0.863	0.682	0.669	0.758	0.753
10	0.932	0.92	0.825	0.815	0.684	0.653	0.748	0.725
11	0.927	0.93	0.838	0.856	0.695	0.73	0.76	0.788
12	0.925	0.924	0.82	0.824	0.704	0.687	0.758	0.754
13	0.923	0.921	0.824	0.851	0.683	0.657	0.747	0.742
14	0.923	0.928	0.826	0.796	0.681	0.639	0.746	0.709

Table B.8 Naive Bayes - Cumulative Distribution Results (1 Month Test)

Cumulation Months	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
1	0.913	0.899	0.728	0.66	0.762	0.724	0.745	0.691
2	0.91	0.898	0.708	0.634	0.78	0.786	0.742	0.702
3	0.919	0.909	0.756	0.732	0.754	0.712	0.755	0.722
4	0.894	0.882	0.64	0.564	0.823	0.804	0.72	0.663
5	0.901	0.896	0.676	0.673	0.782	0.807	0.726	0.734
6	0.902	0.888	0.677	0.657	0.787	0.72	0.73	0.687
7	0.872	0.873	0.582	0.584	0.817	0.759	0.68	0.66
8	0.914	0.915	0.695	0.72	0.862	0.893	0.769	0.797
9	0.911	0.918	0.689	0.709	0.852	0.869	0.762	0.781
10	0.914	0.914	0.694	0.692	0.861	0.84	0.769	0.759
11	0.91	0.924	0.686	0.74	0.849	0.888	0.759	0.807
12	0.888	0.882	0.61	0.603	0.902	0.883	0.727	0.717
13	0.911	0.92	0.686	0.72	0.849	0.872	0.759	0.789
14	0.91	0.911	0.679	0.626	0.867	0.86	0.762	0.725

Table B.9 Decision Tree - Cumulative Distribution Results (2 Month Test)

Cumulation Months	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
1	0.902	0.818	0.809	0.876	0.536	0.556	0.645	0.681
2	0.907	0.826	0.811	0.886	0.578	0.575	0.675	0.697
3	0.898	0.782	0.806	0.847	0.511	0.518	0.625	0.593
4	0.895	0.784	0.8	0.848	0.492	0.463	0.61	0.599
5	0.892	0.785	0.772	0.855	0.498	0.464	0.606	0.601
6	0.896	0.781	0.816	0.847	0.488	0.455	0.611	0.592
7	0.887	0.787	0.753	0.845	0.477	0.478	0.584	0.611
8	0.88	0.774	0.704	0.829	0.483	0.445	0.573	0.579
9	0.872	0.776	0.646	0.83	0.51	0.451	0.57	0.584
10	0.889	0.797	0.715	0.853	0.55	0.505	0.621	0.634
11	0.895	0.803	0.742	0.844	0.564	0.535	0.641	0.655
12	0.889	0.805	0.714	0.838	0.554	0.546	0.624	0.661
13	0.891	0.795	0.725	0.831	0.553	0.517	0.627	0.638

Table B.10 Random Forest - Cumulative Distribution Results (2 Month Test)

Cumulation Months	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
1	0.914	0.83	0.892	0.908	0.549	0.572	0.679	0.702
2	0.909	0.825	0.875	0.922	0.529	0.544	0.659	0.685
3	0.906	0.819	0.862	0.893	0.518	0.546	0.647	0.678
4	0.909	0.821	0.829	0.878	0.568	0.566	0.674	0.688
5	0.907	0.818	0.835	0.865	0.553	0.567	0.666	0.685
6	0.912	0.818	0.858	0.873	0.565	0.56	0.682	0.682
7	0.906	0.818	0.833	0.882	0.546	0.742	0.66	0.681
8	0.895	0.809	0.792	0.873	0.502	0.531	0.615	0.66
9	0.895	0.804	0.763	0.846	0.538	0.536	0.631	0.656
10	0.901	0.818	0.757	0.825	0.599	0.608	0.669	0.7
11	0.905	0.824	0.776	0.839	0.6	0.613	0.677	0.708
12	0.899	0.816	0.749	0.733	0.591	0.591	0.66	0.691
13	0.899	0.818	0.779	0.864	0.549	0.569	0.644	0.686

Table B.11 SVM - Cumulative Distribution Results (2 Month Test)

Cumulation Months	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
1	0.818	0.815	0.748	0.923	0.497	0.512	0.597	0.659
2	0.888	0.815	0.749	0.92	0.492	0.516	0.594	0.661
3	0.919	0.83	0.841	0.808	0.631	0.672	0.721	0.734
4	0.9	0.798	0.68	0.682	0.753	0.792	0.715	0.732
5	0.918	0.834	0.759	0.75	0.744	0.786	0.752	0.767
6	0.921	0.845	0.767	0.764	0.753	0.805	0.76	0.784
7	0.92	0.843	0.768	0.764	0.742	0.795	0.755	0.779
8	0.92	0.84	0.763	0.755	0.755	0.801	0.759	0.777
9	0.919	0.843	0.762	0.761	0.746	0.801	0.754	0.78
10	0.922	0.838	0.801	0.771	0.704	0.761	0.749	0.766
11	0.919	0.842	0.767	0.764	0.738	0.791	0.752	0.777
12	0.92	0.843	0.768	0.765	0.741	0.795	0.754	0.78
13	0.921	0.843	0.766	0.752	0.761	0.821	0.763	0.785

Table B.12 MLP - Cumulative Distribution Results (2 Month Test)

Cumulation Months	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
1	0.898	0.811	0.854	0.927	0.468	0.498	0.604	0.648
2	0.904	0.822	0.846	0.913	0.516	0.542	0.641	0.68
3	0.913	0.821	0.845	0.858	0.587	0.583	0.693	0.694
4	0.918	0.832	0.843	0.834	0.625	0.649	0.717	0.73
5	0.92	0.837	0.837	0.834	0.648	0.665	0.731	0.74
6	0.921	0.838	0.821	0.827	0.668	0.679	0.737	0.746
7	0.919	0.83	0.859	0.854	0.612	0.619	0.715	0.718
8	0.901	0.819	0.793	0.856	0.549	0.58	0.649	0.692
9	0.9	0.81	0.804	0.854	0.524	0.55	0.635	0.669
10	0.905	0.82	0.761	0.811	0.623	0.63	0.685	0.709
11	0.91	0.823	0.778	0.805	0.646	0.652	0.706	0.72
12	0.915	0.828	0.833	0.85	0.612	0.616	0.705	0.714
13	0.925	0.857	0.82	0.838	0.704	0.732	0.757	0.781

Table B.13 Naive Bayes - Cumulative Distribution Results (2 Month Test)

Cumulation Months	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
1	0.815	0.72	0.471	0.561	0.888	0.921	0.615	0.697
2	0.893	0.801	0.633	0.658	0.849	0.895	0.725	0.758
3	0.743	0.639	0.386	0.491	0.924	0.934	0.544	0.644
4	0.814	0.739	0.469	0.58	0.898	0.922	0.616	0.712
5	0.817	0.742	0.473	0.582	0.898	0.923	0.62	0.714
6	0.819	0.744	0.477	0.584	0.897	0.922	0.623	0.715
7	0.896	0.81	0.631	0.656	0.904	0.961	0.743	0.78
8	0.9	0.823	0.644	0.679	0.89	0.939	0.747	0.788
9	0.9	0.823	0.642	0.675	0.898	0.95	0.749	0.79
10	0.9	0.823	0.642	0.675	0.898	0.95	0.749	0.789
11	0.823	0.766	0.483	0.601	0.951	0.984	0.641	0.746
12	0.898	0.827	0.638	0.681	0.896	0.95	0.746	0.793
13	0.91	0.834	0.683	0.698	0.854	0.923	0.759	0.795

C SAMPLING EXPERIMENTS

Table C.1 Hybrid BRUS + Random Oversampling Results

Algorithm	Fraud/Legitimate Balance	Accuracy	Precision	Recall	F1 Score
Decision Tree	1:5 (Downsampled)	0.888	0.746	0.502	0.6
	1:25 (5xUpsampled)	0.894	0.877	0.417	0.566
	1:25 (12xUpsampled)	0.79	0.951	0.373	0.536
	1:25 (25xUpsampled)	0.686	0.977	0.383	0.55
	1:100 (20xUpsampled)	0.885	0.906	0.33	0.484
	1:100 (50xUpsampled)	0.752	0.957	0.255	0.403
Random Forest	1:5 (Downsampled)	0.903	0.826	0.533	0.648
	1:25 (5xUpsampled)	0.892	0.931	0.374	0.534
	1:25 (12xUpsampled)	0.799	0.968	0.393	0.559
	1:25 (25xUpsampled)	0.694	0.982	0.398	0.566
	1:100 (20xUpsampled)	0.882	0.923	0.304	0.458
	1:100 (50xUpsampled)	0.772	0.973	0.318	0.479
SVM	1:5 (Downsampled)	0.912	0.758	0.698	0.727
	1:25 (5xUpsampled)	0.896	0.869	0.434	0.579
	1:25 (12xUpsampled)	0.814	0.895	0.482	0.627
	1:25 (25xUpsampled)	0.738	0.814	0.619	0.703
	1:100 (20xUpsampled)	0.919	0.808	0.662	0.727
	1:100 (50xUpsampled)	0.877	0.82	0.802	0.811
MLP	1:5 (Downsampled)	0.915	0.83	0.62	0.71
	1:25 (5xUpsampled)	0.896	0.888	0.424	0.574
	1:25 (12xUpsampled)	0.79	0.905	0.395	0.55
	1:25 (25xUpsampled)	0.698	0.932	0.429	0.588
	1:100 (20xUpsampled)	0.898	0.811	0.487	0.608
	1:100 (50xUpsampled)	0.765	0.832	0.356	0.499
Naive Bayes	1:5 (Downsampled)	0.862	0.658	0.367	0.471
	1:25 (5xUpsampled)	0.851	0.571	0.393	0.465
	1:25 (12xUpsampled)	0.689	0.525	0.432	0.474
	1:25 (25xUpsampled)	0.59	0.634	0.432	0.513
	1:100 (20xUpsampled)	0.91	0.685	0.832	0.751
	1:100 (50xUpsampled)	0.82	0.65	0.981	0.781

Table C.2 Hybrid BRUS + SMOTE Results

Algorithm	Fraud/Legitimate Balance	Accuracy	Precision	Recall	F1 Score
Decision Tree	1:5 (Downsampled)	0.888	0.746	0.502	0.6
	1:25 (5XSMOTE)	0.963	0.52	0.445	0.48
	1:25 (12XSMOTE)	0.965	0.556	0.451	0.498
	1:25 (25XSMOTE)	0.963	0.529	0.451	0.486
	1:100 (20XSMOTE)	0.978	0.208	0.441	0.282
	1:100 (50XSMOTE)	0.975	0.181	0.443	0.257
Random Forest	1:5 (Downsampled)	0.903	0.826	0.533	0.648
	1:25 (5XSMOTE)	0.969	0.672	0.39	0.494
	1:25 (12XSMOTE)	0.97	0.703	0.391	0.503
	1:25 (25XSMOTE)	0.971	0.703	0.403	0.512
	1:100 (20XSMOTE)	0.982	0.23	0.352	0.278
	1:100 (50XSMOTE)	0.98	0.192	0.338	0.245
SVM	1:5 (Downsampled)	0.912	0.758	0.698	0.727
	1:25 (5XSMOTE)	0.974	0.824	0.414	0.551
	1:25 (12XSMOTE)	0.974	0.824	0.414	0.551
	1:25 (25XSMOTE)	0.974	0.824	0.414	0.551
	1:100 (20XSMOTE)	0.991	0.565	0.438	0.493
	1:100 (50XSMOTE)	0.991	0.565	0.438	0.493
MLP	1:5 (Downsampled)	0.915	0.83	0.62	0.71
	1:25 (5XSMOTE)	0.971	0.718	0.413	0.524
	1:25 (12XSMOTE)	0.971	0.728	0.393	0.511
	1:25 (25XSMOTE)	0.972	0.755	0.398	0.521
	1:100 (20XSMOTE)	0.983	0.261	0.428	0.324
	1:100 (50XSMOTE)	0.982	0.23	0.377	0.286
Naive Bayes	1:5 (Downsampled)	0.862	0.658	0.367	0.471
	1:25 (5XSMOTE)	0.939	0.27	0.346	0.303
	1:25 (12XSMOTE)	0.939	0.27	0.345	0.303
	1:25 (25XSMOTE)	0.939	0.27	0.345	0.303
	1:100 (20XSMOTE)	0.929	0.1	0.794	0.177
	1:100 (50XSMOTE)	0.929	0.1	0.794	0.178

Table C.3 Hybrid BRUS + SMOTENC Results

Algorithm	Fraud/Legitimate Balance	Accuracy	Precision	Recall	F1 Score
Decision Tree	1:5 (Downsampled)	0.888	0.746	0.502	0.6
	1:25 (5XSMOTENC)	0.893	0.77	0.511	0.614
	1:25 (12XSMOTENC)	0.887	0.726	0.524	0.608
	1:25 (25XSMOTENC)	0.878	0.692	0.491	0.574
	1:100 (20XSMOTENC)	0.885	0.776	0.435	0.558
	1:100 (50XSMOTENC)	0.879	0.74	0.426	0.541
Random Forest	1:5 (Downsampled)	0.903	0.826	0.533	0.648
	1:25 (5XSMOTENC)	0.887	0.812	0.421	0.555
	1:25 (12XSMOTENC)	0.886	0.777	0.445	0.565
	1:25 (25XSMOTENC)	0.885	0.777	0.44	0.562
	1:100 (20XSMOTENC)	0.882	0.826	0.374	0.515
	1:100 (50XSMOTENC)	0.883	0.801	0.4	0.533
SVM	1:5 (Downsampled)	0.912	0.758	0.698	0.727
	1:25 (5XSMOTENC)	0.897	0.668	0.767	0.714
	1:25 (12XSMOTENC)	0.859	0.549	0.872	0.673
	1:25 (25XSMOTENC)	0.838	0.509	0.897	0.649
	1:100 (20XSMOTENC)	0.899	0.672	0.777	0.72
	1:100 (50XSMOTENC)	0.864	0.56	0.866	0.68
MLP	1:5 (Downsampled)	0.915	0.83	0.62	0.71
	1:25 (5XSMOTENC)	0.904	0.787	0.588	0.673
	1:25 (12XSMOTENC)	0.905	0.728	0.689	0.708
	1:25 (25XSMOTENC)	0.891	0.63	0.842	0.721
	1:100 (20XSMOTENC)	0.914	0.8	0.644	0.713
	1:100 (50XSMOTENC)	0.899	0.697	0.699	0.698
Naive Bayes	1:5 (Downsampled)	0.862	0.658	0.367	0.471
	1:25 (5XSMOTENC)	0.899	0.737	0.616	0.671
	1:25 (12XSMOTENC)	0.869	0.565	0.934	0.704
	1:25 (25XSMOTENC)	0.863	0.555	0.903	0.688
	1:100 (20XSMOTENC)	0.909	0.691	0.826	0.753
	1:100 (50XSMOTENC)	0.867	0.565	0.934	0.704

D FEATURE SELECTION EXPERIMENTS

Table D.1 Decision Tree - CHI2 Feature Selection Results

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
20	0.883	0.766	0.736	0.828	0.470	0.406	0.574	0.545
25	0.893	0.785	0.777	0.863	0.502	0.449	0.610	0.591
30	0.893	0.789	0.775	0.857	0.504	0.467	0.611	0.604
35	0.896	0.794	0.787	0.868	0.520	0.478	0.626	0.616
40	0.897	0.798	0.796	0.891	0.514	0.474	0.624	0.619
45	0.899	0.800	0.794	0.869	0.533	0.495	0.638	0.630
50	0.899	0.801	0.778	0.858	0.552	0.506	0.646	0.637
55	0.905	0.816	0.788	0.873	0.586	0.545	0.672	0.671
60	0.902	0.806	0.790	0.867	0.563	0.518	0.658	0.648
65	0.898	0.795	0.769	0.856	0.554	0.489	0.644	0.623
70	0.895	0.788	0.758	0.852	0.547	0.465	0.635	0.602
75	0.893	0.780	0.751	0.841	0.536	0.447	0.626	0.584
1524	0.906	0.810	0.822	0.889	0.559	0.515	0.665	0.652

Table D.2 Random Forest - CHI2 Feature Selection Results

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
20	0.887	0.766	0.759	0.825	0.475	0.409	0.585	0.547
25	0.897	0.790	0.793	0.875	0.519	0.456	0.627	0.599
30	0.903	0.802	0.813	0.878	0.541	0.495	0.650	0.633
35	0.906	0.809	0.832	0.896	0.551	0.506	0.663	0.647
40	0.896	0.795	0.820	0.867	0.484	0.480	0.608	0.618
45	0.908	0.828	0.876	0.918	0.524	0.551	0.655	0.689
50	0.905	0.816	0.864	0.917	0.512	0.515	0.643	0.659
55	0.908	0.817	0.840	0.904	0.555	0.525	0.669	0.664
60	0.901	0.813	0.834	0.906	0.510	0.512	0.633	0.654
65	0.902	0.820	0.797	0.868	0.555	0.567	0.655	0.686
70	0.902	0.806	0.829	0.884	0.518	0.505	0.638	0.643
75	0.903	0.813	0.806	0.855	0.552	0.550	0.655	0.669
1524	0.920	0.838	0.933	0.947	0.559	0.564	0.699	0.707

Table D.3 SVM - CHI2 Feature Selection Results

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
20	0.887	0.764	0.849	0.913	0.390	0.349	0.535	0.505
25	0.904	0.815	0.712	0.717	0.711	0.766	0.712	0.741
30	0.910	0.834	0.748	0.768	0.697	0.745	0.721	0.756
35	0.906	0.830	0.747	0.729	0.662	0.809	0.702	0.767
40	0.920	0.859	0.779	0.810	0.730	0.772	0.754	0.791
45	0.913	0.847	0.768	0.753	0.689	0.829	0.726	0.790
50	0.927	0.864	0.794	0.793	0.764	0.818	0.778	0.806
55	0.936	0.864	0.790	0.802	0.762	0.806	0.776	0.804
60	0.920	0.853	0.765	0.755	0.754	0.851	0.760	0.800
65	0.915	0.844	0.740	0.734	0.756	0.858	0.748	0.791
70	0.911	0.813	0.814	0.805	0.604	0.604	0.693	0.691
75	0.912	0.839	0.754	0.741	0.699	0.823	0.726	0.780
1524	0.904	0.801	0.889	0.897	0.483	0.478	0.626	0.623

Table D.4 MLP - CHI2 Feature Selection Results

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
20	0.886	0.744	0.957	0.987	0.335	0.262	0.497	0.414
25	0.902	0.801	0.750	0.687	0.622	0.781	0.680	0.730
30	0.898	0.785	0.901	0.950	0.438	0.397	0.589	0.560
35	0.907	0.821	0.811	0.743	0.580	0.736	0.676	0.739
40	0.906	0.796	0.918	0.844	0.478	0.500	0.629	0.628
45	0.915	0.822	0.920	0.938	0.537	0.518	0.678	0.667
50	0.917	0.838	0.867	0.798	0.595	0.709	0.706	0.751
55	0.914	0.835	0.895	0.862	0.552	0.620	0.683	0.721
60	0.905	0.783	0.907	0.957	0.479	0.388	0.627	0.552
65	0.900	0.754	0.686	0.593	0.745	0.912	0.714	0.719
70	0.908	0.819	0.864	0.938	0.531	0.508	0.658	0.659
75	0.910	0.841	0.800	0.774	0.615	0.758	0.695	0.766
1524	0.894	0.786	0.922	0.940	0.398	0.407	0.556	0.568

Table D.5 Naive Bayes - CHI2 Feature Selection Results

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
20	0.885	0.735	0.724	0.630	0.505	0.562	0.595	0.594
25	0.880	0.761	0.657	0.656	0.589	0.650	0.621	0.653
30	0.855	0.744	0.556	0.615	0.654	0.691	0.601	0.651
35	0.888	0.777	0.687	0.684	0.600	0.660	0.640	0.672
40	0.896	0.781	0.728	0.707	0.598	0.622	0.657	0.662
45	0.901	0.800	0.751	0.741	0.610	0.645	0.673	0.690
50	0.900	0.799	0.734	0.724	0.631	0.675	0.679	0.699
55	0.900	0.796	0.738	0.721	0.625	0.670	0.677	0.694
60	0.903	0.806	0.736	0.724	0.654	0.705	0.693	0.715
65	0.901	0.801	0.720	0.713	0.666	0.708	0.692	0.710
70	0.902	0.800	0.722	0.712	0.672	0.707	0.696	0.710
75	0.896	0.785	0.707	0.696	0.642	0.667	0.673	0.681
1524	0.533	0.593	0.116	0.368	0.273	0.249	0.163	0.297

Table D.6 Decision Tree - PCA Feature Selection Results

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
20	0.860	0.768	0.605	0.726	0.469	0.525	0.528	0.609
25	0.879	0.765	0.680	0.710	0.521	0.535	0.589	0.610
30	0.867	0.765	0.628	0.722	0.498	0.514	0.555	0.601
35	0.879	0.773	0.684	0.735	0.511	0.534	0.585	0.619
40	0.884	0.783	0.692	0.739	0.545	0.572	0.610	0.645
45	0.876	0.753	0.697	0.718	0.458	0.464	0.553	0.564
50	0.882	0.768	0.683	0.714	0.543	0.543	0.605	0.617
55	0.886	0.794	0.704	0.742	0.546	0.616	0.615	0.673
60	0.858	0.740	0.603	0.677	0.446	0.466	0.512	0.552
65	0.865	0.777	0.666	0.760	0.545	0.512	0.574	0.612
70	0.873	0.772	0.665	0.732	0.478	0.534	0.556	0.617
75	0.860	0.766	0.601	0.723	0.490	0.518	0.540	0.604
1524	0.906	0.810	0.822	0.889	0.559	0.515	0.665	0.652

Table D.7 Random Forest - PCA Feature Selection Results

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
20	0.903	0.820	0.868	0.872	0.498	0.557	0.633	0.680
25	0.910	0.840	0.890	0.904	0.526	0.598	0.661	0.720
30	0.902	0.822	0.895	0.907	0.469	0.539	0.615	0.676
35	0.902	0.815	0.876	0.893	0.479	0.526	0.620	0.662
40	0.903	0.826	0.899	0.898	0.472	0.558	0.619	0.688
45	0.894	0.815	0.878	0.903	0.426	0.518	0.574	0.659
50	0.897	0.814	0.871	0.882	0.448	0.530	0.592	0.662
55	0.895	0.793	0.889	0.870	0.427	0.468	0.577	0.608
60	0.898	0.806	0.890	0.886	0.442	0.499	0.591	0.639
65	0.897	0.808	0.899	0.899	0.430	0.499	0.582	0.642
70	0.894	0.789	0.878	0.849	0.423	0.470	0.571	0.605
75	0.896	0.815	0.885	0.892	0.435	0.526	0.583	0.662
1524	0.920	0.838	0.933	0.947	0.559	0.564	0.699	0.707

Table D.8 SVM - PCA Feature Selection Results

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
20	0.907	0.818	0.758	0.722	0.652	0.764	0.701	0.743
25	0.917	0.838	0.781	0.750	0.698	0.792	0.738	0.771
30	0.918	0.843	0.785	0.756	0.703	0.803	0.741	0.779
35	0.919	0.840	0.788	0.754	0.701	0.793	0.742	0.773
40	0.917	0.831	0.785	0.748	0.691	0.767	0.735	0.758
45	0.919	0.850	0.812	0.792	0.667	0.764	0.732	0.778
50	0.914	0.845	0.802	0.799	0.644	0.735	0.715	0.766
55	0.914	0.852	0.815	0.820	0.630	0.729	0.711	0.772
60	0.922	0.856	0.801	0.784	0.709	0.801	0.752	0.792
65	0.916	0.851	0.819	0.812	0.638	0.738	0.717	0.773
70	0.914	0.844	0.818	0.813	0.621	0.711	0.706	0.759
75	0.912	0.850	0.828	0.834	0.597	0.703	0.693	0.703
1524	0.904	0.801	0.889	0.897	0.483	0.478	0.626	0.623

Table D.9 MLP - PCA Feature Selection Results

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
20	0.908	0.858	0.811	0.869	0.585	0.691	0.680	0.770
25	0.912	0.868	0.831	0.883	0.594	0.711	0.693	0.788
30	0.908	0.838	0.801	0.846	0.594	0.648	0.682	0.734
35	0.917	0.852	0.840	0.876	0.621	0.665	0.714	0.756
40	0.897	0.807	0.834	0.878	0.477	0.511	0.607	0.646
45	0.906	0.824	0.816	0.840	0.563	0.605	0.666	0.703
50	0.907	0.844	0.845	0.896	0.543	0.620	0.662	0.733
55	0.908	0.821	0.844	0.865	0.549	0.567	0.665	0.685
60	0.917	0.850	0.826	0.857	0.639	0.679	0.720	0.757
65	0.910	0.832	0.833	0.860	0.576	0.612	0.681	0.715
70	0.906	0.831	0.838	0.874	0.542	0.595	0.658	0.708
75	0.907	0.811	0.825	0.840	0.561	0.558	0.668	0.670
1524	0.894	0.786	0.922	0.940	0.398	0.407	0.556	0.568

Table D.10 Naive Bayes - PCA Feature Selection Results

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
20	0.819	0.658	0.465	0.502	0.548	0.752	0.503	0.602
25	0.813	0.653	0.453	0.497	0.566	0.766	0.503	0.603
30	0.812	0.660	0.450	0.503	0.548	0.756	0.494	0.604
35	0.817	0.667	0.464	0.510	0.595	0.788	0.521	0.619
40	0.808	0.651	0.442	0.495	0.563	0.766	0.495	0.602
45	0.811	0.661	0.449	0.505	0.574	0.787	0.504	0.615
50	0.794	0.636	0.414	0.482	0.561	0.767	0.476	0.592
55	0.782	0.618	0.383	0.464	0.493	0.695	0.431	0.556
60	0.782	0.613	0.377	0.457	0.469	0.671	0.418	0.544
65	0.776	0.599	0.354	0.442	0.417	0.631	0.383	0.520
70	0.768	0.589	0.335	0.432	0.402	0.618	0.366	0.508
75	0.768	0.564	0.337	0.406	0.405	0.580	0.368	0.478
1524	0.533	0.593	0.116	0.368	0.273	0.249	0.163	0.297

Table D.11 Decision Tree - AutoEncoder Feature Selection Results

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
20	0.874	0.780	0.660	0.404	0.511	0.553	0.576	0.635
25	0.857	0.771	0.585	0.717	0.490	0.555	0.533	0.626
30	0.855	0.726	0.592	0.666	0.419	0.412	0.491	0.509
35	0.869	0.761	0.652	0.729	0.462	0.488	0.541	0.585
40	0.872	0.761	0.657	0.720	0.486	0.503	0.558	0.592
45	0.868	0.781	0.633	0.739	0.502	0.563	0.560	0.639
50	0.865	0.766	0.641	0.712	0.439	0.541	0.521	0.615
55	0.856	0.754	0.594	0.696	0.428	0.507	0.497	0.587
60	0.867	0.775	0.640	0.741	0.471	0.537	0.543	0.623
65	0.866	0.769	0.645	0.719	0.441	0.543	0.524	0.619
70	0.861	0.750	0.630	0.712	0.413	0.464	0.499	0.562
75	0.870	0.777	0.655	0.756	0.474	0.520	0.550	0.617
1524	0.906	0.810	0.822	0.889	0.559	0.515	0.665	0.652

Table D.12 Random Forest - AutoEncoder Feature Selection Results

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
20	0.892	0.804	0.891	0.885	0.404	0.496	0.556	0.636
25	0.896	0.811	0.889	0.890	0.429	0.516	0.579	0.653
30	0.885	0.786	0.934	0.937	0.333	0.408	0.491	0.568
35	0.887	0.789	0.927	0.920	0.350	0.427	0.508	0.583
40	0.883	0.789	0.897	0.892	0.338	0.443	0.491	0.592
45	0.892	0.807	0.904	0.920	0.393	0.483	0.548	0.633
50	0.882	0.786	0.894	0.891	0.334	0.408	0.486	0.560
55	0.877	0.769	0.911	0.896	0.292	0.374	0.442	0.528
60	0.871	0.747	0.907	0.884	0.252	0.307	0.394	0.456
65	0.878	0.771	0.919	0.880	0.295	0.389	0.447	0.539
70	0.883	0.782	0.921	0.887	0.331	0.423	0.486	0.573
75	0.878	0.761	0.893	0.864	0.306	0.367	0.456	0.515
1524	0.920	0.838	0.933	0.947	0.559	0.564	0.699	0.707

Table D.13 SVM - AutoEncoder Feature Selection Results

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
20	0.907	0.814	0.807	0.754	0.580	0.683	0.675	0.717
25	0.906	0.843	0.746	0.756	0.659	0.807	0.700	0.781
30	0.916	0.855	0.845	0.821	0.612	0.741	0.710	0.779
35	0.923	0.859	0.810	0.775	0.702	0.834	0.752	0.803
40	0.930	0.875	0.846	0.814	0.714	0.827	0.774	0.820
45	0.929	0.885	0.827	0.814	0.729	0.863	0.775	0.838
50	0.922	0.860	0.822	0.794	0.680	0.804	0.744	0.799
55	0.924	0.871	0.826	0.805	0.694	0.826	0.754	0.815
60	0.931	0.887	0.847	0.824	0.716	0.854	0.776	0.839
65	0.930	0.875	0.857	0.822	0.700	0.814	0.771	0.814
70	0.931	0.876	0.843	0.816	0.712	0.830	0.777	0.823
75	0.927	0.873	0.838	0.809	0.697	0.827	0.761	0.818
1524	0.904	0.801	0.889	0.897	0.483	0.478	0.626	0.623

Table D.14 MLP - AutoEncoder Feature Selection Results

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
20	0.901	0.803	0.827	0.896	0.518	0.487	0.637	0.631
25	0.907	0.827	0.802	0.844	0.588	0.611	0.679	0.709
30	0.909	0.830	0.813	0.843	0.589	0.624	0.683	0.717
35	0.917	0.844	0.796	0.857	0.677	0.657	0.732	0.743
40	0.904	0.809	0.860	0.884	0.506	0.513	0.637	0.649
45	0.903	0.829	0.772	0.854	0.600	0.610	0.675	0.712
50	0.902	0.802	0.827	0.865	0.522	0.506	0.640	0.639
55	0.905	0.816	0.918	0.903	0.473	0.524	0.624	0.663
60	0.900	0.829	0.799	0.881	0.540	0.582	0.645	0.701
65	0.914	0.837	0.852	0.874	0.589	0.618	0.696	0.724
70	0.911	0.832	0.858	0.898	0.557	0.581	0.675	0.705
75	0.916	0.841	0.872	0.884	0.586	0.620	0.701	0.729
1524	0.894	0.786	0.922	0.940	0.398	0.407	0.556	0.568

Table D.15 Naive Bayes - AutoEncoder Feature Selection Results

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
20	0.884	0.697	0.617	0.535	0.821	0.941	0.704	0.682
25	0.867	0.676	0.566	0.517	0.872	0.958	0.687	0.671
30	0.871	0.697	0.574	0.534	0.878	0.966	0.694	0.687
35	0.867	0.691	0.568	0.529	0.852	0.941	0.682	0.678
40	0.863	0.692	0.560	0.531	0.836	0.930	0.671	0.676
45	0.875	0.689	0.591	0.528	0.826	0.927	0.689	0.673
50	0.867	0.705	0.567	0.541	0.871	0.953	0.687	0.690
55	0.873	0.687	0.587	0.527	0.809	0.913	0.680	0.668
60	0.856	0.689	0.546	0.528	0.840	0.937	0.662	0.676
65	0.838	0.721	0.509	0.558	0.829	0.921	0.631	0.695
70	0.846	0.712	0.525	0.549	0.834	0.925	0.644	0.689
75	0.866	0.697	0.577	0.537	0.749	0.884	0.652	0.668
1524	0.533	0.593	0.116	0.368	0.273	0.249	0.163	0.297

E
PROFILING EXPERIMENTS

Table E.1 Credit Card Profiling Test Results (Decision Tree)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.846	0.737	0.842	0.876	0.579	0.567	0.686	0.689
10	0.865	0.758	0.886	0.921	0.612	0.577	0.724	0.71
15	0.848	0.752	0.881	0.92	0.55	0.564	0.677	0.699
20	0.847	0.747	0.897	0.931	0.535	0.546	0.67	0.688
23	0.838	0.727	0.89	0.919	0.505	0.512	0.644	0.658
25	0.84	0.728	0.898	0.926	0.507	0.51	0.648	0.658
30	0.843	0.738	0.897	0.932	0.52	0.526	0.659	0.673
35	0.847	0.742	0.873	0.93	0.553	0.537	0.677	0.681
40	0.857	0.754	0.876	0.929	0.591	0.563	0.706	0.701
45	0.858	0.752	0.905	0.932	0.571	0.556	0.701	0.696
50	0.851	0.75	0.889	0.932	0.559	0.552	0.686	0.694
55	0.851	0.753	0.882	0.94	0.561	0.553	0.686	0.696
60	0.86	0.772	0.9	0.936	0.583	0.594	0.708	0.727
65	0.864	0.776	0.909	0.943	0.593	0.6	0.718	0.733
70	0.853	0.78	0.841	0.94	0.608	0.609	0.706	0.739
75	0.858	0.772	0.855	0.94	0.614	0.592	0.715	0.726
100	0.859	0.793	0.845	0.945	0.629	0.633	0.721	0.758
150	0.872	0.775	0.916	0.943	0.615	0.597	0.736	0.731
300	0.877	0.79	0.917	0.942	0.633	0.629	0.749	0.754
1513	0.867	0.748	0.928	0.947	0.587	0.537	0.719	0.685

Table E.2 Credit Card Profiling Test Results (Random Forest)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.858	0.76	0.893	0.891	0.58	0.604	0.703	0.72
10	0.888	0.796	0.923	0.928	0.671	0.651	0.777	0.765
15	0.891	0.799	0.926	0.932	0.679	0.656	0.784	0.77
20	0.894	0.803	0.944	0.943	0.675	0.655	0.787	0.773
23	0.895	0.804	0.948	0.955	0.674	0.648	0.788	0.772
25	0.892	0.806	0.948	0.955	0.666	0.651	0.782	0.774
30	0.895	0.807	0.945	0.951	0.679	0.657	0.791	0.777
35	0.89	0.79	0.948	0.955	0.659	0.62	0.777	0.752
40	0.885	0.789	0.95	0.958	0.638	0.615	0.764	0.749
45	0.89	0.803	0.957	0.956	0.651	0.644	0.775	0.77
50	0.901	0.818	0.962	0.96	0.688	0.672	0.802	0.791
55	0.901	0.82	0.958	0.958	0.688	0.677	0.801	0.793
60	0.898	0.817	0.959	0.958	0.677	0.672	0.794	0.79
65	0.902	0.82	0.964	0.969	0.688	0.67	0.803	0.792
70	0.904	0.825	0.961	0.964	0.696	0.684	0.807	0.801
75	0.885	0.79	0.961	0.965	0.629	0.612	0.76	0.749
100	0.893	0.812	0.971	0.978	0.649	0.647	0.778	0.779
150	0.904	0.828	0.976	0.977	0.686	0.679	0.806	0.802
300	0.895	0.811	0.978	0.973	0.653	0.649	0.783	0.779
1513	0.873	0.781	0.976	0.973	0.577	0.589	0.725	0.734

Table E.3 Credit Card Profiling Test Results (SVM)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.73	0.54	0.985	0.999	0.07	0.102	0.13	0.184
10	0.828	0.684	0.993	0.996	0.412	0.385	0.582	0.555
15	0.828	0.684	0.993	0.996	0.412	0.385	0.582	0.555
20	0.828	0.684	0.993	0.996	0.412	0.385	0.582	0.555
23	0.828	0.684	0.993	0.996	0.412	0.385	0.582	0.555
25	0.828	0.684	0.993	0.996	0.412	0.385	0.582	0.555
30	0.847	0.726	0.944	0.921	0.502	0.508	0.655	0.655
35	0.847	0.726	0.945	0.921	0.502	0.508	0.656	0.655
40	0.853	0.728	0.945	0.907	0.523	0.523	0.673	0.664
45	0.858	0.754	0.948	0.912	0.541	0.575	0.689	0.706
50	0.859	0.75	0.928	0.918	0.558	0.561	0.697	0.697
55	0.862	0.758	0.933	0.918	0.564	0.578	0.703	0.71
60	0.861	0.757	0.94	0.925	0.557	0.573	0.7	0.707
65	0.859	0.753	0.946	0.936	0.545	0.556	0.692	0.698
70	0.86	0.756	0.945	0.934	0.55	0.563	0.695	0.703
75	0.86	0.757	0.948	0.939	0.549	0.563	0.695	0.704
100	0.859	0.752	0.948	0.939	0.544	0.552	0.691	0.695
150	0.859	0.754	0.947	0.934	0.544	0.559	0.691	0.699
300	0.858	0.747	0.944	0.921	0.543	0.553	0.69	0.691
1513	0.858	0.765	0.937	0.936	0.548	0.582	0.691	0.717

Table E.4 Credit Card Profiling Test Results (MLP)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.847	0.716	0.937	0.921	0.507	0.488	0.658	0.638
10	0.864	0.772	0.87	0.93	0.624	0.599	0.727	0.729
15	0.884	0.798	0.942	0.931	0.639	0.654	0.762	0.768
20	0.876	0.78	0.947	0.94	0.609	0.609	0.741	0.739
23	0.864	0.751	0.93	0.919	0.576	0.564	0.711	0.699
25	0.872	0.774	0.931	0.925	0.605	0.607	0.734	0.733
30	0.882	0.798	0.953	0.946	0.626	0.643	0.756	0.766
35	0.873	0.787	0.946	0.938	0.595	0.624	0.731	0.75
40	0.883	0.812	0.955	0.949	0.626	0.668	0.756	0.784
45	0.878	0.811	0.948	0.946	0.615	0.669	0.746	0.784
50	0.883	0.803	0.942	0.931	0.636	0.665	0.759	0.776
55	0.893	0.809	0.95	0.943	0.666	0.667	0.783	0.781
60	0.882	0.798	0.955	0.945	0.624	0.642	0.755	0.765
65	0.9	0.836	0.951	0.944	0.69	0.723	0.8	0.818
70	0.898	0.83	0.938	0.938	0.696	0.715	0.799	0.812
75	0.889	0.809	0.951	0.951	0.652	0.66	0.773	0.78
100	0.889	0.809	0.946	0.956	0.655	0.657	0.774	0.779
150	0.874	0.792	0.943	0.953	0.604	0.624	0.737	0.755
300	0.863	0.741	0.954	0.965	0.555	0.512	0.702	0.669
1513	0.883	0.805	0.954	0.958	0.629	0.647	0.758	0.773

Table E.5 Credit Card Profiling Test Results (Naive Bayes)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.467	0.576	0.35	0.547	0.975	0.994	0.515	0.706
10	0.864	0.75	0.812	0.75	0.694	0.767	0.748	0.758
15	0.878	0.783	0.827	0.774	0.735	0.814	0.778	0.794
20	0.888	0.805	0.772	0.742	0.87	0.947	0.818	0.832
23	0.877	0.793	0.768	0.747	0.827	0.902	0.796	0.817
25	0.86	0.782	0.699	0.713	0.908	0.962	0.79	0.819
30	0.874	0.822	0.716	0.749	0.939	0.98	0.812	0.849
35	0.868	0.817	0.709	0.748	0.925	0.97	0.803	0.845
40	0.879	0.829	0.73	0.76	0.928	0.974	0.817	0.854
45	0.881	0.823	0.739	0.761	0.91	0.955	0.816	0.847
50	0.906	0.851	0.814	0.807	0.876	0.933	0.844	0.865
55	0.898	0.838	0.815	0.807	0.839	0.897	0.827	0.85
60	0.897	0.834	0.817	0.808	0.832	0.888	0.824	0.846
65	0.895	0.833	0.807	0.798	0.841	0.901	0.824	0.847
70	0.898	0.84	0.805	0.8	0.857	0.918	0.83	0.855
75	0.901	0.844	0.8	0.794	0.88	0.939	0.838	0.86
100	0.885	0.813	0.772	0.769	0.857	0.909	0.812	0.833
150	0.863	0.766	0.743	0.737	0.807	0.842	0.774	0.786
300	0.846	0.727	0.729	0.722	0.748	0.761	0.738	0.741
1513	0.639	0.634	0.425	0.63	0.692	0.692	0.527	0.659

Table E.6 PNR Profiling Test Results (Decision Tree)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.864	0.773	0.873	0.903	0.622	0.623	0.727	0.737
10	0.851	0.765	0.864	0.903	0.578	0.606	0.693	0.725
15	0.865	0.78	0.885	0.928	0.616	0.618	0.726	0.742
20	0.856	0.763	0.902	0.929	0.566	0.581	0.696	0.715
23	0.847	0.739	0.906	0.929	0.529	0.53	0.668	0.675
25	0.85	0.749	0.904	0.923	0.54	0.557	0.676	0.695
30	0.843	0.741	0.891	0.917	0.522	0.543	0.658	0.682
35	0.848	0.748	0.895	0.927	0.541	0.551	0.674	0.691
40	0.854	0.767	0.904	0.939	0.556	0.582	0.689	0.719
45	0.854	0.768	0.906	0.94	0.557	0.584	0.689	0.72
50	0.843	0.749	0.902	0.926	0.514	0.553	0.655	0.692
55	0.841	0.743	0.9	0.917	0.51	0.546	0.651	0.685
60	0.848	0.751	0.907	0.923	0.532	0.561	0.671	0.698
65	0.851	0.757	0.904	0.921	0.546	0.573	0.681	0.707
70	0.841	0.743	0.896	0.92	0.512	0.546	0.652	0.685
75	0.847	0.752	0.903	0.927	0.529	0.56	0.667	0.699
100	0.853	0.77	0.908	0.933	0.549	0.594	0.684	0.726
150	0.845	0.745	0.898	0.935	0.527	0.54	0.664	0.684
300	0.842	0.742	0.877	0.922	0.531	0.543	0.661	0.683
1512	0.846	0.741	0.913	0.934	0.518	0.532	0.661	0.678

Table E.7 PNR Profiling Test Results (Random Forest)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.884	0.787	0.906	0.901	0.669	0.656	0.77	0.759
10	0.876	0.784	0.911	0.919	0.634	0.633	0.747	0.75
15	0.889	0.795	0.917	0.931	0.679	0.647	0.78	0.763
20	0.892	0.807	0.935	0.948	0.674	0.659	0.783	0.777
23	0.879	0.782	0.939	0.946	0.623	0.608	0.749	0.74
25	0.884	0.794	0.939	0.949	0.642	0.632	0.762	0.759
30	0.887	0.798	0.934	0.946	0.658	0.643	0.772	0.765
35	0.888	0.802	0.942	0.957	0.653	0.642	0.772	0.768
40	0.864	0.752	0.942	0.956	0.567	0.54	0.708	0.69
45	0.89	0.809	0.954	0.962	0.652	0.653	0.775	0.778
50	0.898	0.819	0.958	0.962	0.678	0.672	0.794	0.791
55	0.897	0.816	0.955	0.961	0.678	0.668	0.793	0.788
60	0.886	0.806	0.953	0.957	0.638	0.65	0.764	0.774
65	0.894	0.813	0.962	0.964	0.66	0.659	0.783	0.783
70	0.889	0.815	0.957	0.963	0.647	0.663	0.773	0.786
75	0.892	0.81	0.957	0.966	0.659	0.651	0.78	0.778
100	0.897	0.819	0.966	0.971	0.668	0.666	0.79	0.791
150	0.894	0.819	0.976	0.982	0.652	0.659	0.782	0.789
300	0.887	0.812	0.974	0.977	0.629	0.648	0.764	0.779
1512	0.876	0.786	0.987	0.988	0.58	0.59	0.73	0.739

Table E.8 PNR Profiling Test Results (SVM)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.814	0.665	0.993	0.995	0.362	0.348	0.531	0.516
10	0.814	0.665	0.993	0.995	0.362	0.348	0.531	0.516
15	0.823	0.681	0.993	0.996	0.394	0.379	0.564	0.549
20	0.823	0.681	0.993	0.996	0.394	0.379	0.564	0.549
23	0.823	0.681	0.993	0.996	0.394	0.379	0.564	0.549
25	0.823	0.681	0.993	0.996	0.394	0.379	0.564	0.549
30	0.833	0.704	0.923	0.913	0.465	0.467	0.618	0.618
35	0.833	0.706	0.927	0.91	0.461	0.472	0.616	0.621
40	0.843	0.729	0.928	0.901	0.499	0.529	0.649	0.666
45	0.847	0.735	0.927	0.912	0.513	0.533	0.661	0.673
50	0.848	0.738	0.921	0.91	0.52	0.54	0.665	0.678
55	0.852	0.755	0.921	0.914	0.536	0.576	0.678	0.706
60	0.851	0.75	0.919	0.91	0.534	0.567	0.676	0.698
65	0.854	0.76	0.914	0.907	0.548	0.592	0.685	0.716
70	0.856	0.764	0.917	0.908	0.553	0.6	0.69	0.723
75	0.858	0.766	0.909	0.894	0.568	0.616	0.699	0.73
100	0.857	0.759	0.911	0.897	0.562	0.598	0.695	0.717
150	0.853	0.747	0.932	0.916	0.531	0.556	0.677	0.692
300	0.853	0.743	0.931	0.91	0.533	0.553	0.678	0.688
1512	0.85	0.753	0.931	0.933	0.522	0.558	0.669	0.699

Table E.9 PNR Profiling Test Results (MLP)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.882	0.782	0.926	0.905	0.647	0.641	0.762	0.75
10	0.862	0.736	0.954	0.952	0.552	0.51	0.699	0.664
15	0.878	0.763	0.926	0.91	0.632	0.595	0.751	0.72
20	0.856	0.735	0.93	0.908	0.545	0.537	0.687	0.675
23	0.871	0.774	0.943	0.934	0.592	0.601	0.728	0.731
25	0.85	0.718	0.921	0.907	0.528	0.501	0.671	0.645
30	0.868	0.789	0.931	0.929	0.59	0.637	0.722	0.756
35	0.87	0.791	0.933	0.93	0.594	0.64	0.726	0.758
40	0.873	0.791	0.946	0.953	0.595	0.622	0.731	0.753
45	0.882	0.796	0.926	0.918	0.646	0.66	0.761	0.768
50	0.866	0.756	0.933	0.933	0.579	0.563	0.714	0.702
55	0.885	0.809	0.954	0.95	0.636	0.662	0.763	0.78
60	0.866	0.791	0.935	0.936	0.58	0.636	0.716	0.757
65	0.883	0.812	0.943	0.934	0.637	0.681	0.76	0.787
70	0.884	0.812	0.938	0.933	0.642	0.683	0.763	0.788
75	0.875	0.776	0.931	0.921	0.617	0.614	0.742	0.737
100	0.874	0.792	0.933	0.948	0.611	0.628	0.739	0.756
150	0.885	0.812	0.938	0.953	0.645	0.664	0.764	0.783
300	0.888	0.812	0.932	0.946	0.663	0.672	0.775	0.786
1512	0.849	0.726	0.964	0.964	0.497	0.481	0.656	0.642

Table E.10 PNR Profiling Test Results (Naive Bayes)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.812	0.673	0.88	0.743	0.408	0.552	0.558	0.634
10	0.84	0.719	0.784	0.73	0.62	0.714	0.693	0.722
15	0.853	0.749	0.828	0.78	0.625	0.712	0.712	0.744
20	0.855	0.765	0.794	0.766	0.679	0.78	0.732	0.773
23	0.858	0.768	0.77	0.75	0.726	0.819	0.747	0.783
25	0.863	0.778	0.781	0.759	0.734	0.832	0.757	0.794
30	0.846	0.799	0.687	0.742	0.862	0.932	0.764	0.826
35	0.85	0.802	0.694	0.743	0.867	0.936	0.771	0.829
40	0.86	0.815	0.709	0.754	0.88	0.948	0.785	0.84
45	0.883	0.834	0.796	0.803	0.805	0.895	0.8	0.847
50	0.881	0.831	0.8	0.811	0.787	0.874	0.794	0.841
55	0.88	0.826	0.798	0.807	0.785	0.869	0.791	0.837
60	0.882	0.826	0.803	0.807	0.787	0.868	0.795	0.837
65	0.882	0.826	0.802	0.805	0.79	0.872	0.796	0.837
70	0.881	0.823	0.793	0.795	0.801	0.884	0.797	0.837
75	0.885	0.83	0.79	0.792	0.825	0.907	0.807	0.845
100	0.87	0.803	0.752	0.762	0.826	0.895	0.787	0.823
150	0.833	0.738	0.675	0.7	0.816	0.856	0.739	0.77
300	0.819	0.707	0.671	0.694	0.739	0.765	0.703	0.728
1512	0.571	0.612	0.372	0.61	0.693	0.67	0.484	0.639

Table E.11 Record Profiling Test Results (Decision Tree)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.824	0.715	0.868	0.93	0.466	0.48	0.606	0.633
10	0.829	0.711	0.881	0.936	0.474	0.467	0.617	0.623
15	0.822	0.714	0.879	0.926	0.45	0.481	0.596	0.633
20	0.823	0.726	0.886	0.931	0.447	0.501	0.594	0.652
23	0.845	0.759	0.905	0.955	0.52	0.556	0.661	0.703
25	0.849	0.76	0.908	0.954	0.533	0.559	0.672	0.705
30	0.845	0.755	0.903	0.944	0.524	0.555	0.663	0.699
35	0.849	0.761	0.913	0.948	0.531	0.564	0.671	0.707
40	0.851	0.742	0.914	0.938	0.536	0.532	0.676	0.679
45	0.846	0.745	0.9	0.946	0.529	0.532	0.667	0.681
50	0.848	0.74	0.91	0.942	0.528	0.523	0.668	0.673
55	0.852	0.753	0.914	0.945	0.54	0.55	0.679	0.695
60	0.856	0.753	0.902	0.953	0.567	0.545	0.696	0.693
65	0.858	0.768	0.915	0.946	0.562	0.581	0.697	0.719
70	0.866	0.774	0.918	0.941	0.591	0.595	0.719	0.729
75	0.864	0.765	0.915	0.956	0.585	0.567	0.714	0.712
100	0.859	0.74	0.935	0.945	0.553	0.523	0.695	0.674
150	0.84	0.74	0.852	0.943	0.543	0.523	0.663	0.673
300	0.837	0.745	0.841	0.939	0.541	0.537	0.659	0.683
1505	0.869	0.821	0.886	0.945	0.63	0.691	0.737	0.799

Table E.12 Record Profiling Test Results (Random Forest)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.839	0.746	0.866	0.884	0.527	0.58	0.655	0.7
10	0.852	0.747	0.886	0.908	0.561	0.563	0.687	0.695
15	0.846	0.725	0.91	0.92	0.521	0.506	0.663	0.653
20	0.837	0.711	0.925	0.938	0.477	0.467	0.629	0.624
23	0.852	0.728	0.932	0.939	0.53	0.501	0.676	0.653
25	0.853	0.729	0.93	0.945	0.536	0.5	0.68	0.654
30	0.858	0.732	0.935	0.944	0.549	0.506	0.691	0.659
35	0.86	0.732	0.953	0.948	0.546	0.504	0.694	0.658
40	0.862	0.737	0.952	0.953	0.552	0.512	0.699	0.666
45	0.86	0.742	0.952	0.955	0.546	0.521	0.694	0.674
50	0.864	0.748	0.949	0.953	0.563	0.534	0.707	0.684
55	0.864	0.75	0.952	0.957	0.561	0.535	0.706	0.686
60	0.87	0.752	0.952	0.959	0.582	0.539	0.722	0.69
65	0.874	0.771	0.953	0.957	0.595	0.579	0.732	0.721
70	0.86	0.75	0.952	0.961	0.546	0.534	0.694	0.686
75	0.871	0.771	0.959	0.966	0.581	0.573	0.723	0.719
100	0.867	0.779	0.962	0.966	0.564	0.588	0.711	0.731
150	0.856	0.736	0.959	0.96	0.528	0.504	0.681	0.661
300	0.876	0.769	0.961	0.961	0.596	0.571	0.736	0.716
1505	0.869	0.764	0.97	0.966	0.565	0.559	0.714	0.708

Table E.13 Record Profiling Test Results (SVM)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.876	0.786	0.92	0.913	0.627	0.644	0.746	0.755
10	0.878	0.789	0.936	0.922	0.622	0.643	0.748	0.757
15	0.888	0.814	0.91	0.902	0.68	0.715	0.779	0.798
20	0.877	0.78	0.905	0.891	0.643	0.651	0.752	0.752
23	0.903	0.838	0.918	0.915	0.731	0.752	0.814	0.826
25	0.903	0.838	0.918	0.916	0.731	0.752	0.814	0.826
30	0.905	0.841	0.914	0.914	0.741	0.761	0.819	0.83
35	0.906	0.841	0.918	0.921	0.744	0.754	0.822	0.829
40	0.915	0.878	0.93	0.932	0.764	0.822	0.839	0.873
45	0.894	0.811	0.925	0.919	0.69	0.693	0.791	0.79
50	0.892	0.808	0.925	0.918	0.683	0.686	0.786	0.785
55	0.892	0.808	0.925	0.919	0.683	0.686	0.786	0.786
60	0.892	0.808	0.925	0.919	0.683	0.686	0.786	0.786
65	0.905	0.857	0.931	0.93	0.725	0.778	0.815	0.847
70	0.91	0.87	0.931	0.935	0.746	0.801	0.828	0.863
75	0.912	0.87	0.931	0.935	0.753	0.802	0.833	0.863
100	0.914	0.89	0.936	0.942	0.755	0.836	0.836	0.886
150	0.903	0.849	0.933	0.933	0.718	0.759	0.811	0.837
300	0.899	0.83	0.939	0.932	0.699	0.721	0.801	0.813
1505	0.89	0.82	0.954	0.957	0.653	0.679	0.775	0.795

Table E.14 Record Profiling Test Results (MLP)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.88	0.795	0.908	0.898	0.654	0.677	0.76	0.772
10	0.894	0.816	0.919	0.916	0.695	0.706	0.792	0.797
15	0.885	0.81	0.931	0.93	0.653	0.681	0.768	0.786
20	0.879	0.787	0.927	0.919	0.635	0.64	0.753	0.754
23	0.888	0.806	0.925	0.93	0.668	0.67	0.776	0.779
25	0.889	0.809	0.925	0.927	0.673	0.68	0.779	0.785
30	0.884	0.8	0.939	0.934	0.641	0.657	0.762	0.771
35	0.881	0.809	0.945	0.939	0.628	0.67	0.755	0.782
40	0.889	0.813	0.932	0.942	0.665	0.676	0.776	0.787
45	0.889	0.812	0.941	0.934	0.659	0.682	0.775	0.788
50	0.892	0.83	0.937	0.939	0.675	0.714	0.785	0.812
55	0.907	0.861	0.918	0.933	0.746	0.785	0.823	0.852
60	0.893	0.815	0.935	0.944	0.678	0.679	0.786	0.79
65	0.892	0.825	0.933	0.942	0.676	0.702	0.784	0.804
70	0.863	0.757	0.939	0.945	0.565	0.557	0.706	0.701
75	0.88	0.795	0.929	0.953	0.635	0.631	0.754	0.759
100	0.882	0.791	0.931	0.939	0.643	0.634	0.76	0.757
150	0.858	0.737	0.95	0.956	0.539	0.509	0.687	0.664
300	0.864	0.75	0.945	0.955	0.563	0.536	0.706	0.687
1505	0.86	0.76	0.969	0.97	0.536	0.548	0.69	0.7

Table E.15 Record Profiling Test Results (Naive Bayes)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.772	0.799	0.561	0.718	0.995	0.999	0.717	0.836
10	0.852	0.871	0.669	0.805	0.969	0.987	0.792	0.887
15	0.871	0.866	0.704	0.799	0.96	0.987	0.812	0.883
20	0.874	0.883	0.714	0.823	0.947	0.982	0.814	0.895
23	0.864	0.892	0.689	0.833	0.973	0.987	0.807	0.904
25	0.869	0.896	0.696	0.838	0.973	0.987	0.811	0.906
30	0.877	0.898	0.711	0.841	0.97	0.986	0.821	0.908
35	0.89	0.901	0.738	0.846	0.963	0.986	0.836	0.911
40	0.911	0.912	0.787	0.865	0.95	0.981	0.861	0.919
45	0.925	0.914	0.826	0.87	0.94	0.979	0.879	0.921
50	0.925	0.915	0.835	0.874	0.923	0.974	0.877	0.921
55	0.926	0.915	0.839	0.875	0.922	0.974	0.878	0.922
60	0.924	0.912	0.838	0.872	0.917	0.97	0.876	0.918
65	0.919	0.91	0.811	0.863	0.941	0.98	0.871	0.918
70	0.91	0.906	0.782	0.852	0.958	0.987	0.861	0.915
75	0.909	0.903	0.779	0.848	0.96	0.987	0.86	0.912
100	0.872	0.85	0.716	0.805	0.927	0.932	0.808	0.864
150	0.833	0.786	0.661	0.756	0.871	0.861	0.752	0.805
300	0.821	0.764	0.651	0.757	0.823	0.793	0.727	0.775
1505	0.783	0.744	0.612	0.795	0.696	0.672	0.651	0.729

Table E.16 Only Profiling Mode Test Results (Decision Tree)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.748	0.585	0.684	0.763	0.247	0.275	0.363	0.404
10	0.852	0.74	0.847	0.874	0.597	0.575	0.7	0.693
15	0.875	0.77	0.867	0.9	0.673	0.619	0.758	0.733
20	0.88	0.769	0.889	0.908	0.669	0.61	0.764	0.73
23	0.881	0.776	0.881	0.894	0.682	0.638	0.769	0.745

Table E.17 Only Profiling Mode Test Results (Random Forest)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.762	0.603	0.732	0.766	0.286	0.324	0.411	0.455
10	0.875	0.788	0.877	0.881	0.661	0.679	0.754	0.767
15	0.892	0.819	0.901	0.913	0.704	0.714	0.79	0.801
20	0.9	0.834	0.91	0.921	0.726	0.738	0.808	0.819
23	0.912	0.845	0.941	0.942	0.744	0.742	0.831	0.83

Table E.18 Only Profiling Mode Test Results (SVM)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.728	0.536	0.995	0.999	0.064	0.093	0.121	0.171
10	0.748	0.543	0.992	0.998	0.135	0.108	0.237	0.195
15	0.739	0.544	0.968	0.869	0.106	0.129	0.191	0.225
20	0.746	0.574	0.958	0.894	0.133	0.19	0.233	0.314
23	0.751	0.592	0.954	0.882	0.151	0.234	0.261	0.37

Table E.19 Only Profiling Mode Test Results (MLP)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.841	0.724	0.915	0.887	0.501	0.528	0.647	0.662
10	0.871	0.778	0.947	0.934	0.587	0.61	0.725	0.738
15	0.902	0.829	0.924	0.915	0.722	0.734	0.81	0.815
20	0.9	0.829	0.925	0.914	0.713	0.735	0.805	0.814
23	0.899	0.822	0.926	0.93	0.707	0.705	0.802	0.802

Table E.20 Only Profiling Mode Test Results (Naive Bayes)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.433	0.561	0.331	0.539	0.934	0.98	0.489	0.696
10	0.572	0.608	0.393	0.571	0.875	0.937	0.543	0.71
15	0.815	0.807	0.623	0.74	0.915	0.96	0.742	0.836
20	0.82	0.811	0.628	0.74	0.927	0.972	0.749	0.84
23	0.822	0.81	0.632	0.739	0.926	0.971	0.752	0.839

Table E.21 Multi-Modal Profiling Test Results (Decision Tree)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.747	0.588	0.668	0.768	0.253	0.279	0.367	0.410
10	0.861	0.766	0.856	0.888	0.626	0.620	0.723	0.731
15	0.871	0.759	0.895	0.916	0.631	0.584	0.740	0.713
20	0.872	0.758	0.889	0.906	0.640	0.588	0.744	0.713
23	0.860	0.760	0.907	0.919	0.576	0.582	0.704	0.712
25	0.872	0.782	0.923	0.945	0.610	0.610	0.734	0.741
30	0.863	0.765	0.923	0.946	0.578	0.574	0.711	0.715
35	0.865	0.764	0.927	0.946	0.580	0.571	0.714	0.712
40	0.853	0.732	0.906	0.936	0.550	0.511	0.685	0.661
45	0.877	0.771	0.936	0.946	0.619	0.586	0.745	0.724
50	0.869	0.779	0.920	0.947	0.601	0.603	0.727	0.737
55	0.877	0.782	0.933	0.936	0.620	0.616	0.745	0.745
60	0.876	0.788	0.930	0.942	0.621	0.623	0.745	0.750
65	0.870	0.771	0.916	0.949	0.609	0.584	0.732	0.723
70	0.879	0.773	0.931	0.935	0.632	0.598	0.752	0.729
75	0.882	0.776	0.928	0.938	0.644	0.603	0.760	0.734
100	0.870	0.780	0.913	0.939	0.609	0.609	0.731	0.739
150	0.880	0.796	0.929	0.943	0.634	0.640	0.754	0.762
300	0.848	0.735	0.847	0.939	0.583	0.517	0.690	0.666
1524	0.882	0.850	0.932	0.961	0.641	0.736	0.760	0.834

Table E.22 Multi-Modal Profiling Test Results (Random Forest)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.762	0.603	0.732	0.766	0.286	0.324	0.411	0.455
10	0.879	0.789	0.879	0.885	0.677	0.676	0.765	0.766
15	0.912	0.821	0.919	0.924	0.764	0.709	0.834	0.802
20	0.913	0.844	0.922	0.937	0.764	0.746	0.836	0.831
23	0.923	0.858	0.938	0.939	0.786	0.773	0.855	0.848
25	0.924	0.863	0.946	0.946	0.783	0.777	0.860	0.853
30	0.925	0.860	0.955	0.956	0.779	0.762	0.858	0.848
35	0.911	0.832	0.952	0.952	0.731	0.708	0.827	0.812
40	0.929	0.869	0.959	0.958	0.790	0.779	0.866	0.859
45	0.921	0.865	0.960	0.964	0.760	0.764	0.849	0.852
50	0.923	0.857	0.967	0.964	0.760	0.749	0.851	0.843
55	0.925	0.869	0.964	0.966	0.771	0.771	0.857	0.858
60	0.911	0.844	0.963	0.963	0.721	0.722	0.825	0.825
65	0.925	0.863	0.966	0.966	0.767	0.759	0.855	0.850
70	0.928	0.873	0.969	0.966	0.779	0.780	0.863	0.863
75	0.916	0.869	0.965	0.964	0.739	0.773	0.837	0.858
100	0.921	0.860	0.971	0.975	0.749	0.745	0.846	0.845
150	0.923	0.862	0.971	0.972	0.756	0.752	0.851	0.848
300	0.918	0.858	0.978	0.979	0.735	0.738	0.839	0.842
1524	0.901	0.832	0.981	0.979	0.671	0.687	0.797	0.808

Table E.23 Multi-Modal Profiling Test Results (SVM)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.728	0.536	0.995	0.999	0.064	0.093	0.121	0.171
10	0.814	0.665	0.992	0.995	0.362	0.348	0.53	0.516
15	0.881	0.8	0.926	0.917	0.642	0.671	0.758	0.775
20	0.881	0.8	0.926	0.917	0.641	0.669	0.757	0.774
23	0.881	0.809	0.918	0.909	0.648	0.696	0.76	0.788
25	0.884	0.814	0.919	0.909	0.66	0.708	0.768	0.796
30	0.874	0.789	0.906	0.9	0.632	0.661	0.744	0.762
35	0.877	0.795	0.914	0.903	0.638	0.671	0.751	0.77
40	0.89	0.82	0.919	0.91	0.68	0.719	0.782	0.803
45	0.884	0.821	0.894	0.916	0.683	0.715	0.774	0.803
50	0.885	0.845	0.876	0.93	0.704	0.754	0.781	0.833
55	0.899	0.854	0.919	0.933	0.713	0.769	0.803	0.843
60	0.891	0.864	0.881	0.937	0.723	0.788	0.794	0.856
65	0.885	0.851	0.876	0.935	0.701	0.762	0.779	0.839
70	0.884	0.846	0.879	0.931	0.697	0.754	0.777	0.833
75	0.898	0.858	0.903	0.927	0.725	0.785	0.804	0.85
100	0.895	0.877	0.874	0.932	0.744	0.82	0.804	0.873
150	0.883	0.841	0.874	0.933	0.697	0.743	0.776	0.827
300	0.876	0.831	0.858	0.938	0.689	0.717	0.764	0.812
1524	0.886	0.811	0.955	0.956	0.638	0.66	0.765	0.781

Table E.24 Multi-Modal Profiling Test Results (MLP)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.841	0.724	0.915	0.887	0.501	0.528	0.647	0.662
10	0.856	0.744	0.934	0.927	0.543	0.542	0.687	0.684
15	0.891	0.814	0.923	0.914	0.683	0.702	0.785	0.794
20	0.894	0.831	0.932	0.925	0.685	0.729	0.789	0.815
23	0.911	0.838	0.901	0.905	0.78	0.763	0.836	0.828
25	0.906	0.858	0.929	0.931	0.732	0.781	0.819	0.849
30	0.885	0.793	0.93	0.93	0.653	0.643	0.767	0.761
35	0.893	0.815	0.934	0.932	0.681	0.689	0.788	0.792
40	0.902	0.852	0.955	0.947	0.694	0.752	0.804	0.838
45	0.881	0.79	0.956	0.954	0.619	0.62	0.752	0.751
50	0.901	0.848	0.953	0.959	0.692	0.734	0.802	0.831
55	0.911	0.878	0.946	0.955	0.737	0.799	0.828	0.87
60	0.911	0.872	0.944	0.951	0.738	0.791	0.829	0.863
65	0.919	0.887	0.955	0.958	0.756	0.814	0.844	0.88
70	0.867	0.767	0.947	0.952	0.573	0.573	0.714	0.715
75	0.898	0.826	0.941	0.951	0.692	0.696	0.798	0.804
100	0.912	0.871	0.945	0.958	0.739	0.782	0.829	0.861
150	0.916	0.879	0.951	0.959	0.748	0.797	0.837	0.871
300	0.899	0.831	0.962	0.966	0.679	0.695	0.796	0.808
1524	0.911	0.864	0.963	0.972	0.721	0.756	0.825	0.851

Table E.25 Multi-Modal Profiling Test Results (Naive Bayes)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.433	0.561	0.331	0.539	0.934	0.98	0.489	0.696
10	0.482	0.584	0.352	0.554	0.927	0.967	0.51	0.704
15	0.834	0.821	0.648	0.748	0.942	0.979	0.768	0.848
20	0.841	0.823	0.658	0.752	0.94	0.978	0.774	0.85
23	0.915	0.889	0.825	0.841	0.896	0.966	0.859	0.899
25	0.915	0.893	0.824	0.846	0.901	0.967	0.861	0.902
30	0.912	0.887	0.82	0.84	0.893	0.962	0.855	0.897
35	0.912	0.889	0.818	0.842	0.897	0.963	0.856	0.898
40	0.925	0.913	0.84	0.872	0.916	0.972	0.877	0.919
45	0.926	0.914	0.839	0.872	0.922	0.974	0.878	0.92
50	0.927	0.915	0.845	0.877	0.918	0.97	0.88	0.921
55	0.928	0.917	0.858	0.884	0.903	0.965	0.88	0.922
60	0.933	0.919	0.906	0.907	0.859	0.938	0.882	0.922
65	0.93	0.911	0.912	0.91	0.841	0.917	0.875	0.913
70	0.928	0.909	0.915	0.913	0.83	0.909	0.87	0.911
75	0.929	0.908	0.907	0.905	0.842	0.918	0.873	0.911
100	0.932	0.912	0.892	0.894	0.87	0.94	0.881	0.916
150	0.909	0.865	0.868	0.869	0.809	0.868	0.837	0.869
300	0.886	0.811	0.856	0.86	0.73	0.753	0.788	0.803
1524	0.823	0.761	0.697	0.813	0.693	0.692	0.695	0.748

F
DEEP LEARNING EXPERIMENTS

Table F.1 Experimental setup for tuning hyperparameters and network structure of DNN

Experiment	# of Hidden Layers	Layer Formation	Learning Rate	Epoch	Batch Size	Dropout Rate	Input Dropout
Base Exp	2	2x -2x	0.01	100	128	0	N/A
Exp-1	2	2x-2x	0.005	100	128	0	N/A
Exp-2	2	2x-2x	0.001	100	128	0	N/A
Exp-3	2	2x-2x	0.0005	100	128	0	N/A
Exp-4	2	2x-2x	0.0001	100	128	0	N/A
Exp-5	2	2x-2x	0.00005	100	128	0	N/A
Exp-6	2	2x-2x	0.00001	100	128	0	N/A
Exp-7	2	2x-2x	0.00005	250	128	0	N/A
Exp-8	2	2x-2x	0.00001	250	128	0	N/A
Exp-9	2	2x-2x	0.00001	500	128	0	N/A
Exp-10	2	2x-2x	0.00001	500	128	0	N/A
Exp-11	2	2x-2x	0.01	100	512	0	N/A
Exp-12	2	2x-2x	0.01	100	256	0	N/A
Exp-13	2	2x-2x	0.01	100	64	0	N/A
Exp-14	2	2x-2x	0.01	100	32	0	N/A
Exp-15	2	2x-2x	0.01	50	128	0	N/A
Exp-16	2	2x-2x	0.01	250	128	0	N/A
Exp-17	2	2x-2x	0.01	500	128	0	N/A
Exp-18	2	2x-x	0.01	100	128	0	N/A
Exp-19	2	2x-4x	0.01	100	128	0	N/A
Exp-20	2	x-2x	0.01	100	128	0	N/A
Exp-21	2	x	0.01	100	128	0	N/A
Exp-22	2	2x	0.01	100	128	0	N/A
Exp-23	3	2x-2x-2x	0.01	100	128	0	N/A
Exp-24	3	2x-4x-8x	0.01	100	128	0	N/A
Exp-25	3	x-x-x	0.01	100	128	0	N/A
Exp-26	3	x-2x-4x	0.01	100	128	0	N/A
Exp-27	3	2x-2x-2x	0.01	100	128	0.2	TRUE
Exp-28	3	2x-4x-8x	0.01	100	128	0.2	TRUE
Exp-29	3	2x-x-2x	0.01	100	128	0.2	TRUE
Exp-30	2	2x-2x	0.00001	500	128	0.2	TRUE
Exp-31	3	2x-2x-2x	0.00001	500	128	0.2	TRUE
Exp-32	3	2x-4x-8x	0.00001	500	128	0.2	TRUE
Exp-33	3	2x-4x-2x	0.00001	500	128	0.2	TRUE
Exp-34	3	x-2x-4x	0.00001	500	128	0.2	TRUE
Exp-35	3	x-2x-x	0.00001	500	128	0.2	TRUE

Table F.1 *Experimental setup for tuning hyperparameters and network structure of DNN Cont.*

Experiment	# of Hidden Layers	Layer Formation	Learning Rate	Epoch	Batch Size	Dropout Rate	Input Dropout
Exp-36	3	x-x-x	0.00001	500	128	0.2	TRUE
Exp-37	2	2x-2x	0.00001	500	128	0.2	FALSE
Exp-38	3	2x-2x-2x	0.00001	500	128	0.2	FALSE
Exp-39	3	2x-4x-8x	0.00001	500	128	0.2	FALSE
Exp-40	3	2x-4x-2x	0.00001	500	128	0.2	FALSE
Exp-41	3	x-2x-4x	0.00001	500	128	0.2	FALSE
Exp-42	3	x-2x-x	0.00001	500	128	0.2	FALSE
Exp-43	3	x-x-x	0.00001	500	128	0.2	FALSE
Exp-44	3	2x-2x-2x	0.00002	500	128	0.3	FALSE
Exp-45	**3**	**x-2x-4x**	**0.00002**	**500**	**128**	**0.3**	**FALSE**
Exp-46	3	x-x-x	0.00002	500	128	0.3	FALSE

Table F.2 The experimental results of fine-tuning test setup for DNN

Experiment	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
Base Exp	0.895	0.82	0.934	0.943	0.687	0.69	0.792	0.797
Exp-1	0.886	0.8	0.931	0.937	0.656	0.654	0.77	0.77
Exp-2	0.904	0.839	0.94	0.938	0.713	0.734	0.811	0.824
Exp-3	0.901	0.836	0.937	0.934	0.707	0.73	0.805	0.82
Exp-4	0.918	0.868	0.93	0.931	0.775	0.802	0.846	0.862
Exp-5	0.918	0.872	0.947	0.942	0.761	0.799	0.844	0.865
Exp-6	0.919	0.89	0.947	0.946	0.764	0.832	0.846	0.885
Exp-7	0.911	0.854	0.93	0.931	0.75	0.772	0.83	0.844
Exp-8	0.923	0.894	0.942	0.941	0.783	0.846	0.855	0.891
Exp-9	0.879	0.785	0.955	0.953	0.614	0.61	0.747	0.744
Exp-10	0.925	0.899	0.937	0.938	0.796	0.86	0.861	0.897
Exp-11	0.891	0.808	0.938	0.942	0.668	0.665	0.78	0.78
Exp-12	0.887	0.809	0.932	0.932	0.657	0.676	0.771	0.783
Exp-13	0.888	0.819	0.936	0.939	0.66	0.691	0.774	0.796
Exp-14	0.901	0.835	0.922	0.926	0.722	0.735	0.809	0.82
Exp-15	0.905	0.84	0.919	0.93	0.74	0.745	0.82	0.827
Exp-16	0.896	0.829	0.95	0.95	0.679	0.703	0.792	0.808
Exp-17	0.904	0.841	0.924	0.944	0.731	0.732	0.816	0.824
Exp-18	0.889	0.821	0.939	0.94	0.661	0.696	0.776	0.8
Exp-19	0.89	0.803	0.934	0.935	0.668	0.661	0.779	0.775
Exp-20	0.902	0.843	0.926	0.933	0.718	0.747	0.809	0.83
Exp-21	0.888	0.814	0.936	0.937	0.658	0.682	0.773	0.789
Exp-22	0.881	0.797	0.944	0.945	0.627	0.64	0.754	0.763
Exp-23	0.891	0.827	0.933	0.943	0.675	0.706	0.783	0.807
Exp-24	0.908	0.851	0.929	0.933	0.74	0.763	0.824	0.84
Exp-25	0.885	0.798	0.929	0.938	0.654	0.649	0.767	0.767
Exp-26	0.904	0.842	0.928	0.93	0.724	0.748	0.814	0.829
Exp-27	0.927	0.904	0.947	0.94	0.794	0.867	0.864	0.902
Exp-28	0.911	0.856	0.947	0.942	0.732	0.765	0.826	0.844
Exp-29	0.913	0.869	0.947	0.941	0.742	0.794	0.832	0.861
Exp-30	0.923	0.897	0.948	0.943	0.779	0.85	0.855	0.895
Exp-31	0.926	0.9	0.946	0.944	0.792	0.855	0.862	0.898
Exp-32	0.921	0.892	0.946	0.944	0.772	0.839	0.85	0.888
Exp-33	0.926	0.898	0.951	0.945	0.786	0.851	0.861	0.895
Exp-34	0.924	0.896	0.946	0.944	0.782	0.847	0.856	0.893
Exp-35	0.923	0.894	0.947	0.946	0.78	0.842	0.855	0.891
Exp-36	0.928	0.9	0.943	0.938	0.8	0.862	0.866	0.899
Exp-37	0.923	0.892	0.938	0.937	0.786	0.845	0.855	0.889
Exp-38	0.927	0.905	0.938	0.939	0.803	0.87	0.865	0.903
Exp-39	0.935	0.895	0.936	0.935	0.795	0.853	0.86	0.893
Exp-40	0.904	0.848	0.934	0.938	0.72	0.754	0.813	0.836
Exp-41	0.931	0.91	0.935	0.937	0.82	0.883	0.874	0.909
Exp-42	0.925	0.894	0.935	0.933	0.796	0.854	0.86	0.892
Exp-43	0.931	0.908	0.935	0.936	0.818	0.88	0.873	0.907
Exp-44	0.933	0.922	0.951	0.952	0.812	0.881	0.876	0.915
Exp-45	**0.942**	**0.922**	**0.957**	**0.962**	**0.837**	**0.904**	**0.893**	**0.932**
Exp-46	0.938	0.918	0.951	0.955	0.831	0.896	0.887	0.924

G
ENSEMBLE LEARNING EXPERIMENTS

Table G.1 Ensemble Learning Results (Random Forest, SVM, Naive Bayes)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.771	0.646	0.751	0.805	0.316	0.406	0.445	0.54
10	0.876	0.789	0.878	0.886	0.665	0.674	0.757	0.765
15	0.927	0.886	0.897	0.898	0.845	0.877	0.87	0.887
20	0.924	0.882	0.898	0.904	0.831	0.862	0.863	0.882
23	0.923	0.893	0.912	0.904	0.814	0.886	0.86	0.895
25	0.926	0.889	0.921	0.909	0.814	0.87	0.864	0.889
30	0.923	0.886	0.918	0.913	0.805	0.859	0.858	0.885
35	0.922	0.884	0.923	0.914	0.797	0.854	0.855	0.883
40	0.933	0.904	0.927	0.919	0.836	0.89	0.879	0.904
45	0.934	0.907	0.935	0.926	0.829	0.888	0.879	0.907
50	0.933	0.911	0.943	0.938	0.819	0.884	0.876	0.91
55	0.935	0.923	0.946	0.942	0.822	0.905	0.88	0.924
60	0.926	0.909	0.946	0.943	0.791	0.874	0.861	0.907
65	0.929	0.912	0.949	0.945	0.799	0.88	0.867	0.911
70	0.926	0.906	0.949	0.942	0.787	0.87	0.86	0.904
75	0.926	0.908	0.946	0.938	0.789	0.878	0.86	0.907
100	0.935	0.921	0.947	0.94	0.824	0.902	0.881	0.921
150	0.927	0.898	0.947	0.939	0.793	0.856	0.863	0.896
300	0.92	0.877	0.951	0.946	0.763	0.806	0.846	0.871
1524	0.911	0.857	0.963	0.965	0.72	0.747	0.824	0.842

Table G.2 Ensemble Learning Results (Random Forest , MLP , Naive Bayes)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.832	0.722	0.811	0.821	0.548	0.585	0.654	0.683
10	0.878	0.807	0.865	0.873	0.687	0.73	0.766	0.795
15	0.912	0.856	0.895	0.9	0.789	0.808	0.839	0.851
20	0.912	0.86	0.904	0.911	0.781	0.805	0.838	0.854
23	0.93	0.883	0.916	0.91	0.837	0.855	0.874	0.882
25	0.926	0.888	0.931	0.93	0.806	0.844	0.864	0.885
30	0.924	0.876	0.944	0.939	0.786	0.811	0.858	0.87
35	0.921	0.87	0.942	0.936	0.778	0.802	0.852	0.863
40	0.928	0.887	0.947	0.939	0.796	0.832	0.865	0.883
45	0.924	0.888	0.951	0.952	0.78	0.824	0.857	0.883
50	0.926	0.88	0.953	0.954	0.783	0.803	0.86	0.872
55	0.932	0.898	0.954	0.954	0.804	0.841	0.873	0.894
60	0.927	0.89	0.957	0.953	0.785	0.827	0.862	0.885
65	0.933	0.9	0.962	0.958	0.801	0.841	0.874	0.896
70	0.921	0.878	0.962	0.958	0.756	0.796	0.847	0.87
75	0.926	0.894	0.957	0.957	0.78	0.831	0.859	0.889
100	0.931	0.89	0.959	0.961	0.798	0.818	0.871	0.884
150	0.93	0.891	0.957	0.956	0.795	0.824	0.869	0.885
300	0.917	0.858	0.967	0.967	0.739	0.748	0.838	0.844
1524	0.919	0.863	0.968	0.974	0.745	0.752	0.842	0.849

Table G.3 Ensemble Learning Results (Decision Tree , SVM , Naive Bayes)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.759	0.631	0.717	0.813	0.28	0.363	0.402	0.502
10	0.867	0.768	0.865	0.888	0.642	0.626	0.737	0.734
15	0.897	0.837	0.88	0.889	0.748	0.779	0.809	0.831
20	0.901	0.842	0.882	0.889	0.763	0.791	0.818	0.837
23	0.906	0.861	0.906	0.907	0.756	0.811	0.824	0.856
25	0.909	0.86	0.912	0.905	0.761	0.811	0.83	0.855
30	0.901	0.834	0.908	0.899	0.735	0.76	0.813	0.824
35	0.903	0.84	0.914	0.906	0.735	0.767	0.815	0.831
40	0.913	0.863	0.92	0.913	0.768	0.809	0.837	0.858
45	0.913	0.851	0.925	0.913	0.761	0.784	0.835	0.844
50	0.914	0.877	0.924	0.931	0.768	0.82	0.838	0.872
55	0.92	0.89	0.938	0.933	0.778	0.846	0.85	0.887
60	0.92	0.894	0.94	0.934	0.774	0.853	0.849	0.891
65	0.916	0.886	0.942	0.939	0.758	0.831	0.84	0.882
70	0.913	0.866	0.943	0.931	0.745	0.797	0.833	0.859
75	0.915	0.867	0.935	0.927	0.76	0.803	0.838	0.861
100	0.921	0.889	0.93	0.93	0.789	0.847	0.854	0.886
150	0.911	0.861	0.922	0.923	0.756	0.794	0.831	0.854
300	0.888	0.838	0.872	0.936	0.719	0.734	0.788	0.823
1524	0.906	0.856	0.949	0.955	0.715	0.753	0.816	0.842

Table G.4 Ensemble Learning Results (Decision Tree , MLP , Naive Bayes)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.834	0.736	0.802	0.828	0.568	0.612	0.665	0.704
10	0.869	0.8	0.847	0.872	0.671	0.714	0.749	0.785
15	0.899	0.845	0.882	0.896	0.754	0.789	0.813	0.839
20	0.9	0.847	0.886	0.899	0.754	0.789	0.815	0.84
23	0.922	0.876	0.904	0.907	0.818	0.844	0.859	0.874
25	0.918	0.881	0.92	0.925	0.786	0.835	0.848	0.877
30	0.912	0.857	0.932	0.929	0.753	0.78	0.833	0.848
35	0.914	0.861	0.933	0.93	0.759	0.788	0.837	0.853
40	0.916	0.875	0.946	0.942	0.755	0.804	0.84	0.868
45	0.913	0.852	0.945	0.943	0.743	0.756	0.832	0.839
50	0.92	0.879	0.946	0.952	0.768	0.804	0.848	0.872
55	0.926	0.892	0.945	0.943	0.793	0.839	0.862	0.888
60	0.922	0.882	0.949	0.943	0.774	0.819	0.853	0.877
65	0.926	0.89	0.955	0.953	0.781	0.827	0.859	0.885
70	0.903	0.837	0.952	0.944	0.702	0.724	0.808	0.82
75	0.92	0.866	0.954	0.947	0.763	0.782	0.848	0.857
100	0.926	0.885	0.949	0.945	0.788	0.823	0.861	0.88
150	0.922	0.886	0.948	0.948	0.776	0.821	0.853	0.88
300	0.903	0.837	0.952	0.958	0.702	0.713	0.808	0.817
1524	0.922	0.893	0.959	0.965	0.764	0.82	0.851	0.887

Table G.5 Ensemble Learning Results (SVM , MLP , Naive Bayes)

Feature Count	Accuracy	C-Accuracy	Precision	C-Precision	Recall	C-Recall	F1 Score	C-F1 Score
5	0.848	0.746	0.919	0.894	0.523	0.571	0.667	0.697
10	0.869	0.776	0.942	0.935	0.586	0.605	0.723	0.734
15	0.914	0.867	0.911	0.903	0.781	0.829	0.841	0.864
20	0.915	0.868	0.915	0.904	0.781	0.831	0.843	0.866
23	0.92	0.877	0.901	0.891	0.814	0.866	0.855	0.878
25	0.92	0.888	0.917	0.904	0.797	0.873	0.853	0.888
30	0.911	0.865	0.914	0.907	0.765	0.821	0.833	0.862
35	0.911	0.863	0.918	0.906	0.763	0.818	0.833	0.86
40	0.922	0.889	0.927	0.916	0.795	0.864	0.856	0.889
45	0.916	0.862	0.931	0.919	0.767	0.801	0.841	0.856
50	0.932	0.914	0.941	0.939	0.816	0.891	0.874	0.914
55	0.931	0.92	0.943	0.94	0.812	0.902	0.873	0.921
60	0.928	0.915	0.944	0.94	0.798	0.891	0.865	0.915
65	0.924	0.91	0.947	0.944	0.784	0.876	0.858	0.908
70	0.912	0.876	0.946	0.94	0.74	0.81	0.831	0.87
75	0.923	0.894	0.946	0.935	0.779	0.852	0.855	0.892
100	0.93	0.918	0.94	0.936	0.81	0.901	0.87	0.918
150	0.921	0.896	0.94	0.935	0.777	0.856	0.851	0.894
300	0.912	0.867	0.948	0.944	0.737	0.787	0.829	0.858
1524	0.907	0.861	0.954	0.962	0.716	0.758	0.818	0.848

Table G.6 F1 Score Comparison of Machine Learning and Ensemble Learning (Quantity Based)

Feature Count	DT	RF	SVM	MLP	NB	RF SVM NB	RF MLP NB	DT SVM NB	DT MLP NB	SVM MLP NB
5	0.367	0.411	0.121	0.647	0.489	0.445	0.654	0.402	0.665	0.667
10	0.723	0.765	0.53	0.687	0.51	0.757	0.766	0.737	0.749	0.723
15	0.74	0.834	0.758	0.785	0.768	0.87	0.839	0.809	0.813	0.841
20	0.744	0.836	0.757	0.789	0.774	0.863	0.838	0.818	0.815	0.843
23	0.704	0.855	0.76	0.836	0.859	0.86	0.874	0.824	0.859	0.855
25	0.734	0.86	0.768	0.819	0.861	0.864	0.864	0.83	0.848	0.853
30	0.711	0.858	0.744	0.767	0.855	0.858	0.858	0.813	0.833	0.833
35	0.714	0.827	0.751	0.788	0.856	0.855	0.852	0.815	0.837	0.833
40	0.685	0.866	0.782	0.804	0.877	0.879	0.865	0.837	0.84	0.856
45	0.745	0.849	0.774	0.752	0.878	0.879	0.857	0.835	0.832	0.841
50	0.727	0.851	0.781	0.802	0.88	0.876	0.86	0.838	0.848	0.874
55	0.745	0.857	0.803	0.828	0.88	0.88	0.873	0.85	0.862	0.873
60	0.745	0.825	0.794	0.829	0.882	0.861	0.862	0.849	0.853	0.865
65	0.732	0.855	0.779	0.844	0.875	0.867	0.874	0.84	0.859	0.858
70	0.752	0.863	0.777	0.714	0.87	0.86	0.847	0.833	0.808	0.831
75	0.76	0.837	0.804	0.798	0.873	0.86	0.859	0.838	0.848	0.855
100	0.731	0.846	0.804	0.829	0.881	0.881	0.871	0.854	0.861	0.87
150	0.754	0.851	0.776	0.837	0.837	0.863	0.869	0.831	0.853	0.851
300	0.69	0.839	0.764	0.796	0.788	0.846	0.838	0.788	0.808	0.829
1525	0.76	0.797	0.765	0.825	0.695	0.824	0.842	0.816	0.851	0.818
Max F1	0.76	0.866	0.804	0.844	0.882	0.881	0.874	0.854	0.862	0.874

Table G.7 F1 Score Comparison of Machine Learning and Ensemble Learning (Cost Based)

Feature Count	DT	RF	SVM	MLP	NB	RF SVM NB	RF MLP NB	DT SVM NB	DT MLP NB	SVM MLP NB
5	0.41	0.455	0.171	0.662	0.696	0.54	0.683	0.502	0.704	0.697
10	0.731	0.766	0.516	0.684	0.704	0.765	0.795	0.734	0.785	0.734
15	0.713	0.802	0.775	0.794	0.848	0.887	0.851	0.831	0.839	0.864
20	0.713	0.831	0.774	0.815	0.85	0.882	0.854	0.837	0.84	0.866
23	0.712	0.848	0.788	0.828	0.899	0.895	0.882	0.856	0.874	0.878
25	0.741	0.853	0.796	0.849	0.902	0.889	0.885	0.855	0.877	0.888
30	0.715	0.848	0.762	0.761	0.897	0.885	0.87	0.824	0.848	0.862
35	0.712	0.812	0.77	0.792	0.898	0.883	0.863	0.831	0.853	0.86
40	0.661	0.859	0.803	0.838	0.919	0.904	0.883	0.858	0.868	0.889
45	0.724	0.852	0.803	0.751	0.92	0.907	0.883	0.844	0.839	0.856
50	0.737	0.843	0.833	0.831	0.921	0.91	0.872	0.872	0.872	0.914
55	0.745	0.858	0.843	0.87	0.922	0.924	0.894	0.887	0.888	0.921
60	0.75	0.825	0.856	0.863	0.922	0.907	0.885	0.891	0.877	0.915
65	0.723	0.85	0.839	0.88	0.913	0.911	0.896	0.882	0.885	0.908
70	0.729	0.863	0.833	0.715	0.911	0.904	0.87	0.859	0.82	0.87
75	0.734	0.858	0.85	0.804	0.911	0.907	0.889	0.861	0.857	0.892
100	0.739	0.845	0.873	0.861	0.916	0.921	0.884	0.886	0.88	0.918
150	0.762	0.848	0.827	0.871	0.869	0.896	0.885	0.854	0.88	0.894
300	0.666	0.842	0.812	0.808	0.803	0.871	0.844	0.823	0.817	0.858
1525	0.834	0.808	0.781	0.851	0.748	0.842	0.849	0.842	0.887	0.848
Max F1	0.834	0.863	0.873	0.88	0.922	0.924	0.896	0.891	0.888	0.921

PUBLICATIONS FROM THE THESIS

Papers

1. M. T. Aras, M. A. Guvensan, "A multi-modal profiling fraud-detection system for capturing suspicious airline ticket activities," Applied Sciences, vol. 13, no. 24, p. 13 121, 2023.

Conference Papers

1. M. T. Aras, M. A. Güvensan, "Challenges and key points for fraud detection in aviation," in 2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), IEEE, 2021, pp. 1–6.