جامعة سيدي محمد بن عبد الله بفاس
UNIVERSITÉ SIDI MOHAMED BEN ABDELLAH DE FES

كلية العلوم ظهر المهراز فاس
FACULTE DES SCIENCES DHAR EL MAHRAZ FES

N° d'ordre : ……..

Centre des Etudes : ***Sciences et Techniques et Sciences Médicales***

Formation Doctorale : Sciences et Technologies de l'Information et de la Communication
Discipline : Informatique
Spécialité : Informatique
Laboratoire : d'Informatique, Signaux, Automatique et Cognitivisme

THÈSE DE DOCTORAT

Titre de la thèse

Fraud Detection using Deep Learning Architectures

Présentée par :

Mme EL HLOULI Fatima Zohra

Soutenue le 24 Février 2024 devant le jury :

Nom et prénom	**Etablissement**	**Grade**	**Qualité**
Tairi Hamid	FSDM	PES	Président
Aghoutane Badraddine	Faculté des Sciences de Meknès	PES	Rapporteur & examinateur
Khamjane Aziz	École Nationale des Sciences Appliquées d'Al Hoceima	PH	Rapporteur & examinateur
Nfaoui El habib	FSDM	PES	Rapporteur & examinateur
El Fazazy Khalid	FSDM	PH	Examinateur
Mahraz Mohamed Adnane	FSDM	PH	Examinateur
Yahyaouy Ali	FSDM	PES	Examinateur
Riffi Jamal	FSDM	PH	Encadrant

To my parents, Mohamed and Zakia,

To my husband Mohamed,

To my sons Safouane, Mehdi and Lina

To the soul of my Aunt Aicha,

To my brother Taha and my sister Soukaina,

To all those to whom I owe what I have become.

Contents

Contents 1
Acknowledgments 4
Abstract 5
Résumé 6
List of Tables 7
List of Figures 8
Abbreviations 10
General Introduction 11
1. Research Hypothesis 13
2. Technical Goals 13
3. Contributions 14
4. Thesis structure 16
Chapter I: State of The Art 17
1. Introduction 17
2. Types of financial fraud 17
2.1. Mortgage Fraud 18
2.2. Money Laundering 18
2.3. Financial Statement Fraud 19
2.4. Insurance Fraud 19
2.5. Cryptocurrency Fraud 20
2.6. Credit Card Fraud 20
3. Credit card fraud detection techniques 22
3.1. Data Mining Techniques 22
3.1.1. Random Forest 23
3.1.2. Decision Tree 24
3.1.3. Support Vector Machine 25
3.1.4. Naïve Bayes 28
3.1.5. K-Nearest Neighbors 30
3.1.6. Logistic Regression 31
3.1.7. Hidden Markov model (HMM) 32
3.1.8. Fuzzy Logic-based system Fuzzy 32
3.1.9. Extreme Learning Machine 34
3.1.10. Artificial Neural Network 35
3.1.11. CNN 36
3.1.12. LSTM 38

3.1.13. Autoencoder 39
3.2. Metaheuristic algorithms 41
3.2.1. Genetic algorithm (GA) 41
3.2.2. Practicle Swarm Optimization PSO 42
3.2.3. Bat Algorithm 43
3.2.4. Dandelion Algorithm 43
3.2.5. Group Search Optimization 44
4. Challenges in Fraud Detection 44
5. Practical implications 46
5.1. Reducing financial losses 46
5.2. Enhanced transaction security 47
5.3. Operational optimization 47
5.4. Protecting personal data 48
5.5. Adaptability to new threats 49
6. Conclusion 49
Chapter II: Imbalance Resolution for Credit Card Fraud Detection 51
1. Introduction 51
2. Traditional Imbalanced Data Techniques 53
2.1. Oversampling 53
2.2. Synthetic minority oversampling technique (Smote) 53
2.3. Synthetic sampling (AdaSyn) 54
2.4. Random oversampling (ROS) 54
2.5. Undersampling 55
2.6. Random undersampling (RUS) 55
3. Adopting Architectural and Technical models 55
3.1. Model 1: SMOTE-enhanced Fraud Detection Model 55
3.2. Model 2: Autoencoder-based Fraud Detection Model 57
3.3. Model 3: Multi-phase Preprocessed Fraud Detection Model 58
4. Preprocessing Data Models for Enhanced Predictive Analytics 61
4.1. Datasets and Features 61
4.2. Synthetic Minority Over-Sampling Technique (SMOTE) approach 61
4.3. Autoencoder-based Approach 63
4.4. Multi-phase Preprocessing approach 63
5. Comparative Analysis and Effectiveness 64
5.1.Evaluation metrics 64
5.2. Results found 65
5.2.1. SMOTE-enhanced Model 65

5.2.2. Autoencoder-based Model 67
5.2.3. Multi-phase Preprocessed Model 69
5.3. Comparative Analysis 73
6. Conclusion 75
Chapter III: Optimization Model and Performance Improvement 79
1. Introduction 79
2. Optimization Techniques for Deep Learning Architectures in Fraud Detection 80
3. Adaptation of Dandelion Optimization Algorithm to ELM Architecture 82
3.1. Dandelion Optimizer Algorithm 82
3.2. Kernel Extreme Learning Machine 86
3.3. Multilayer Extreme Learning Machines 87
3.4. S-AE-KELM-DA 88
4. Evaluation of Optimized ELM Model Performance 91
4.1.Datasets 91
4.2. Experiment results 92
4.3. Comparative results 95
5. Conclusion 98
Chapter IV: Integrating Data Imbalance Resolution and Model Optimization for Credit Card Fraud Detection 99
1. Introduction 99
2. Related Works 101
2.1. Reptile Search Algorithm 101
2.2. Other optimization algorithms 102
2.3. Weighted Extreme Learning Machine 103
3. Optimizing WB-ELM by RSA 105
4. Experiments and Comparative Results 107
General Conclusion and Perspectives 115
1. Limitations and ways to improve 117
2. Future developments 118
3. Emerging technology 119
4. Cooperation and collaboration 119
5. Conclusion 120
Publication list 121
References 121

Acknowledgments

I want to express my heartfelt appreciation and gratitude to the following individuals for their invaluable support, guidance, and encouragement throughout my research and the completion of this thesis.

First and foremost, I am deeply indebted to my thesis director, Dr. Jamal Riffi, for his unwavering dedication, expertise, and mentorship. His insightful feedback, constructive criticism, and commitment to my academic and research pursuits have been instrumental in shaping this work. I am truly fortunate to have had the opportunity to learn from him and benefit from his profound knowledge of deep learning.

I sincerely thank Dr. Hamid Tairi, the laboratory director, for providing the necessary resources, infrastructure, and a conducive research environment. I also thank him for agreeing to honor me by presiding over the jury for my thesis defense.

My sincere thanks to Mr. Aghoutane Badraddine, Professor at the Faculty of Sciences, Meknes, Mr. Khamjane Aziz, Professor at the National School of Applied Sciences of Al Hoceima, and Mr. Nfaoui El Habib, Professor at the Faculty of Sciences, Dhar Mehraz, Fes, for taking the time to evaluate this work and especially for honoring me by agreeing to be reviewers of this thesis.

I would also like to thank Mr. El Fazazy Khalid, Mr. Mahraz Mohamed Adnane, and Mr. Yahyaouy Ali, professors at the Faculty of Sciences Dhar El Mahraz, Fes, for agreeing to examine this work and be part of this jury.

I am grateful to my colleagues and fellow researchers at the laboratory for their collaboration, fruitful discussions, and shared experiences that enriched my research path.

Furthermore, I would like to express my gratitude for the assistance provided by my friends and family, whose continual encouragement and comprehension provided a permanent motivation.

Finally, I express my gratitude to all the researchers whose work and contributions have been a source of inspiration and knowledge for this research.

This research would not have been possible without the support and encouragement of these individuals.

Thank you for being part of this path...

Abstract

The coronavirus pandemic has led to significant growth in online banking transactions. This development is explained by the travel restrictions and social distancing measures imposed during the pandemic, which pushed many people to favor online payments and remote transactions for their daily needs. The increasing focus on fraud detection and prevention, particularly in the area of credit card issuers, stems from the significant annual financial losses due to fraudulent use of cards. Reducing fraud can result in significant operational cost savings, prompting extensive research in this area.

In fraud detection, the goal is to obtain a high-performance model that accurately identifies fraudulent transactions while minimizing false positives. However, this challenge also involves considering the complexity of the model and the time required for detection. A successful model must find a balance between these three aspects. A model that is too complex can be difficult to interpret, while a model that is too simple can compromise accuracy. Additionally, the time it takes to detect fraud is crucial, especially in situations requiring rapid responses. Thus, the challenge consists of finding an effective model that is sufficiently precise and fast, adapted to the specific needs of fraud detection, while taking into account resource constraints and responsiveness requirements. Artificial neural networks have shown promise in fraud detection. Deep learning methods provide advanced capabilities for analyzing unstructured or sequential data, discovering hidden patterns, and detecting sophisticated fraud schemes.

As for optimization methods such as Dandelion, Particle Swarm, and the Group Search algorithm, they are often used to efficiently search for solutions in complex search spaces, to find optimal parameters in machine learning models, or to tune the hyperparameters of the algorithms. Using these methods, our approach appears to include a combination of learning architectures and techniques applied to unbalanced and balanced datasets, common in fraud detection scenarios. To establish a framework for comparison, we researched feature selection, preprocessing, and different performance metrics. Our experiments used real credit card transaction data and employed distinct machine and deep learning techniques such as Fully Connected Neural Network (FCNN), as well as two innovative methods based on: Extreme Learning Machine (ELM) and Autoencoder (AE).

Keywords: Fraud, Credit cards, Data preprocessing, Imbalanced data, Supervised learning, Fully Connected Neural networks, Extreme learning machine, Autoencoder, Majority voting, Dandelion algorithm.

Résumé

La pandémie de coronavirus a engendré une croissance significative des transactions bancaires effectuées en ligne. Cette évolution s'explique par les restrictions de déplacement et les mesures de distanciation sociale imposées pendant la pandémie, poussant de nombreuses personnes à privilégier les paiements en ligne et les transactions à distance pour leurs besoins quotidiens. La concentration accrue sur la détection et la prévention de la fraude, notamment dans le domaine des émetteurs de cartes de crédit, découle des importantes pertes financières annuelles dues à une utilisation frauduleuse des cartes. La réduction de la fraude peut engendrer d'importantes économies sur les coûts opérationnels, ce qui suscite des recherches approfondies dans ce domaine.

Dans la détection de fraudes, l'objectif est d'obtenir un modèle performant qui identifie avec précision les transactions frauduleuses tout en minimisant les faux positifs. Cependant, ce défi implique également de considérer la complexité du modèle et le temps nécessaire à la détection. Un modèle performant doit trouver un équilibre entre ces trois aspects. Un modèle trop complexe peut être difficile à interpréter, tandis qu'un modèle trop simple peut compromettre la précision. De plus, le temps nécessaire pour détecter une fraude est crucial, surtout dans des situations nécessitant des réponses rapides. Ainsi, le défi consiste à trouver un modèle efficace, suffisamment précis et rapide, adapté aux besoins spécifiques de détection de fraude, tout en tenant compte des contraintes de ressources et des impératifs de réactivité.

Les réseaux de neurones artificiels se sont révélés prometteurs pour la détection des fraudes. Ils fournissent une base solide pour créer des modèles prédictifs basés sur des données structurées, tandis que les méthodes d'apprentissage profond offrent des capacités avancées pour analyser des données non structurées ou séquentielles, découvrir des modèles cachés et détecter des stratagèmes de fraude sophistiqués.

Quant aux méthodes d'optimisation telles que le Dandelion, Particle swarm et le Group search algorithme, elles sont souvent utilisées pour rechercher efficacement des solutions dans des espaces de recherche complexes, pour trouver les paramètres optimaux dans les modèles d'apprentissage automatique ou pour ajuster les hyperparametres des algorithmes.

En utilisant ces méthodes, notre approche semble inclure une combinaison d'architectures et techniques d'apprentissage appliquées à des ensembles de données déséquilibrés et équilibrés, fréquents dans les scénarios de détection de fraudes. Pour établir un cadre de comparaison, nous avons mené des recherches sur la sélection des caractéristiques, le prétraitement et les différentes mesures de performance.

Nos expériences ont utilisé des données réelles de transactions par carte de crédit et ont employé des techniques distinctes d'apprentissage automatique et profond tel que le réseau de neurones entièrement connecté (FCNN), ainsi que deux méthodes innovantes basé sur : l'Extreme Learning Machine (ELM) et l'Autoencodeur (AE).

Mots-clés : Fraude, Cartes de crédit, Prétraitement de données, Données déséquilibrées, Apprentissage supervisé, Réseaux neuronaux entièrement connecté, Extreme learning machine, Autoencoder, Majority voting, Dandelion algorithm.

List of Tables

Table.1 Information about the datasets

Table.2 Accuracy rates for ELM with different activation functions.

Table.3 A confusion matrix for CCFD using ELM

Table.4 A confusion matrix for CCFD using MLP

Table.5 Optimized parameters.

Table.6 Computing Results on the Dataset

Table.7 Results of the proposed models compared to previous studies

Table.8 Information about the datasets

Table.9 Computing results of the dataset

Table.10 Time cost (seconds) of testing different algorithms

Table.11 Parameter settings

Table.12 Comparing results with previous work

Table.13 Datasets information

Table.14 Computing Results of Dataset

Table.15 Time cost (seconds) for different optimization algorithms during the testing phase

Table.16 Parameter settings

Table.17 Comparing Results with Previous Work

List of Figures

Figure1. Common types of financial fraud
Figure2. Taxonomy of fraud detection techniques
Figure3. Binary Decision Tree
Figure4. Separation between Support Vector and Hyperplane
Figure5. Artificial neural network architecture
Figure6. CNN Architecture
Figure7. LSTM cell structure
Figure8. The architecture of AE with a single encoding and decoding layer
Figure9. Optimization algorithm used for credit card fraud detection
Figure10. Percentage of fraudulent transactions before and after oversampling the original dataset
Figure11. Cumulative Feature Importance
Figure12. Overall steps for detecting Fraudulent transactions from an imbalanced dataset
Figure13. Overall steps for detecting Fraudulent transactions from the imbalanced dataset
Figure14. The training time of MLP and ELM
Figure15. The predicting time of MLP and ELM
Figure16. Training and validation accuracy of the proposed model on CCFD
Figure17. Loss Function of training and validation proposed model on CCFD
Figure18. Training and validation accuracy of the proposed model on the German dataset.
Figure19. Loss function of training and validation proposed model on German dataset
Figure20. Confusion matrix of CCFD
Figure21. Confusion matrix of a German dataset
Figure22. Evaluation of AE Loss Function vs. Epochs
Figure23. Performance Evaluation of Recall using the Proposed Model on CCFD
Figure24. Performance Evaluation of Precision using the Proposed Model on CCFD
Figure25. Performance evaluation of recall using the Proposed Model on Synthetic financial dataset
Figure26. Performance evaluation of precision using the Proposed Model on Synthetic financial dataset
Figure27. Stacked AE-KELM
Figure28. Flow chart of optimized Stacked AE-KELM with DA
Figure29. Training ELM accuracy rates with on datasets various activation functions

Figure30. Performance of KELM, S-AEKELM, S-AEKELM-PSO, S-AEKELM-BAT, and S-AEKELM-DA on fourth datasets (F1-Score)
Figure31. Training the accuracy of S-AEKELM-DA in proportion to the number of AE-KELM
Figure32. Flow chart of optimizing WB-ELM by RSA
Figure33. Convergence of the G-mean for multiple choice of C and σ for Gaussian kernel on five benchmark datasets

Abbreviations

AdaB	AdaBoosted
AI	Artificial Intelligence
ANN	Artificial Neural Network
CART	Classification And Regression Tree
CCFD	Credit card fraud dataset
CNN	Convolutional Neural Network
DA	Dandelion Algorithm
ELM	Extreme Learning Machine
FL	Fuzzy Logic
FCDN	Fully Connected Deep Network
GA	Genetic Algorithm
GMDH	Group Method of Data Handling
GSO	Group Search Optimization
HMM	Hidden Markov Machine
KELM	Kernel Extreme Learning Machine
KNN	K-Nearest Neighbors
LSTM	Long Short-Term Memory
ML	Machine Learning
MLFF	Multilayer Feed Forward Neural Network
MLP	Multilayer Perceptron
PNN	Probabilistic Neural Network
PSO	Practical Swarm Optimization
RF	Random Forest
RL	Regression linear
RNN	Recurrent Neural Network
RSA	Reptile Search Algorithm
RT	Random Tree
SVM	Support Vector Machine
S-AEKELM-DA	Stacked Autoencoder Kernel Extreme Learning Machine
SMOTE	Synthetic Minority Oversampling Technique
TWRF	Trees Weighting Random Forest

General Introduction

In today's digital age, where financial transactions increasingly rely on credit cards, the specter of credit card fraud emerges as an enduring and constantly evolving menace.

Fraud in the payments sector refers to the illicit utilization of credit card information without a real cardholder's knowledge. A fraudulent transaction typically arises from the unauthorized use of a stolen credit card or card number.

Credit card fraud presents a formidable challenge that impacts not only individuals but also financial institutions and the broader economy, extending its consequences far beyond immediate financial losses. The imperative for effective credit card fraud detection mechanisms has reached an unprecedented level of significance. As contemporary technology, including the Internet, mobile devices such as phones and laptops, and social media, continues its rapid expansion and evolution, it has led to a corresponding increase in fraudulent activities (Boutaher et al., 2020). This increase has translated into billions of dollars in losses for businesses, encouraging considerable efforts to explore techniques for fraud detection. This transformation of our technological framework has reshaped the dynamics of financial security, prompting a reassessment of how we protect financial transactions from malicious activities that exploit vulnerabilities in the digital world.

According to the Federal Trade Commission (FTC), credit card fraud was one of the most prevalent types of fraud in the United States in 2021, with 390,000 cases reported. However, that number only scratches the surface of the issue. The payments industry auditor Nilson Report published a prediction in December 2022 that states that over the following ten years, card fraud losses in the United States will rise to $165.1 billion, affecting all age groups and all states. This includes transactions conducted over the phone, online, and by mail order and is predicted to have cost the United States $5.72 billion in losses in 2022, according to Insider Intelligence.

The widespread use of credit cards for various online and offline transactions has made them a prime target for fraudulent activities. Fraudsters have devised sophisticated tactics to compromise credit card security, ranging from stolen card information and unauthorized transactions to more advanced techniques such as account takeovers and identity theft. As a result, the magnitude of credit card fraud is significant, resulting in substantial financial losses for both cardholders and credit card companies.

Moreover, credit card fraud can lead to a ripple effect of adverse consequences. Cardholders may experience inconvenience and financial strain as they attempt to resolve fraudulent charges, dispute transactions, and restore their financial well-being. Financial institutions incur significant operational costs in addressing fraud, reimbursing victims, and implementing protective measures. The reputational damage to both cardholders and credit card companies can be long-lasting and erode trust in the financial system.

These technological advancements are not limited to the use of lone scammers. Robbers and criminal groups with sophisticated networks have also invested in advancing and changing their methods (Bhattacharyya et al., 2011). Because fraudsters are constantly changing their strategies to take advantage of the weaknesses in the current preventative measures, it is necessary to continuously develop and enhance these fraud detection techniques, many of which are directed toward the financial sector (West & Bhattacharya, 2016).

Considering these challenges, the development of robust credit card fraud detection systems is paramount. Traditional rule-based approaches have demonstrated limitations in keeping up with the evolving strategies of fraudsters. Therefore, implementing advanced technologies—in particular, deep learning has become an appropriate means to improve the precision and effectiveness of credit card fraud detection.

In essence, credit card fraud detection plays a pivotal role in safeguarding financial transactions and the financial stability of individuals and institutions. The purpose of this study is to support further initiatives to lessen the effects of credit card theft and strengthen public confidence in the safety and security of electronic payments. In a cloud setting, artificial intelligence is essential for controlling financial risk. Numerous studies have looked into ways to detect online transaction fraud; however, the techniques now in use are unable to carry out detection with a high degree of precision. The first line of defense against fraudulent practices that may damage financial institutions, companies, and individuals is fraud prevention techniques. These methods cover a wide range of approaches, tools, and tactics intended to detect, decrease, and end fraudulent activity. Adopting strong fraud protection measures has become essential to protecting assets, data, and customer trust in an era where the financial and digital sectors are growing increasingly interconnected. This thesis focuses on the application of deep learning techniques to the realm of credit card fraud detection to address the multifaceted challenges associated with this issue.

1. Research Hypothesis

Our research hypothesis says that a deep neural network will do better on lists of transactions that have been properly pre-processed and feature-selected than several machine learning methods, including KNN, Naive Bayes, SVM, RF, HMM, and LR. This is in contrast to classifying individual transactions.

In this thesis, we seek to address several research questions that arise from our hypothesis:

a. Which features and pre-processing methods are important for the fraud detection model?

b. How may imbalanced datasets be handled?

c. In the context of imbalanced data sets, what level of performance can be attained, and what constitutes a meaningful performance metric?

d. Can the model learn intricate patterns and anomalies associated with fraudulent transactions, enabling it to accurately detect and prevent fraudulent transactions in real time?

e. Furthermore, can the model adapt to evolving fraud schemes and maintain high performance over time through continuous updating with new data and the incorporation of feedback mechanisms?

2. Technical Goals

The creation of computer systems that are capable of performing tasks that intelligent humans typically carry out is a subfield of computer science known as AI. Numerous businesses are already using AI to increase productivity and accuracy in their operations. Artificial intelligence is a perfect tool for preventing fraud since it can make decisions and take in information far more quickly than people. It can continuously check transaction data for anomalies and highlight any actions that appear to stray too much from the usual.

In fraud detection systems, machine learning is a form of artificial intelligence that finds patterns in data and uses that information to anticipate future events. A wide range of industries, including healthcare, insurance, and financial services, employ it for fraud detection and prevention. Because fraud can take many forms, companies use machine learning to detect suspicious activity and use it to identify fraud, prevent it from occurring in their systems, and protect their customers from identity theft and financial loss.

An organization might, for instance, utilize AI to train a machine learning system to identify typical financial habits that point to fraud risk based on daily credit card transactions.

Machine learning is the field of computer algorithms that use data to anticipate future events. Machine learning algorithms can be applied to detect trends in data that point to fraudulent activities to avoid fraud. To make it easier to identify fraudulent transactions, a business might, for instance, utilize ML algorithms to evaluate daily credit card transactions and search for unusual spending patterns that point to a fraud risk.

A subclass of machine learning techniques known as "Deep Learning" uses several non-linear processing units to extract and transform features. Hierarchically, these processing units identify intermediate representations. The properties identified in one layer serve as the foundation for the processing of the layer before it. According to (A. Roy et al., 2018), deep learning algorithms acquire intermediate notions in this manner between raw input and target.

According to (Hinton et al., 2012)(Collobert & Weston, 2008), the first applications of deep learning were in automatic speech recognition, image recognition, and natural language processing. Deep learning algorithms have been applied more recently to fields including mobile advertising, recommendation systems, and customer relationship management, which are required for the prediction of human behavior (Ngai et al., 2011)(Zhai et al., 2016).

All these applications, however, necessitate selecting between various Deep Learning topologies (i.e., the model's structure) in addition to the model's construction parameters (Han et al., 2015)(Denil et al., 2013) Deep learning offers a promising solution to the issue of financial fraud detection, according to a McKinsey article published in April 2017. It allows institutions to make the best use of all of their historical customer data as well as real-time transaction details that are recorded at the time of the transaction (Khateeb & Alrabba, 2023).

Examining various DL topologies for their ability to enhance credit card fraud detection in real-time scenarios, this research has systematically explored a range of DL architectures and their associated parameters. The primary objective was to identify the optimal set of parameters that not only effectively detect fraudulent transactions but also minimize false alarms, ensuring the practical applicability and reliability of the model.

3. Contributions

This thesis represents a comprehensive exploration of credit card fraud detection, spanning various contributions, each offering unique or many contributions to the field. These contributions can be summarized as follows:

Effective Imbalance Resolution:

- The first contribution aimed to detect credit card fraud using two artificial neural network algorithms, Multilayer Perceptron (MLP) and Extreme Learning Machine (ELM).
- The second contribution introduced a pioneering approach that combines two architectures: the AE and the MLP, forming a cohesive FCDN. Leveraging the AE to extract pertinent features from input transactions serves to reduce data complexity while retaining essential information. The extracted features are then classified using an optimized, fully connected deep network (FCDN) to obtain the model's optimal parameters.
- The third contribution presented a novel approach that combines AE and FCDN models to address this issue. There are three steps in the process: using fraudulent transactions to train an AE, using another AE to reduce the number of dimensions, and then using the encoded representations as input for FCDN classification. To further enhance the model's performance, we introduced an additional FCDN trained on the preprocessed data using the Synthetic Minority Oversampling Technique (SMOTE). The predictions from both AE, AE-FCDN, and the FCDN are combined using a majority voting approach. This approach offers superior performance to alternative optimization algorithms, particularly real-time credit card fraud detection. AE and FCDN have advanced the capabilities of credit card fraud detection. These models address class imbalance issues, harness ensemble techniques, and outperform existing methods, setting new benchmarks for the field.

Innovative Model Development:

- The development of innovative models that combine efficient model optimization. The optimization technique introduced in the contribution entitled “Detecting Fraudulent Transactions Using Stacked AE Kernel ELM Optimized by the Dandelion Algorithm” (S-AEKELM-DA) has simplified model architecture, eliminated subjective node selection, and enhanced computational efficiency. This contribution paves the way for more efficient and less time-consuming model development.

Practical Insights and Recommendations:

- Our work contributes valuable insights and recommendations for enhancing financial security measures and reducing potential losses from credit card fraud. These recommendations, informed by the findings of our research, provide actionable guidance for stakeholders and organizations concerned with safeguarding financial transactions in the modern era.

Enhanced Model Efficiency:

- Across the three contributions, our research emphasizes the importance of enhancing model efficiency, both in terms of computational complexity and execution time. These contributions have a direct impact on the practical applicability of credit card fraud detection systems.

Collectively, these contributions demonstrate significant advancements in credit card fraud detection, offering innovative models, efficient optimization techniques, and real-world solutions to address the challenges of class imbalance and improve the reliability and accuracy of detection systems. Our work not only enriches the academic understanding of the subject but also provides practical tools and insights for enhancing financial security in a rapidly evolving digital world.

4. Thesis structure

Three chapters make up this thesis. The lines that follow give a brief overview of each chapter:
Chapter One: [State of the Art] presents the CCF detection background by navigating through various techniques, from data mining to sophisticated approaches in deep learning and metaheuristics algorithms.
Chapter Two: [Imbalance Resolution for Credit Card Fraud Detection] serves as a comprehensive guide to the proposed models, offering a detailed exploration of their architecture and the components employed to resolve imbalanced data.
Chapter Three: [Optimization Model and Performance Improvement] talks about a new way to find fraudulent transactions called stacked AE kernel ELM optimized by the dandelion algorithm (S-AEKELM-DA). The primary objective was to enhance the kernel extreme learning machine (KELM) performance by integrating the dandelion technique into a stacked AE architecture.
Finally, a general conclusion and perspectives serve to summarize key findings and insights from the study and offer recommendations for future development. Additionally, this research aims to provide practical insights and recommendations for financial institutions and individuals to fortify their defenses against credit card fraud in the modern era.

Chapter I: State of The Art

1. Introduction

Financial fraud is a global phenomenon that poses a significant threat to financial institutions, consumers, and the economy as a whole. Technological advancements and the increasing sophistication of fraudsters have made detecting and preventing financial fraud more complex than ever before. In this context, it is imperative to gain a deep understanding of the various facets of this problem and explore the techniques and methods used to address it.

This chapter aims to provide a detailed overview of the current state of research and practices in the field of financial fraud detection, with a particular focus on credit card fraud. We will begin by examining the different types of financial fraud impacting financial institutions and consumers. Subsequently, we will focus on the specific techniques used to detect credit card fraud, analyzing their strengths and weaknesses.

We will also explore the major challenges facing fraud detection techniques, including the increasing complexity of fraud schemes, technological limitations, and regulatory constraints. Understanding these challenges is essential for developing effective strategies for fraud prevention and detection.

Finally, we will address the practical implications of fraud detection for financial institutions, consumers, and society as a whole. We will highlight the importance of collaboration between the public and private sectors as well as ongoing innovation in the field of fraud detection to effectively address this complex challenge.

2. Types of financial fraud

Due to the extensive use of credit cards, credit card fraud continues to be a common form of fraud, even with this attention. Therefore, data relevant to the goals of this research may be obtained from studies conducted in the financial industry. Thus, the methods and research used by the financial industry were reviewed in the section that follows from 2009 to 2019 (Al-Hashedi & Magalingam, 2021). The primary categories of financial fraud are graphically depicted in Figure 1.

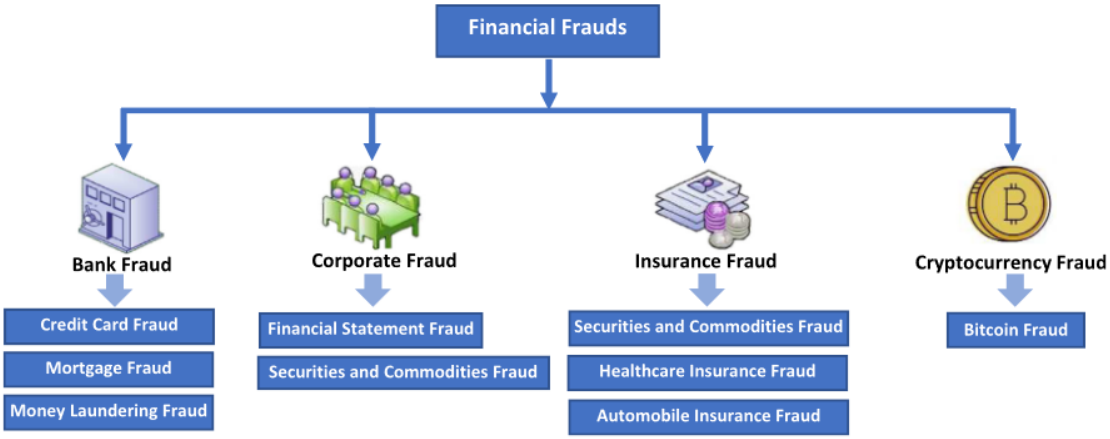


Figure1. Common types of financial fraud (Al-Hashedi & Magalingam, 2021)

2.1. Mortgage Fraud

Mortgage fraud refers to the intentional misrepresentation or manipulation of mortgage transactions for personal gain or profit. This type of fraud can manifest during loan applications or property purchases and can take various forms. Examples include inflating property valuations, providing false information about income and employment, or misrepresenting intentions regarding property occupancy to qualify for larger loans. Other tactics involve using individuals with good credit as straw buyers to obtain loans for those with poor credit, participating in property flipping schemes to artificially inflate property values, and concealing secondary loans obtained for down payments. Mortgage fraud results in significant financial losses and legal consequences for lenders, borrowers, and the overall stability of the housing market. To combat these fraudulent practices, lenders employ stringent verification procedures and utilize fraud detection technologies.

2.2. Money Laundering

Money laundering involves the deliberate conversion of illegally obtained revenue through financial transactions to conceal its true source or nature. By portraying these funds as legally acquired, the goal is to create the illusion of legitimacy. Money laundering has detrimental effects on society as a whole, exacerbating criminal behavior and perpetuating a cycle of crime. Criminals and organized groups blur the lines between lawful and unlawful financial activities, impeding investigations into illegal behavior and enabling criminals to reinvest "cleaned" money back into their operations. Moreover, money laundering facilitates various other criminal activities, including terrorism financing and illicit arms trading, posing serious threats to global stability and security. Efforts to combat money laundering are crucial to dismantling

these complex criminal networks and safeguarding the integrity of financial systems and society at large.

2.3. Financial Statement Fraud

Business financial statements are in-depth records that detail their current financial condition and previous financial activity (Doshi et al., 2023). Loans, income, expenses, and profits are just a few of the financial details that are included in these accounts. They also offer managerial analysis to describe the success of the business and foresee potential issues in the future. These financial records are necessary for determining the health and viability of an organization as well as for gauging its general success and bankability.

These financial records provide a thorough picture of the organization's state and serve as the foundation for the financial statements of the organization. This data helps determine the organization's level of achievement and financial capability. Sadly, there are instances of financial statement fraud where data is changed to create the appearance of being more favorable than it is. (Mienye & Sun, 2021) involves rectifying incorrect claims to make the organization appear more favorable than it is. Several sources with a variety of objectives are driving these fraudulent activities. These can entail inflating stock prices, obtaining personal bank loans using false information, reducing tax liabilities using false information, or enticing as many investors as possible with an exaggerated picture of financial security. Financial statement fraud compromises financial data and deceives stakeholders who depend on accurate reporting to make informed decisions. Such fraudulent activities need to be addressed if financial reporting systems are to continue to be open, reliable, and reputable.

2.4. Insurance Fraud

Insurance fraud entails purposefully committing unlawful acts to exploit insurance coverage for financial gain at the expense of insurance companies. The primary objective of insurance is to mitigate potential financial risks and protect the financial interests of individuals and organizations alike. Various parties, including agents and policyholders, may be implicated in instances of insurance fraud, which can occur at any stage of the insurance process. Such fraudulent activities involve fabricating or exaggerating loss-related assets or staging accidents to inflate repair and injury costs unjustly for financial benefit. Insurance fraud pervades multiple sectors, including healthcare, crop protection, and automotive coverage.

In cases of auto insurance claims, perpetrators may forge documents with inflated expenses purportedly linked to staged motor accidents. Similarly, within healthcare insurance,

individuals may file false claims for nonexistent medical services to illicitly obtain compensation for costly procedures or treatments. Moreover, fraudulent practices extend to the crop insurance industry, where actors exploit losses attributed to fluctuations in agricultural prices or unforeseen natural disasters. These dishonest activities not only result in significant financial repercussions but also erode the fundamental trust essential to insurance transactions. A comprehensive investigation and robust countermeasures are essential to identify, prevent, and mitigate insurance fraud due to its far-reaching consequences (Sadgali et al., 2018). Understanding the intricacies and diverse manifestations of insurance fraud across various domains is imperative for devising targeted strategies that uphold the integrity of insurance systems and promote transparency and accountability.

2.5. Cryptocurrency Fraud

Cryptocurrency fraud refers to a specific type of deception aimed at deceiving unsuspecting individuals through false promises of investment opportunities or services. These fraudulent schemes target less knowledgeable individuals by enticing them with the prospect of significant financial gains. Many individuals, particularly those unfamiliar with online investing, become ensnared in these schemes, often resembling Ponzi schemes, due to their lack of awareness about the intricacies of such investments.

Recognizing that malicious actors exploit Bitcoin for illicit activities, including theft and unauthorized use of digital currencies, is crucial. The inherent decentralization and minimal regulation of cryptocurrencies contribute to the prevalence of fraudulent activities within the Bitcoin industry, deliberately targeting unsuspecting individuals. These deceptive practices lure numerous unwitting victims, resulting in substantial profits, often amounting to millions of dollars. Vasek and Moore's research (Vasek & Moore, 2019) highlights several instances of cryptocurrency fraud, such as btcQuick, Coin Opened, Ubitex, and BTC Promo, collectively amassing an estimated $11 million. The unique allure of cryptocurrencies and their decentralized nature inadvertently facilitate the proliferation of scams, emphasizing the urgent need for robust defenses and heightened public awareness.

2.6. Credit Card Fraud

Credit refers to the practice of conducting electronic monetary transactions without the use of physical currency, commonly facilitated through credit cards, which store customer information and credit services. This method has become integral to electronic banking and is extensively

utilized for online transactions and e-commerce activities. However, the proliferation of credit card usage has coincided with a surge in various forms of fraudulent activity. Fraudsters exploit credit cards for illicit transactions, resulting in substantial financial losses for both financial institutions and cardholders. Furthermore, the proliferation of counterfeit cards has facilitated the execution of fraudulent activities. To identify credit card fraud, suspicious transactions are typically categorized as either genuine or fraudulent/suspicious. Individuals who conduct legitimate transactions do so in a legal, open, and honest manner without using dishonest means. Conversely, unauthorized or irregular access to the system by individuals leads to counterfeit or suspicious transactions, often causing significant disruptions. Unauthorized credit card fraud occurs when a credit card is used without the owner's consent, with the legitimate cardholder unaware of the fraudulent activity. Consequently, scammers may gain unauthorized access to a user's account or conduct fraudulent transactions. Credit card fraud primarily manifests in two forms: online and offline. Online fraud involves perpetrators conducting transactions, particularly online purchases, via web browsers, mobile devices, or the internet. Offline fraud occurs when criminals use stolen credit cards to make purchases while posing as authorized users.

Credit card fraud occurs in a variety of ways. Here are six of the most common types.

Card-not-present fraud. Card-not-present fraud happens when someone uses your credit card information, but not your actual card, to carry out an unauthorized transaction online, over the phone, or by mail.

Identity theft. Identity theft takes place when someone uses your personal information, such as your Social Security number or credit card number, to commit fraud or other crimes. In 2021, the Federal Trade Commission received nearly 1.4 million reports of identity theft.

Skimming. Skimming happens when criminals illegally place devices on ATMs, point-of-sale terminals, or gas pumps to steal data, including card numbers and PINs. With this information, crooks can generate phony credit or debit cards. Financial losses from skimming are estimated to be more than $1 billion each year.

Phishing. Phishing enables fraudsters to steal your information, such as passwords or account numbers, through emails or text messages. Criminals can then gain access to your credit card, bank, and email accounts. They might also sell your information to other crooks.

Account takeover. Account takeover occurs when a cybercriminal assumes ownership of someone else’s online account with a stolen username or password. Generally, crooks buy usernames and passwords on the so-called “dark web”.

Lost or stolen credit cards. If you lose your credit card or it's stolen, someone else may use it to make unauthorized transactions. You should report a lost or stolen card to the card issuer as soon as possible.

3. Credit card fraud detection techniques

Credit card fraud detection is the collective term for the policies, tools, methodologies, and practices that credit card companies and financial institutions take to combat identity fraud and stop fraudulent transactions. In recent years, as the amount of data has exploded and the number of payment card transactions has skyrocketed, credit fraud detection has become largely digitized and automated. Most modern solutions leverage artificial intelligence (AI) and machine learning (ML) to manage data analysis, predictive modeling, decision-making, fraud alerts, and remediation activities that occur when individual instances of credit card fraud are detected. Various methodologies are employed to detect fraudulent credit card transactions, which can be categorized and identified as follows:

3.1. Data Mining Techniques

Here are several key data mining techniques utilized in fraud detection

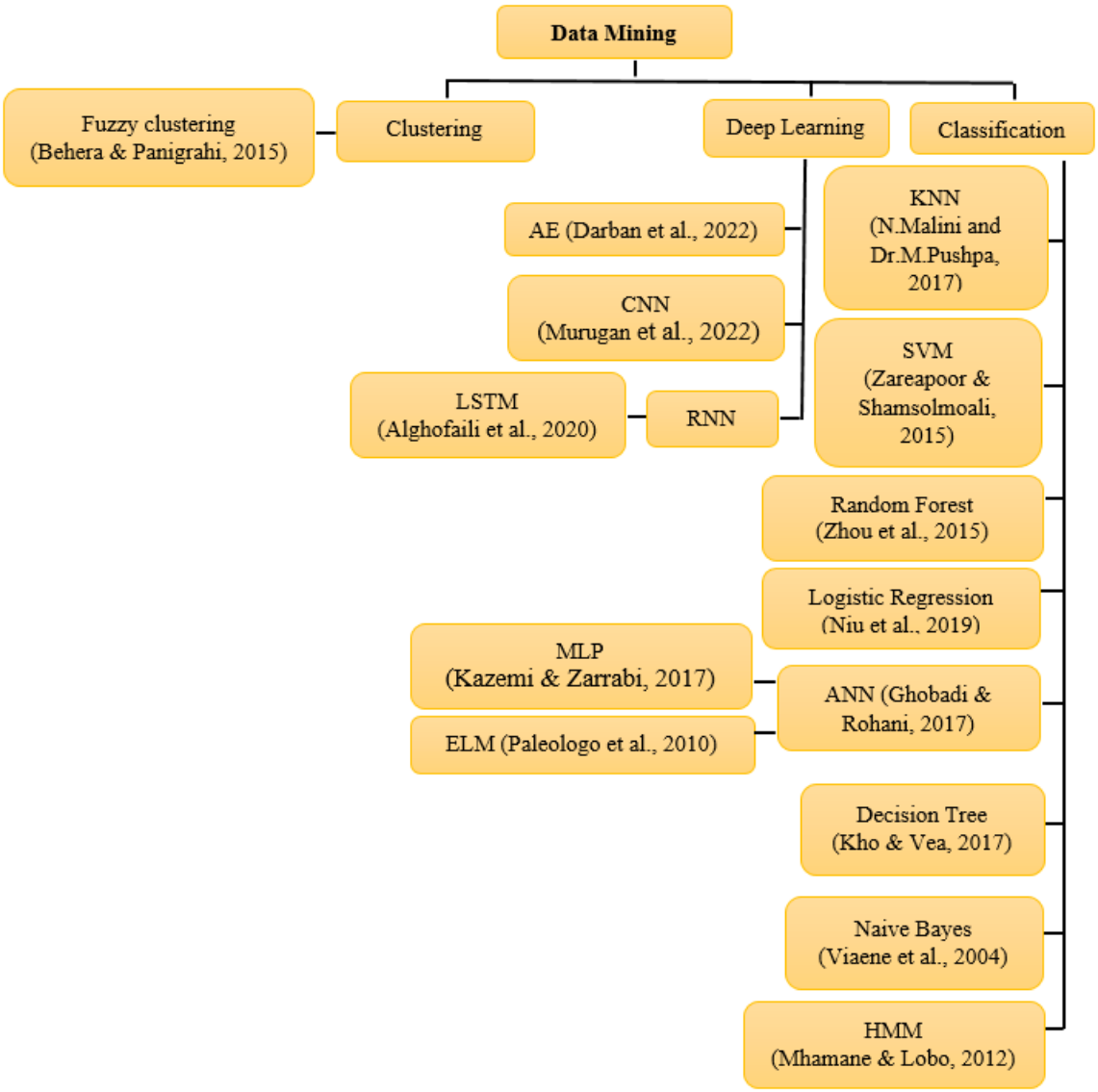


Figure2. Taxonomy of techniques used for fraud detection

3.1.1. Random Forest

Random forest, one of the ensemble methods, is a combination of multiple tree predictors such that each tree depends on a random independent dataset and all trees in the forest are of the same distribution (Chandola et al., 2009). The capacity of a random forest depends not only on the strength of individual trees but also on the correlation between different trees. The stronger the strength of a single tree and the less the correlation of different trees, the better the performance of the random forest. The variation of trees comes from their randomness, which involves bootstrapping samples and randomly selecting a subset of data attributes (Xuan et al., 2018).

Random forest has been used in many areas. (Zhou et al., 2015) used to solve the problem of cost-sensitive feature selection. The algorithm puts the feature cost into the construction process of a basic decision tree and then generates a low-cost feature subset. When constructing a base tree, features are randomly selected. The probability of the feature is inversely proportional to the cost associated with it. (Xuan et al., 2018) make a comparison for different implementations of random forest for credit card fraud detection. (Winham et al., 2013) put forward a weighted random forest to improve predictive performance. They compute the weight based on the out-of-bag error of the decision tree. (H. B. Li et al., 2009) proposed the TWRF method and employed it to classify high-dimensional noisy data. They add some noisy features to the data to make it more difficult to deal with. As they said, it will be better to verify the effectiveness of their algorithm (Seyedhossein & Hashemi, 2010) proposed a method of aggregating profiles that exploits the inherent patterns in time series of transactions, and fraud detection is performed online at the end of a day or the end of a period, respectively. In their work, they evaluate and compare several techniques, such as support vector machines and random forests, for predicting credit card fraud, and the conclusion is that random forests have the best performance among these techniques with the process of aggregation. However, the aggregated method in this work fails to detect fraud in real time.

3.1.2. Decision Tree

Decision trees use a binary tree with successive nodes to classify the data. As the sample traverses the tree, according to the node being created, the tree is divided into subsets and eventually stored in mutually exclusive subgroups (Popat & Chaudhary, 2018). It is also referred to as a classification and regression tree (CART)(Lucas & Jurgovsky, 2020).

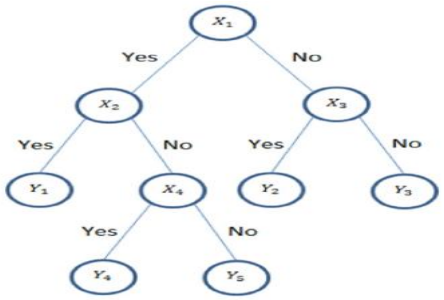


Figure3. Binary Decision Tree (West & Bhattacharya, 2016)

In a study by (Kho & Vea, 2017), an investigation into the transactional behavior of credit card users was conducted to discern between legitimate and anomalous transactions. Employing a range of classification techniques such as Naive Bayes (NB), Bayes-Net, J48, Random Tree

(RT), libSVM, and MOLEM, the researchers aimed to determine the most efficient technique. The study utilized two synthetic datasets for analysis. Upon evaluation, it was revealed that Random Tree (RT) and J48 emerged as the top-performing classifiers. Notably, Random Tree exhibited superior accuracy, achieving a rate of 93.50% compared to J48's 94.32%. In their study, (J & K.S, 2017) introduced a supervised methodology aimed at categorizing credit card transaction data into normal or suspicious categories through the comparison of several classification techniques, including Support Vector Machines (SVM), Random Forest (RF), and Decision Trees (DT). Through rigorous evaluation employing diverse accuracy metrics and measures, the researchers determined that DT exhibited superior performance among the classification methods, boasting notably high accuracy. However, within the realm of traditional insurance fraud detection, instances where individuals submit falsified insurance claims to illicitly obtain benefits present a significant challenge, as existing systems may struggle to accurately identify them as fraudulent. Therefore, in their investigation, (R. Roy & K, 2017) directed their attention toward the identification of auto or vehicle fraud through the utilization of machine learning methodologies. The writers looked at three different classification algorithms side by side: Decision Trees (DT), Random Forest (RF), and Naïve Bayes (NB). They wanted to make insurance fraud detection more accurate and precise by using raw data with a set of attributes. The findings of the study demonstrated the superior performance of DT and RF techniques over Naïve Bayes (NB).

The issue of imbalanced data has long been a challenge affecting detection accuracy. (Anusandhan, 2018) proposed a novel approach aimed at identifying fraud within the auto insurance sector, employing an adaptive oversampling technique to rectify imbalanced classes within the insurance dataset. The authors of the study used three supervised classifiers on the training dataset: Multi-Layer Perceptron, Support Vector Machine (SVM), and Decision Tree (DT). This helped them make relevant classifier models that could tell the difference between abnormal classes and normal ones. Additionally, the authors incorporated the 10-fold cross-validation method to enhance the performance of the classifier models. The outcome revealed a notably high fraud detection rate utilizing SVM and DT classifiers.

3.1.3. Support Vector Machine

Support vector machines transform the linear issue into a higher dimensional feature space. This allows sophisticated, non-linear issues like credit card fraud detection to be solved using linear classification, without increasing computational complexity. Kernel function issue to

remodel the dataset. It is considered as a mapping between the input space point and a higher dimensional space point. The kernel function is outlined by:

$$K(X_1, X_2) = < \Theta(X_1), \Theta(X_2) > \quad (1)$$

Where $\Theta: X \rightarrow H$ is used to map the point between input space X and higher dimensional space H. When the kernel function is, applied to the dataset, a hyperplane is employed for class separation, which is in the form of:

$$<W , X> + b = 0 \quad (2)$$

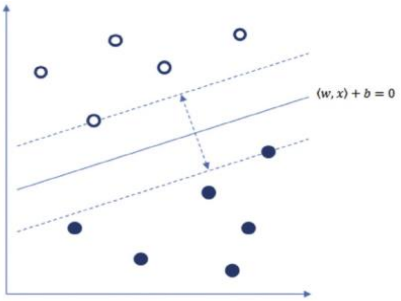


Figure4. Separation between Support Vector and Hyperplane (Popat & Chaudhary, 2018)

As shown in Figure 3, the hyperplane is built in the simplest way to maximize the separation between both classes. After that, it helps in reducing the potential errors generated by overtraining. The SVM classification is outlined below:

$$\sum_i \alpha i Y i K(X_i, X) + b = 0 \quad (3)$$

There are several kernel functions, like the Gaussian radial basis function and the polynomial function (Bhattacharyya et al., 2011). The dataset and the necessity of classification define which function we have to apply.

(Xu & Liu, 2012) conducted a study aimed at identifying fraudulent transactions in online credit card usage through the implementation of an optimized Support Vector Machine (SVM) model. The efficacy of the proposed model was evaluated using a dataset sourced from a commercial bank's operations to establish a classification training set. A comparison test was done between the suggested SVM model and an ID3+BP hybrid model. The results showed that SVM performed better, proving that it can be used. A different study, (Rajak & Mathai, 2016) described an intelligent way to find fraud using a hybrid model that combines the SVM and Danger Theory optimization algorithms. Through their investigation, the authors demonstrated that this hybrid approach surpassed previous methodologies utilizing solely SVM, particularly in terms of F-measures and time complexity.

(Zareapoor & Shamsolmoali, 2015) conducted an assessment of three advanced credit card fraud detection techniques, namely K-Nearest Neighbors (KNN), Naïve Bayes (NB), and

Support Vector Machine (SVM). Because bank fraud is getting worse, these methods were tested on a real-world credit card dataset from the UCSD-FICO competition using bagging ensemble classifiers. The findings indicated that the bagging classifier employing the trio of methods exhibited commendable evaluation performance, making it suitable for credit card fraud detection. (Gyamfi & Abdulai, 2019) in their paper introduced supervised learning methods utilizing a combination of SVM to discern between legal and illegal customer behavior in credit card transactions. The authors employed a blend of SVM, linear regression, and logistic regression to enhance the accuracy of fraud detection. While the results confirmed the effectiveness of the proposed techniques in detecting banking fraud, they also provided novel insights into the application of supervised learning methods for fraud detection in credit card transactions.

Their study, (Mareeswari & Gunasekaran, 2016) introduced a novel approach to detecting credit card fraudulent behavior by employing a hybrid support vector machine (HSVM), communal, and spike detection methods aimed at addressing the limitations inherent in existing systems. Transaction classification into fraudulent and legitimate categories was conducted based on predictions using a credit card dataset. The authors demonstrated that the proposed method yielded superior results compared to the two existing methodologies.

In a separate study, (Sundarkumar et al., 2016) proposed a methodology to enhance fraud identification within insurance companies utilizing a one-class support vector machine (OCSVM)-based under-sampling technique. Various classification techniques, including Logistic Regression, Group Method of Data Handling (GMDH), Support Vector Machine (SVM), and Decision Tree (DT) were employed. The results of the study indicated that Decision Tree (DT) outperformed other classification techniques while simultaneously reducing system complexity.

(Francis et al., 2011) In their research, Francis et al. (2011) investigated the design of an automated medical billing system geared towards identifying fraudulent activities in medical insurance claims, utilizing Support Vector Machine (SVM) technology. The primary objective of this study was to expedite the process of medical fraud detection, thus providing crucial support to experts in this field. The outcomes of their research indicated a notable enhancement in the accuracy of fraud detection within insurance claims, thereby facilitating faster identification by medical fraud detection experts.

While significant strides have been made in the realm of credit card fraud detection in recent years, challenges persist when employing machine learning techniques for fraud identification in credit card transactions. (Mishra et al., 2018) talked about problems like the uneven

distribution of training data and the lack of marked-up fraudulent transactions, which might make it harder for fraud detection methods to work correctly.

(Jeragh & Alsulaimi, 2019) In their work, Jeragh and Alsulaimi (2019) proposed an innovative unsupervised learning model for detecting credit card fraud that combined a One-Class Support Vector Machine (OSVM) with an autoencoder. The authors applied this combined model to the Kaggle credit card dataset, employing measures and metrics to address the dataset's inherent skewness. Despite this challenge, the integrated techniques successfully distinguished between fraudulent and legitimate transactions, yielding promising results, particularly when evaluated using performance metrics.

Additionally, (Qingshan, 2009) introduced a model for uncovering fraudulent financial statements leveraging Support Vector Machine (SVM) technology. The author standardized the dataset to mitigate the influence of variable dimensions and conducted various experiments using different sets of variables. The outcomes of these experiments supported the findings of previous studies, suggesting deliberate falsification within the issued financial statements.

3.1.4. Naïve Bayes

The naive Bayes (NB) classifier is widely used in supervised learning and is often regarded as the simplest type of Bayesian network classifier (Friedman et al., 1997). This model learns the conditional probabilities of each attribute in the training dataset, given the class label. Classification is performed by applying Bayes' theorem to calculate the probability of the label given the features or attributes of the instance. A crucial assumption for this computation's validity is that all features in the dataset are independent of each other, given the class labels' values. With input data denoted as x = (x1,⋯, xn) ∈ R^n and target labels y, this structural assumption simplifies the model to the following form:

$$p(x|y) = \prod_{k=1}^{n} p(x_k|y) \quad (4)$$

The idea that predictors are independent of the class is often seen as too strict in many areas. However, the naive Bayes (NB) classifier has always done well, as shown by many real-world studies (Viaene et al., 2004)(Viaene et al., 2002). Remarkably, this holds even when the independence assumption is considered unrealistic (Friedman et al., 1997)(Viaene et al., 2004) employed NB classifiers to predict fraudulent automobile insurance claims, building on their previous research (Viaene et al., 2002). They proposed enhancing the model's performance through the application of the AdaBoost algorithm (Freund et al., 1996). The fundamental

concept behind boosting involves sequentially constructing classifiers trained on various iterations of the original training set.

Reweighting or resampling creates the various training sets, with the next model giving more weight to any data instances that one of the classifiers in the sequence had a difficult time classifying. Upon termination of training after a fixed number of iterations, the classifiers are then combined using a majority voting scheme (Hilal et al., 2022). In AdaBoost, each classifier's vote is adaptively weighted according to its quality of classification (Ridgeway et al., 1998). The concept of perturbing the data in stages allows the NB model to focus incrementally on the regions of data that are more difficult to learn (Viaene et al., 2004). The proposed model is extended further with a weighted evidence-based voting formulation for the case of NB with AdaBoost, as proposed by (Ridgeway et al., 1998). While retaining the predictive performance of the NB classifier, the weights of evidence voting enable more interpretable results and explanations that AdaBoost would otherwise eliminate. In the literature, the Massachusetts automobile claim data set was used, which consisted of 1,399 samples of data. The proposed naïve Bayes AdaBoosted weights of error (ABWOE) classifier, which was made up of 25 base classifiers, was trained on two-thirds of the dataset. The model was compared against a simple NB classifier, an AdaBoosted classifier (AB), as well as a model using majority voting instead of the proposed method. The models were judged on their accuracy, AUC, and a logarithmic score (L) with $\{L \in [0,\infty]\}$, with 0 being the best score (Viaene et al., 2004)(Hilal et al., 2022). The proposed ABWOE classifier outperformed the rest of the models for all performance measures, with 84.43% accuracy, 89.19% AUC, and the lowest L of 0.3697. The proposed voting framework is confirmed to be superior, as the model with majority voting had an AUC of 50%, indicating no discrimination capacity and output predictions are random guesses. The plain NB had an accuracy of 83.03% and an AUC of 88.56%, slightly shy of the ABWOE's performance. A similar study conducted by (Phua et al., 2004) studied the effects of using a hybrid stacking-bagging training approach using an NB, C4.5 DT, and MLP. The purpose of the research is to address the issues of skewed or imbalancedness associated with the poor performance of classifiers. Bagging makes use of base classifiers such as NB, fitting them with randomly sampled subsets of the training data. Each base classifier then votes on the class of a sample, and the majority vote is chosen as the final classification (Hilal et al., 2022).

3.1.5. K-Nearest Neighbors

The K-nearest neighbor (KNN) algorithm is widely employed in detection systems and has demonstrated effectiveness in credit card fraud detection using supervised learning techniques. KNN classifies new instance queries by how close they are to existing data points, according to its 1991 introduction by Aha, Albert, and Kibler. The performance of KNN depends on three key factors: the chosen distance metric for neighbor determination, the distance rule utilized for classification, and the number of neighbors considered in the classification process.

When looking at different supervised statistical pattern recognition methods for finding credit card fraud, KNN always does very well without making any assumptions about how the data is distributed. For KNN-based fraud detection methods to work, they need to guess two important things: the distance or similarity measure between data instances and the number of K neighbors that should be used for classification. By calculating the nearest data point to each incoming transaction, KNN can effectively identify fraudulent transactions. Increasing K values aids in noise reduction within the dataset.

Different distance metrics are utilized depending on the nature of the attributes. For continuous attributes, the Euclidean distance is commonly employed, while categorical attributes may use the Jaccard coefficient. For multivariate data, distances are computed for each attribute and then aggregated. Optimization of the KNN algorithm can be achieved through the selection of appropriate distance metrics.

According to (Chandola et al., 2009) and (Savage et al., 2014), the Euclidean distance is one of the most widely used methods for computing distance. (N.Malini and Dr.M.Pushpa, 2017) analyzed credit card fraud employing both outlier detection and the K-Nearest Neighbor (KNN) algorithm. The authors examined the performance outcomes of these techniques within the credit card approval system, with empirical evidence indicating the accuracy and effectiveness of the KNN method in fraud detection.

Furthermore, (Indrajani et al., 2017) presented a study focused on recognizing fraudulent transactions in debit card usage, aiming to address the challenge identified in a prior study utilizing the Hidden Markov Model (HMM). The authors combined Chi-Square Automatic Interaction Detection (CHAID) and k-Nearest Neighbor (k-NN) to increase the accuracy of finding strange transactions. The findings of this study demonstrated that the CHAID method significantly improved performance in recognizing fraudulent debit card transactions. (Badriyah et al., 2018) proposed a methodology for detecting fraud in auto insurance, utilizing a Nearest Neighbor-based approach that incorporates three distinct methods: density-based, distance-based, and interquartile range, applied to car insurance data. The study emphasizes the

examination of the impact of feature selection processes on accuracy scores. The results of the investigation revealed a notable enhancement in detection accuracy with the utilization of feature selection methods.

In a separate study, (Awoyemi et al., 2017a) assessed the performance of three machine learning models applied to credit card transactions for identifying fraudulent behavior. The authors used a comparison test to compare the Logistic Regression (LR), Naïve Bayes (NB), and K-Nearest Neighbor (K-NN) methods. They used a credit card dataset from European cardholders that included 284,807 transactions. K-Nearest Neighbor (K-NN) did better at finding fraudulent transactions than both Logistic Regression (LR) and Naïve Bayes (NB), according to the results of the comparison study.

3.1.6. Logistic Regression

Logistic regression is used for binary classification, which is based on statistical methods. It uses a linear model (Ngai et al., 2011). Hence, it is used to perform regression on a group of variables (Ravisankar et al., 2011). It is a commonly used technique for predicting patterns in data with unambiguous or numeric attributes (Ngai et al., 2011). To calculate probability, it uses a series of input vectors and a dependent response variable, using a logarithm. Probability lies with the specific class. For binary classification, the response variable is given below:

$$Yi=\begin{cases}0\\1\end{cases} \quad (5)$$

Hence, the formula for calculating within a sample xi belongs in class one is given by:

$$P(Yi=1/\ Xi) = \frac{\exp(W0+W^{T}Xi)}{1+\exp(W0+W^{T}Xi)} \quad (6)$$

Where W0 and W are the regression standardization parameters, W0 represents the intercept and W represents the coefficient vector (Ravisankar et al., 2011).

(Sahin & Duman, 2011)have built 13 alternative models based on ANN and LR. ANN models give more accuracy than Logistic Regression when their performance is compared over the test datasets. (Niu et al., 2019) performed a comparison study of credit card fraud detection by using various supervised and unsupervised approaches. In particular, this study looks at 6 supervised classification models: Logistic Regression (LR), K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Decision Tree (DT), Random Forest (RF), and Extreme Gradient Boosting (XGB). It also looks at 4 unsupervised anomaly detection models: One-Class SVM (OCSVM), Auto-Encoder (AE), Restricted Boltzmann Machine (RBM), and Generative Adversarial

Networks (GAN). The experimental results show that supervised models perform slightly better than unsupervised models in this study.

3.1.7. Hidden Markov model (HMM)

The Hidden Markov Model (HMM) represents a sophisticated dual-embedded stochastic process, offering a more intricate modeling approach compared to traditional Markov models. In this case, the platform works like a Markov process with no states that can be seen. The HMM states stay hidden, but their outputs that depend on their states can be seen (Zojaji & Atani, 2016). (Khan et al., 2012) employed the HMM framework to analyze cardholder behavior by monitoring transaction activities. They used a method that divided transaction values into low, medium, and high levels. Then, they used clustering algorithms to separate legal transactions from ones that might be fraudulent using data from all the parameters. The experimental findings underscored the efficacy of HMM in simplifying system complexities.

Their study, (Mhamane & Lobo, 2012) focused on enhancing the detection of Internet bank fraud through the utilization of Hidden Markov Models (HMMs). Their research aimed to identify legitimate users and detect suspicious behaviors by proposing a novel framework. The outcomes of their investigation addressed the limitations of prior research and notably improved detection accuracy. Similarly, (X. Wang et al., 2018) conducted simulation experiments aimed at enhancing the effectiveness of bank fraud detection. Their study targeted the detection of online payment fraud utilizing HMM in conjunction with K-means clustering analysis applied to real-world bank transaction data. Using K-means to group the data and HMM to train on normal account activity, they were able to find a significant improvement in their ability to spot fraudulent bank transactions.

In a related study, (Srivastava et al., 2008) conducted research aimed at augmenting the accuracy and efficacy of identifying credit card frauds using HMMs. Additionally, the authors incorporated the K-means technique for clustering purposes to identify cluster centroids and consolidate them into cohesive groups. However, the outcomes of this investigation revealed a high rate of false alerts and suboptimal performance.

3.1.8. Fuzzy Logic-based system Fuzzy

Fuzzy Logic (FL) emerges as a potent conceptual framework for managing the representation of data within environments characterized by vagueness and imprecision. FL operates on the

principle that modes of thought are approximated rather than precise, rendering it particularly well-suited for human-like reasoning, particularly in scenarios involving almost logical inference. The integration of FL with Fuzzy combinations offers sophisticated methodologies for addressing complex modeling challenges in novel and expansive ways (Supraja & Saritha, 2018).

Their research, (Behera & Panigrahi, 2015) conducted a study focused on detecting credit card fraud to minimize false alarms. Their approach involved classifying transactions into fraudulent and non-fraudulent categories using a hybrid technique based on two methodologies: fuzzy c-means clustering and neural networks. Due to the unavailability of real credit card datasets, the evaluation was performed on a synthetic dataset. The authors posited that leveraging a combination of clustering techniques and learning mechanisms could enhance the identification of suspicious activities while simultaneously reducing false alarms. Their findings demonstrated that the integrated use of fuzzy clustering resulted in a notable improvement in the detection rate, achieving a True Positive (TP) rate of 93.90% and a False Positive (FP) rate of less than 6.10%.

(Haratinik et al., 2012) introduced a hybrid model named FUZZGY, which integrates fuzzy logic and the Fogg behavioral model to identify suspicious behavior within credit card transactions, particularly focusing on merchant fraud across various payment systems. The FUZZGY model utilized a fuzzy expert system to analyze historical merchant activities, while the Fogg behavioral model assessed merchant behavior based on two interconnected dimensions: motivation and capability to engage in fraudulent activities. Subsequently, the FUZZGY model calculated the degree of suspicion associated with incoming transactions. The study's findings demonstrated enhanced performance and satisfactory outcomes through the integration of these two methodologies.

In a related study, (Nezhad & Shahriari, 2013) proposed an effective approach to detecting fraud within banking systems using fuzzy logic in conjunction with the Neural-fuzzy Takagi–Sugeno training method. Their research aimed to enhance the accuracy of detecting both fraudulent and non-fraudulent patterns in banking transactions, particularly addressing fraud targeting cash card services, particularly those associated with cash-dispenser operations. The authors formulated fuzzy logic rules based on expert knowledge and trained the Neural-fuzzy model using over 350 input data points to improve detection accuracy. The results indicated that incorporating Neural-fuzzy during the training phase significantly enhanced fraud detection within financial systems.

Various data mining methodologies have been put forth to predict and analyze fraud, yet certain techniques prove inefficient when dealing with datasets containing minimal instances of fraud. In response to this challenge, (Supraja & Saritha, 2018) introduced a model that employs Fuzzy Logic to formulate fuzzy rules to enhance fraud detection. The study's findings revealed that the application of Fuzzy Logic is particularly well-suited for large datasets, exhibiting a high performance level in terms of accurate fraud detection rates.

3.1.9. Extreme Learning Machine

Extreme Learning Machine is a machine learning method developed by (Guang-Bin Huang, 2004) It is a single feedforward neural network. Its structure consists of a single hidden layer feedforward artificial neural network (SLFN), where the weights between inputs and hidden nodes and hidden layer biases are chosen arbitrarily and it remain constant during training and predicting phases. The weights that connect hidden nodes to outputs are determined analytically. ELM uses activation functions such as sigmoid, hard limit, Multiquadric, and Gaussian in the hidden layer, and linear functions in the output layer.

For N arbitrary distinct samples (xi, ti), where Xi = [xi1, xi2, . . . , xid] ϵ R^d and Ti = [ti1, ti2, . . . , tim] ϵ R^m , standard SLFNs with N hidden nodes and activation function g p*q are mathematically modeled as : $\sum_{i=1}^{N} \beta_i g(xj) = \sum_{i=1}^{N} \beta_i g(wixj + bi) = oj\ , j = 1, \dots, N$ (7)

Where wi = [wi1, wi2, . . . , wiN]^T is the weight vector connecting the ith hidden node and the input nodes, βi = [βi1, βi2, . . . , βim]^T is the weight vector connecting the ith hidden node and the output nodes, and bi is the thresholding of the ith hidden node. wixj denotes the inner product of wi and xj.

In their study, (Tripathi et al., 2020) delved into application scoring, a process that assesses the suitability of new applicants based on a range of factors, including social and financial data gathered during the application process. Credit scoring offers several advantages to the credit industry, including the ability to calculate and mitigate credit risk, facilitate managerial decision-making, and enhance cash flow (Paleologo et al., 2010)(Tripathi et al., 2018) . The efficacy of credit scoring models plays a crucial role in the profitability of credit institutions, highlighting the importance of improving their predictive performance, particularly for applicants in the non-creditworthy segment.

Presently, credit scoring extends beyond the realm of banking and credit industries, finding applications in diverse sectors such as telecommunications and real estate, where predictive

models are employed to analyze customer behavior. The integration of artificial intelligence may offer a solution to the challenges associated with manual credit scoring processes.

3.1.10. Artificial Neural Network

The architecture of artificial neural networks (ANNs) draws inspiration from the organizational principles and functions of biological neural networks. Similar to the neurons in the human brain, ANNs consist of interconnected nodes arranged in layers. Among the various types of ANNs, the feedforward neural network stands out as a widely used model. It comprises an input layer for receiving external data, an output layer responsible for generating solutions to problems, and one or more hidden layers serving as intermediaries between the input and output layers. Connections between adjacent neurons from the input to output layers form acyclic arcs. ANNs employ training algorithms to adjust neuron weights based on the disparity between target and actual outputs. One commonly used training algorithm for ANNs is the backpropagation algorithm (Dubey et al., 2020).

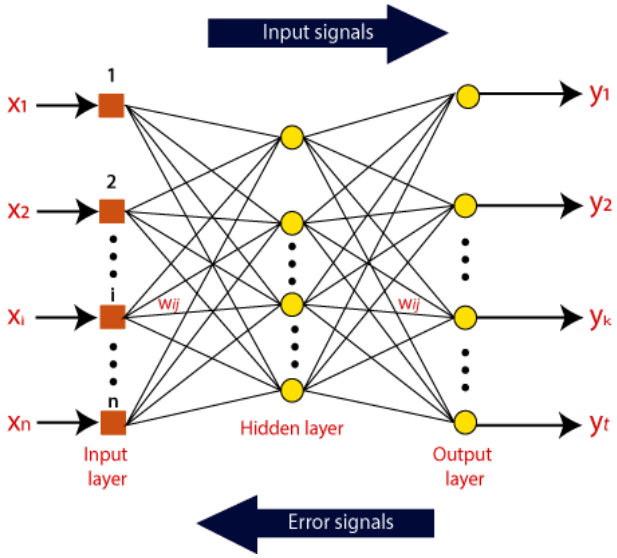


Figure5. Artificial neural network architecture

Their study, (Ghobadi & Rohani, 2017) introduced a hybrid model aimed at detecting credit card fraud, utilizing a cost-sensitive Neural network, and assessing its performance with real transactional data. Recognizing the challenge of imbalanced data due to the limited number of fraudulent transactions in the dataset, the authors employed the Meta Cost procedure to address this issue. They conducted comparative analyses between their proposed model, a Neural Network, and an AIS-based Fraud Detection Model to demonstrate the efficacy of their

approach. Their findings revealed that the proposed method not only increased the detection rate but also minimized the cost associated with false negatives.

In their study, (Randhawa et al., 2018) investigated methods for identifying credit card fraud using machine learning algorithms. Recognizing the challenge of maintaining confidentiality when analyzing real-world credit card data, the study employed 12 machine learning algorithms, including standard neural networks and deep learning models. Initial approaches involved traditional algorithms such as Support Vector Machines (SVM), Naive Bayes (NB), and deep learning models. Additionally, hybrid methods were explored, combining techniques such as AdaBoost and majority voting.

(El Bouchti et al., 2017) provided a comprehensive explanation of Deep Reinforcement Learning (DRL) theory and its potential applications in fraud detection and risk management within the banking sector. While the paper primarily focuses on theoretical aspects, it highlights the competitive performance of DRL in fraud detection. This work presents a novel approach to fraud detection that has garnered attention within the research community.

In a study by (Ravisankar et al., 2011), data mining techniques were utilized to forecast financial statement fraud using a dataset comprising 202 Chinese companies, half of which were involved in illegal activities and the other half were legitimate entities. Techniques such as Genetic Programming (GP), Logistic Regression (LR), Multilayer Feed Forward Neural Network (MLFF), Group Method of Data Handling (GMDH), Support Vector Machines (SVM), and Probabilistic Neural Network (PNN) were employed. The authors incorporated a feature selection phase using the simple t-statistic method. Results indicated that PNN exhibited superior performance compared to other methods, demonstrating higher accuracy, while GP achieved comparatively lower accuracy. This underscores the importance of selecting appropriate features to enhance detection rates.

3.1.11. CNN

In this section, our focus turns to one of the most influential algorithms in deep learning, the Convolutional Neural Network (CNN), originally introduced by Yann LeCun and his colleagues in 1998 (LeCun & Bengio, 1995). CNNs represent a subset of neural networks and stand as highly effective models for image classification. They excel particularly in automatically assigning labels to input images, thereby identifying their respective classes.

Typically, a CNN architecture comprises convolutional layers, pooling layers, and fully connected layers, resembling a Multilayer Perceptron (MLP). The architecture of a CNN can be divided into two primary sections: feature extraction (or hidden layer) and classification.

During the feature extraction phase, the network conducts convolution and pooling operations to detect relevant features within the input images. The convolutional layer scans the input images to identify specific sets of features, generating a series of feature maps as output.

$$G(m,n) = (f * h)[m, n] = \sum_j \sum_k h[j, k] f[m-j, n-k] \quad (8)$$

A key divergence between conventional image processing algorithms and CNNs lies in how filters' weights or values are determined. In CNNs, these parameters are learned dynamically through an optimization algorithm during the training phase. This characteristic enables CNNs to adaptively acquire filters tailored to specific tasks. Following convolution operations on input images, the resulting feature maps undergo non-linear transformations facilitated by activation functions like ReLU (Rectified Linear Unit), which replace negative input values with zeros. The interest of these activation layers is to make the model non-linear and therefore more complex. The following formula can calculate the ReLu function:

$$f(x) = \max(0, x) \quad (9)$$

Then, the results will be transferred to the pooling layer that is used, on the one hand, to reduce the number of parameters by minimizing the size of the characteristic maps and, on the other hand, to introduce translational invariance into the model.

Let $A = [a\ i\ j] \in \mathbb{R}^{n*m}$ be a matrix that represents a characteristic map of a specific region, and that will be presented to the pooling layer. The most commonly employed pooling methods in CNNs include:

- Max pooling is where each input region is substituted with its maximum value.
- Average pooling (or weighted average pooling): which computes the average or weighted sum of values within the input region, with weights often determined based on distance from the region center.

The Fully Connected (FC) layer, typically situated after the CNN architecture, establishes connections to every output neuron and is primarily utilized for classification purposes.

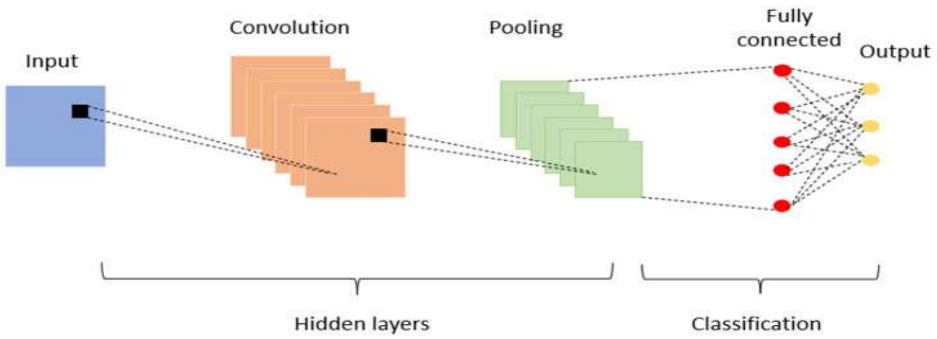


Figure.6 CNN Architecture (Filali et al., 2022)

Furthermore (Fu et al., 2016) introduced a new feature called trading entropy and used a cost-based sampling method to generate synthetic fraudulent samples from real transactions. They used a convolutional neural network (CNN) to tackle the imbalanced data problem and achieved superior performance compared to SVM, NN, and RF.

3.1.12. LSTM

The Long Short-Term Memory (LSTM) technique is an extension of Recurrent Neural Networks (RNNs), which are a type of deep learning neural network designed with memory capabilities. LSTM is particularly well-suited for prediction tasks due to its ability to capture long-term dependencies between input data and prediction outcomes. The architecture of LSTM enables learning over extended sequences, making it effective for identifying longer-term patterns and maintaining prolonged memory. A typical LSTM cell, as illustrated in Figure 6, consists of memory blocks (cells) with multiple states and gates. The cell state serves as the primary conduit for information flow, allowing information to propagate forward without significant alteration. The forget gate (F_t), depicted in Figure 6, regulates which information should be retained or discarded based on a sigmoid function applied to the current input data (X_t). The sigmoid function (σ) determines values that are between 0 and 1; if the value is nearer to 0 it means to forget, and if the value is nearer to 1 means to keep. Also, the cell state vector C_{t-1} controls which elements will be forgotten. The Equation 10 explains how forget gate is calculated: $F_t = \sigma\,(W_f\,[h_{t-1}, X_t] + b_f)$ (10)

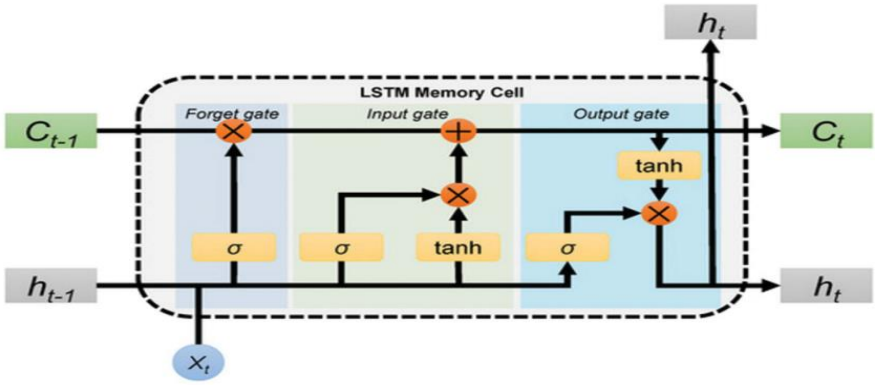


Figure7. LSTM cell structure (Alghofaili et al., 2020)

Where b_f and W_f denote the biases and input weight matrices of the forget gate (Ft). The input gate (I_t) determines which data is necessary to add from the current input (X_t) and also to update the cell state. This gate used the tanh function (N_t) to pass the current input and the hidden state to make values between -1 and 1 to assist regulate the network. Moreover, to generate new

memory, it is then applied to the old cell-stated (C_{t-1}) memory at Equations 9–11. time t-1, which produces new cell-stated (C_t) at time t as in I_t.

$$I_t = \sigma (Wi [h_{t-1}, X_t] + b_i) \quad (11)$$

$$N_t = \tanh (Wf [h_{t-1}, X_t] + b_n) \quad (12)$$

$$C_t = C_{t-1} * f_t + N_t * I_t \quad (13)$$

Where b and W denote the biases and input weight matrices of the input gate (f_t). The output gate (O_t) conditionally determines the output of the next hidden state (h_t) through the output of the sigmoid gate and with new values generated by tanh from the cell state as in Equations 12–13.

$$O_t = \sigma (W_O [h_{t-1}, X_t] + b_O) \quad (14)$$

$$h_t = O_t \tanh (C_t) \quad (15)$$

Where b and W denote the biases and input weight matrices of the output gate (f_t).

3.1.13. Autoencoder

An autoencoder (AE) is a type of unsupervised deep learning network characterized by its symmetric structure with fewer nodes in the middle layers. It consists of an encoder section that compresses inputs into a lower-dimensional representation and a decoder section that reconstructs the input data (Boukerche et al., 2020). According to (Lin et al., 2018), autoencoders replace the input with the input itself (x), making them useful computational tools for learning data representations.

The primary objective of training an AE is to efficiently learn a compact encoding of the data and then reconstruct it. As depicted in Figure 8, the input layer forwards the input data to the hidden layer, where the lower-dimensional encoding is acquired. Subsequently, this encoding is transmitted from the hidden layer to the output layer, where it undergoes decoding and reconstruction to the fullest extent possible.

The number of hidden layers in an AE can vary, with the requirement that each subsequent hidden layer in each part of the network (e.g., the encoder) must have fewer neurons than the previous layer. This architectural design introduces a bottleneck in the network, constraining the amount of information that can pass through and, consequently, promoting a compressed representation of the original input (Kucharski et al., 2020).

The concept behind using autoencoders (AEs) for anomaly detection lies in the premise that anomalies are more difficult to reconstruct by a trained AE compared to normal instances. By employing an AE, a reconstruction error can be calculated for each data sample, serving as an anomaly score (Boukerche et al., 2020) (Darban et al., 2022). Anomalies typically exhibit larger reconstruction errors, enabling their identification and detection.

Recent literature has introduced various types of regularized AEs capable of learning more complex and expressive representations of input features. For instance, sparse autoencoders (SAE) encourage sparsity in layer neurons during training by retaining the top K-most active units (Makhzani & Frey, 2014). Denoising Autoencoders (DAE) are trained to reconstruct a "repaired" input from a corrupted version by initially corrupting the input with stochastic mapping (Vincent et al., 2010). Contractive Autoencoders (CAE) takes this further by learning feature representations resilient to small variations in instances around their neighbors, achieved by incorporating the Frobenius norm of the Jacobian matrix of the encoder's activations as a penalty term (Rifai et al., 2011). Variational Autoencoders (VAE) prevent overfitting by introducing regularization into the representation space by encoding data samples using a prior distribution over the latent space (Pang et al., 2021)(Doersch, 2016). When VAEs are used, they use a reconstruction probability instead of a reconstruction error. This is the chance that a data point can be made from the latent variable using the approximate posterior distribution (Kazemi & Zarrabi, 2017).

Moreover, AEs have been proposed for use in systems aimed at detecting and preventing fraudulent credit card transactions before processing. The authors suggest training an AE to extract relevant features and adding a softmax layer to classify the network's output. The literature investigates the impact of layer size by comparing the results of AEs with different configurations.

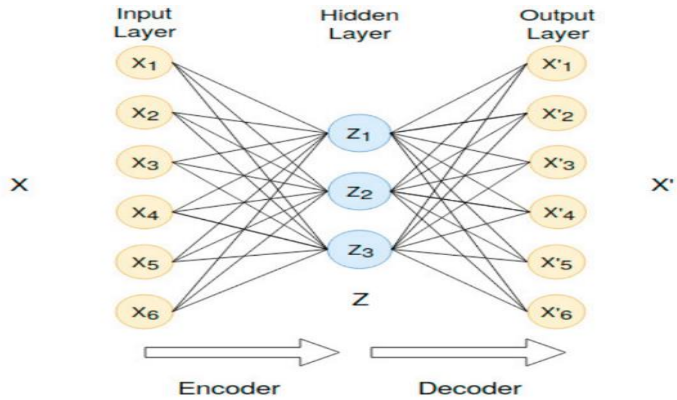


Figure8. The architecture of AE has a single encoding and decoding layer

3.2. Metaheuristic algorithms

Metaheuristic algorithms indeed serve as powerful tools for tackling complex optimization problems that are challenging to solve using conventional techniques. These algorithms draw inspiration from various natural phenomena, including genetics, swarm behavior, and evolutionary processes, to efficiently explore vast search spaces and find optimal or near-optimal solutions.

These techniques have found widespread applicability across diverse fields such as engineering, finance, and computer science, among others, due to their ability to handle intricate problems with numerous variables and constraints. The flexibility and adaptability of metaheuristic algorithms enable them to navigate complex landscapes and search for global optima, which might otherwise be unattainable or computationally expensive using traditional methods.

The work (Hamdi, 2010) exemplifies the significant research and application of metaheuristic algorithms across various domains. Their contributions likely highlight the effectiveness of these algorithms in solving real-world problems, fostering advancements in optimization techniques, and expanding their practical utility in different industries.

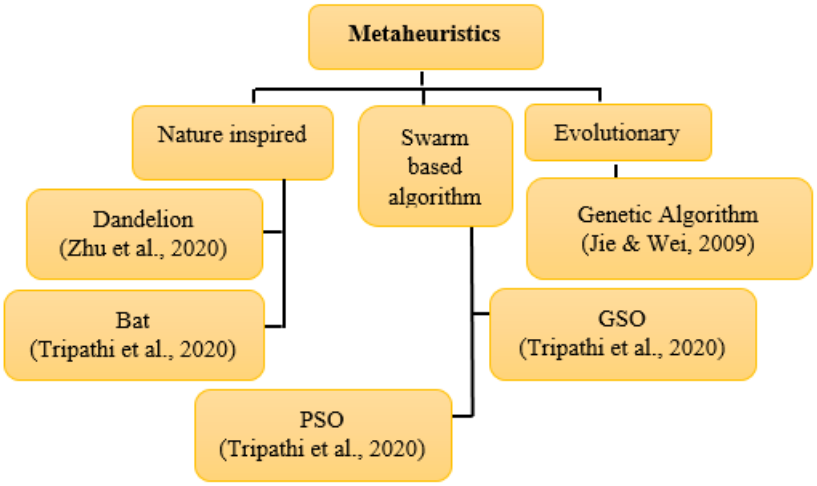


Figure9. Optimization algorithm used for credit card fraud detection

3.2.1. Genetic algorithm (GA)

The Genetic Algorithm (GA) is based on natural evolution. It finds the best solutions by comparing a set of possible solutions, which are usually shown as binary strings called chromosomes (Zojaji & Atani, 2016). (Benchaji & Ouahidi, 2018) proposed utilizing a Genetic algorithm to address the challenge of identifying suspicious transactions on credit cards. The study involved the application of this approach to real-world transactional data, leading to the discovery that all misclassification errors occurred at a similar rate. However, the findings

demonstrated that the proposed solution significantly improved performance, achieving approximately a 200% increase in effectiveness.

(Jie & Wei, 2009) proposed a novel approach for detecting credit card fraud, aiming to address the limitations of traditional classifiers in handling imbalanced datasets, particularly in identifying minority-class objects. Their method combined K-means clustering and Genetic Algorithm (GA), first using K-means for grouping and classifying minority instances, then applying GA within each cluster to generate new instances and create a revised training dataset. The study demonstrated that integrating K-means and genetic algorithms led to improved credit card fraud detection and reduced false alerts.

In a different study, (Jiang et al., 2018) used the Fuzzy Genetic Algorithm Backpropagation Neural Network (FGABPN) to come up with a way to improve the speed and accuracy of finding fraud in financial statements. The GTA Research Service Center provided a dataset for their method. Empirical results indicated that the FGABPN method achieved enhanced accuracy and a high detection rate in identifying fraudulent activities.

3.2.2. Practicle Swarm Optimization PSO

(Seyedhossein & Hashemi, 2010)(Popat & Chaudhary, 2018) established Particle Swarm Optimization (PSO) as a stochastic optimization technique. It mimics the social behavior of organisms, such as birds in a flock or fish in a school, and can be characterized as an autonomously evolving system. PSO operates by initializing a population of particles, akin to birds, randomly across the search space. Each particle adjusts its velocity and seeks the global best position iteratively. At each iteration, particles update their velocity based on their momentum, their individual best position (Pb), and the best position of their neighbors (Pg), ultimately determining their next position.

(Jiang et al., 2018) introduced an advanced learning algorithm called IPSO-ELM, which combines the benefits of PSO and Extreme Learning Machine (ELM). This modified PSO algorithm considers both the output weight norm and the Root Mean Square Error (RMSE) on the validation set when determining input weights. A comparison with the E-ELM, ELM, and LM algorithms showed that the proposed IPSO-ELM algorithm was better at generalization.

3.2.3. Bat Algorithm

The Bat Algorithm (BA) has demonstrated success in solving various challenging optimization problems, surpassing Genetic and Particle Swarm Algorithms in terms of both exploitation and exploration capabilities (Mirjalili et al., 2014)(X. S. Yang, 2010). Inspired by the echolocation ability of microbats, X. S. Yang introduced the Bat Algorithm, which mimics the behavior of bats in search of food or prey using echolocation. The algorithm utilizes idealized rules based on the echolocation characteristics of bats for weight and bias optimization.

(Tripathi et al., 2020) suggested an evolutionary way to fix problems with Extreme Learning Machines (ELM) by using the Bat Algorithm (BA) to make the best weights and biases for a model that evaluates credit risk. Additionally, they introduced a novel activation function to compute the output of hidden neurons. The proposed approach was also tested on various ELM variations to demonstrate its stability.

3.2.4. Dandelion Algorithm

According to (H. Zhu et al., 2019), the Dandelion Algorithm (DA) is a novel swarm intelligence algorithm that draws inspiration from the behavior of dandelion dispersal. DA offers several advantages, including its straightforward calculation process and easy comprehensibility. In DA, the seeding radius of the dandelion is dynamically adjusted, and the algorithm possesses self-learning capabilities to improve the direction of seed dispersion. Due to its effectiveness and simplicity, DA has found successful applications in optimizing Extreme Learning Machines (ELM) for biomedical classification tasks, garnering attention from researchers.

(Gong et al., 2018) applied DA to 12 common functions and compared its performance with that of other algorithms, demonstrating superior results across the test functions. This validated DA's simulation performance. Furthermore, DA has shown promising outcomes in conjunction with ELM across various domains (H. Zhu et al., 2020) (Abualigah, 2021) proposed an enhanced version of the Dandelion Algorithm called Dandelion Algorithm with Probability-Based Mutation (DAPM), which incorporates Levy and Gaussian mutations based on probability. The DAPM algorithm has three probability models: the linear (DAPM L), the binomial (DAPM B), and the exponential (DAPM E). It works better than the basic DA algorithm and other intelligent optimization algorithms. These probability models are applied to optimize Weighted Extreme Learning Machine (WELM) for imbalanced classification

problems, yielding enhanced classification performance. Additionally, the optimized WELMs are employed in credit card fraud detection tasks.

3.2.5. Group Search Optimization

The Group Search Optimization (GSO) algorithm is a population-based optimization technique that utilizes the Particle Swarm (PS) model as its framework. Within this framework, the behaviors and strategies observed in animal resource searching are metaphorically adopted to design effective search strategies. For instance, the general scanning mechanisms found in animals, such as vision, are integrated into the algorithm's producers. Additionally, the scrounging strategies described by (Abualigah, 2021) are incorporated into the GSO algorithm. Furthermore, to prevent entrapment in local minima, some group members disperse from the main group to perform random walks. The GSO algorithm uses two main methods, PSOM (Particle Swarm Optimization Mechanism) and LSOM (Local Search Optimization Mechanism), to create a smart search method that can solve a wide range of optimization problems. Compared to alternative algorithms, GSO is known for its simplicity, adaptability, and ease of implementation, making it suitable for addressing a wide range of optimization challenges.

4. Challenges in Fraud Detection

Utilizing artificial intelligence for fraud detection is a well-established practice, with many enterprises globally implementing machine learning models for this purpose. Typically, modern fraud detection methods involving machine learning require the involvement of domain experts, who have two primary responsibilities: gathering historical transaction data and assisting in the feature generation process for either classic or advanced machine learning models.

In addition, metaheuristic algorithms can be applied to enhance the training process of ML models. For example, genetic algorithms can be used to evolve neural network architectures or optimize the structure of decision trees. Similarly, particle swarm optimization can be employed to adaptively adjust the weights and biases of ML models during training.

Overall, using both machine learning and metaheuristic algorithms together is a powerful way to solve difficult optimization and prediction problems in many areas, such as feature selection, hyperparameter optimization, and model training. By leveraging the strengths of both approaches, researchers and practitioners can develop more robust and efficient ML solutions.

These models are then built based on the features generated from historical data to detect or prevent fraud. However, despite its utility, this approach is not without its challenges, which complicate the fraud detection process:

Class Imbalance: In many fraud detection scenarios, the number of legitimate transactions significantly outweighs the number of fraudulent ones, leading to class imbalance. This imbalance can affect the model's ability to correctly identify fraudulent transactions, as the majority class may dominate the learning process.

Data Quality: The quality and accuracy of data used for fraud detection are paramount. Incomplete or inaccurate data can lead to incorrect assessments of transactions, making it challenging to distinguish between genuine and fraudulent activities.

Data Volume and Velocity: The exponential growth of digital transactions makes it difficult to process and analyze large amounts of data in real time. Traditional systems could find it difficult to handle the enormous quantity and frequency of data produced.

Evolution of Fraud Techniques: Fraudsters are constantly evolving their tactics to circumvent detection systems. They may employ sophisticated techniques, such as identity theft, account takeover, and social engineering, making it difficult for detection systems to keep up.

False Positives: Reducing false positives is a critical challenge. False positives occur when legitimate transactions are incorrectly flagged as fraudulent, causing inconvenience to customers and increasing operational costs for businesses.

Feature Engineering: Selecting and engineering the right features from data is crucial for accurate fraud detection. Deciding which features are relevant and how to transform them can be challenging, especially in complex, high-dimensional datasets.

Model Interpretability: Some machine learning models, especially deep learning models, are often regarded as black boxes. Understanding how these models arrive at their decisions can be a challenge, particularly when providing explanations to customers or regulators is necessary.

Real-Time Detection: In industries like banking and e-commerce, real-time fraud detection is essential. Delayed detection can result in significant financial losses. Building models that can operate in real time while maintaining accuracy is challenging.

Data Privacy and Compliance: Adhering to data privacy regulations, such as GDPR or HIPAA, while collecting and storing customer data for fraud detection can be complex. Striking the right balance between privacy and fraud prevention is an ongoing challenge.

Adversarial Attacks: Sophisticated fraudsters may employ adversarial attacks to manipulate or deceive fraud detection models. This requires constant vigilance to detect and mitigate such attacks.

Scale and Integration: For large organizations with multiple systems and databases, integrating fraud detection solutions into existing infrastructures can be a challenge. Ensuring that the solution scales with the organization's growth is crucial.

Cost-Effectiveness: Building and maintaining fraud detection systems can be costly. Balancing the cost of prevention with the potential financial losses due to fraud is a continuous challenge.

Human Error: Human factors can also introduce vulnerabilities. Employees may inadvertently facilitate fraudulent activities, and insider threats are an ongoing concern.

Emerging Technologies: The introduction of new technologies, such as blockchain, cryptocurrencies, and AI, creates new avenues for fraud. Detecting and preventing fraud in these evolving technological landscapes is an ongoing challenge.

One of these techniques is employing data mining techniques that explore the underlying patterns in a big dataset and further differentiate between normal and suspicious transactions(Awoyemi et al., 2017a). This section reviews the latest fraud detection techniques in financial domains until 2022.

In recent years, deep learning models like MLPs have been gaining increased research attention for this task, and research efforts have even explored text mining and natural language processing (Z. Li et al., 2021).

5. Practical implications

The study contributions have a wide range of practical implications, the main one including the enhanced protection of financial institutions against fraudulent activity. The significant advantages that can be obtained from the suggested models and approaches are highlighted by the following important practical implications:

5.1. Reducing financial losses

Reducing financial losses is a top priority regarding credit card fraud detection. The work performed especially by using deep neural networks, managing imbalanced data, and utilizing optimized AEs, is a major step in the right direction. Using deep neural networks is a promising way to improve fraud detection accuracy and lessen financial losses by more precisely identifying fraudulent activity. In the same way, the particular techniques for rectifying unbalanced data in this study present enhanced detection and reduce losses associated with fraudulent transactions that pass undetected. Additionally, the novel approach of using AEs has offered a promising way to improve the accuracy of fraud detection and maybe lower the financial losses related to those activities.

5.2. Enhanced transaction security

Our contributions bring about substantial improvements in transaction security within the context of credit card fraud detection. Here are some robust enhancements our research provides:

- **Enhanced Accuracy in Fraud Detection:** Utilizing deep neural networks significantly improves the accuracy of detecting fraudulent activities. This enhancement ensures a more precise identification of suspicious patterns, reducing the chances of fraudulent transactions going unnoticed and thereby strengthening transaction security.
- **Improved Handling of Imbalanced Data:** Our methodologies specifically designed to handle imbalanced data lead to a more refined fraud detection process. By addressing data discrepancies, this approach significantly minimizes false negatives and enhances the overall security of transactions.
- **Innovative Optimization Techniques:** The innovative use of optimized AEs alongside specific algorithms showcases remarkable improvements in fraud detection precision. This development ensures a more secure environment for financial transactions by significantly reducing false positives.
- **Comprehensive Security Measures:** The combined effect of these methodologies offers a comprehensive set of tools to fortify transaction security. By significantly reducing false positives and negatives, our contributions provide robust measures to combat fraudulent activities, ensuring a more secure financial ecosystem.

These strong improvements resulting from our research work lay a solid foundation for implementing more secure transaction systems, specifically in the detection and prevention of credit card fraud.

5.3. Operational optimization

Our research significantly optimizes operational processes within credit card fraud detection, offering clear advantages:

- **Enhanced Resource Allocation:** Utilizing advanced methods like deep neural networks allows for more effective resource allocation. This optimization ensures better utilization of computational resources, improving the overall efficiency of fraud detection operations.
- **Improved Accuracy, Reduced Errors:** Our methods for handling imbalanced data greatly minimize false positives and negatives. This optimization leads to more accurate

identification of fraudulent transactions, reducing unnecessary investigations and errors in the fraud detection process.

- **Swift Response to Threats:** Our research has improved the accuracy of spotting fraudulent transactions, allowing for quicker responses to potential threats. This operational optimization accelerates the investigation process, reducing the time and resources needed to tackle fraudulent activities.
- **Adaptable and Scalable Solutions:** Our methodologies demonstrate adaptability across various operational scales. This adaptability ensures that our approaches can be implemented effectively irrespective of transaction volumes, optimizing operational efficiency.
- **Cost-Efficiency and Streamlined Processes:** Overall, our research streamlines processes, leading to cost savings for financial institutions. It minimizes unnecessary operational costs tied to false alarms and inefficient detection methods.

5.4. Protecting personal data

Protecting personal data is crucial to ensuring the confidentiality and security of sensitive user information. Here are some key measures to safeguard personal data:

- **Use Encryption:** Convert sensitive information into a secret code that's hard to read without a special key.
- **Two-Factor Authentication (2FA):** Add an extra layer of security by requiring two ways to verify your identity when logging in, like a password plus a code sent to your phone.
- **Keep Everything Updated:** Regularly update your apps, software, and devices to fix security problems.
- **Be Clear about Data Use:** Tell users how you collect and use their information, and ask for their permission to use it.
- **Control Access:** Only let authorized people access sensitive data and use strong passwords.
- **Secure Physical Devices:** Keep servers and devices storing data in safe places, and back up data often to prevent loss.
- **Train Employees:** Teach staff about keeping data safe to avoid mistakes that could lead to problems.

- **Check for Weaknesses:** Regularly check for potential problems in security and fix them quickly.

5.5. Adaptability to new threats

"Adaptability to new threats" refers to the ability of systems or organizations to adjust and respond effectively to emerging or evolving threats to security. This flexibility is essential to preserving strong security protocols against novel and unanticipated threats. Here's how it can be achieved:

- **Keep Watching for New Threats:** Regularly check for any new or changing security risks, like cyber threats or fraud.
- **Change Security Rules Fast:** Have flexible security rules that can be updated quickly when new risks show up.
- **Use Modern Security Tools:** Use the latest security tools that can handle new types of risks, like smart systems that can spot unusual behavior.
- **Teach Employees About New Risks:** Make sure your team knows about new security risks and what to do about them. Informed people can help keep things safe.
- **Work Together and Share Information:** Team up with other experts and groups to learn about new risks and how to deal with them. Sharing knowledge can help everyone stay safe.
- **Check Security Regularly:** Keep checking for any weak spots in security so you can fix them before they become big problems.
- **Have a Plan for Emergencies:** Make a plan that explains what to do if there's a security problem. Everyone should know their part in keeping things safe.
- **Follow Security Rules and Laws:** Make sure you're following all the rules and laws about security in your industry. They often help keep up with new risks.

6. Conclusion

In this chapter, we navigated through various techniques, from data mining to sophisticated approaches in deep learning and metaheuristics. This comprehensive review highlighted notable advancements while shedding light on the persistent challenge of data imbalance.

It is precisely to address this gap that our focus shifts to the next chapter. In response to the identified challenges, we present three innovative models in Chapter 2, "Imbalance Resolution

for Credit Card Fraud Detection." These models aim to overcome the inherent imbalance in datasets, providing a promising perspective to enhance fraud detection.

The first model confronts the issue of imbalanced datasets, the model introduces the Synthetic Minority Oversampling Technique (SMOTE) to ensure a more balanced representation of fraudulent transactions. With a core objective of detecting credit card fraud, Model 1 employs two potent artificial neural network algorithms, Multilayer Perceptron (MLP) and Extreme Learning Machine (ELM). The second model complements this perspective by leveraging an innovative approach that unifies two architectures, the AE and the MLP, to create a cohesive Fraud Detection Neural Network. By harnessing the AE's capabilities to extract relevant features from input transactions, the model strategically reduces data complexity while preserving crucial information. This dimensionality reduction strategy plays a pivotal role in significantly enhancing the FCDN's performance in detecting fraudulent activities. Lastly, the third model introduces an innovative solution based on a three-phase strategy. Trained on fraudulent transactions, an AE captures intricate patterns, while another reduces dimensionality using encoded representations. The Fraud Detection Neural Network (FCDN) utilizes these encoded features for enhanced classification. Augmenting performance, an additional FCDN addresses imbalances via the Synthetic Minority Oversampling Technique (SMOTE). Notably, predictions from AEs, AE-FCDN, and FCDN converge through a majority voting method, resulting in a robust and accurate credit card fraud detection system.

Chapter II: Imbalance Resolution for Credit Card Fraud Detection

1. Introduction

With the growth of e-commerce and computational devices, the use of credit cards has also been scaled up rapidly for online shopping and other transactions. The growth of credit card transactions has posed challenges in detecting fraudulent transactions before banking and financial organizations (Naik & Kanikar, 2019). Fraudulent transactions can be done using either a stolen card from external or internal sources or a credit card obtained by giving false information. A customer who is bankrupt and might not be able to pay can also do it (Makki et al., 2019). Several researchers have extensively investigated the detection of such fraudulent credit card activities (Awoyemi et al., 2017b); (Dornadula & Geetha, 2019); (Makki et al., 2019); (Naik & Kanikar, 2019); (Sailusha et al., 2020). Most of the proposed techniques have used machine learning to identify whether a transaction is fraudulent or legitimate. In general, supervised machine learning algorithms have been used to detect fraudulent transactions. One of the principal challenges faced in the process of detecting fraudulent transactions is the dataset. In most cases, the dataset used to develop such machine learning models has an imbalanced class distribution. The probability of fraudulent credit card transactions is extremely small compared to valid ones. It leads to the imbalanced proportions of labels in the dataset. One class, the majority class, has a maximum number of instances compared to another class, the minority class. This leads to a learning model biased towards the majority class because the model treats the minority class as noise or outliers (M. Galar et al., 2012). Assume a dataset has two classes; the first is the negative class that represents the legitimate transactions, and the second is the positive class that represents the fraudulent transactions. The dataset has an imbalanced ratio of 1:1000. In this dataset, the number of instances in the negative class is 1000, and the number of instances in the positive class is only one. In this case, a classifier needs to ignore 999 negative class instances to obtain the maximum accuracy. The accuracy of the prediction for the majority class will be much higher than the accuracy of the prediction for the minority class (Arpit Singh & Purohit, 2015). Nevertheless, the great interest in developing a detection system is to identify fraudulent transactions. In other words, we would like the model to predict fraudulent transactions with high accuracy. Suppose a legitimate transaction is predicted as a fraudulent transaction. In this case, the inquiries can be performed by the card issuing organization. The transition will be undetected and the goal of developing the model will not be achieved if the classification model predicts a fraudulent transaction as a valid

transaction. To create an intelligent model or classification model that can detect fraudulent transactions with high accuracy, the imbalanced distribution of data should be reduced. In other words, the dataset used to build and evaluate the classification model should have a balanced class distribution. There are several ways the imbalanced class problem can be handled. In the past, several methods have been proposed to tackle the class imbalance problem that occurred in several domains (Abd Elrahman & Abraham, 2013a); (Bejjanki et al., 2020);(M. Galar et al., 2012); (Tian & Liu, 2020). The approaches can be categorized into these four categories (Abd Elrahman & Abraham, 2013b); (Fernández et al., 2018);(D. Galar et al., 2011): data-level, algorithm-level, cost-sensitive, and classifiers ensembling approach. The data-level approach, also known as the external approach, is used during the preprocessing stage of the dataset (Lu et al., 2011). In the data-level approach, undersampling and oversampling techniques are two of the approaches that have been used in several domains to handle class imbalance problems (P. Yang et al., 2014). In the algorithm-level methods, also known as an internal approach, the problem can be taken care of by modifying the classification algorithm to enhance the magnitude of positive class samples (Quinlan, 1991); (Wu & Chang, 2005); (Zadrozny & Elkan, 2001). In the cost-sensitive approach, instead of each instance being either correctly or incorrectly classified, each instance is given a misclassification cost. Thus, instead of trying to optimize the accuracy, the problem is to minimize the total misclassification cost (Alejo et al., 2013) ;(Thach et al., 2008);(C. Y. Yang et al., 2009). The classifier's ensemble approach combines multiple classifiers into one classifier to handle the class imbalance problem (Guo & Viktor, 2004)(Khalilia et al., 2011);(Yun & Bing, 2008).

In this chapter, we leverage insights derived from our conference titled 'Credit Card Fraud Detection Based on Multilayer Perceptron and Extreme Learning Machine Architectures' alongside two seminal research articles. The first contribution, 'Deep Neural Network for Detection of Fraudulent Transactions, delves into advanced neural network methodologies tailored for fraud detection. The second article, 'Credit Card Fraud Detection: Addressing Imbalanced Datasets with a Multi-Phase Approach,' offers a comprehensive strategy to mitigate imbalanced data challenges in fraud detection through a multi-phase methodology. By integrating and synthesizing the methodologies outlined in these academic contributions, this chapter endeavors to offer an in-depth analysis and comparison of effective techniques curated from these sources. Our objective is to present a comprehensive and nuanced exploration of methodologies specifically designed to address the complexities of handling imbalanced datasets within the realm of credit card fraud detection.

2. Traditional Imbalanced Data Techniques

To handle the class imbalance problems that occur in different domains, many approaches have been proposed for the detection of credit-card fraudulent transactions.

2.1.Oversampling

Oversampling, also known as upsampling (Amit Singh et al., 2022), is a sampling technique, which balances the dataset by replicating the minority class samples. The oversampling method creates the superset without losing the original data, but it may lead to an overfitting problem and more complex computational computing when the size of the data is large enough. Oversampling methods are classified into two types: Random oversampling and informative oversampling (Ramyachitra & Manikandan, 2014). The random oversampling method is the process of replicating the minority class samples for balancing the dataset by selecting them randomly. However, the informative sampling method is used to generate synthetic data points of the minority class in defined criteria for balancing the dataset. There are several oversampling approaches have been proposed in the past to balance the class distribution in the imbalanced dataset.

2.2. Synthetic minority oversampling technique (Smote)

Smote proposed by (Nitesh V. Chawla et al., 2002), is a very popular class imbalance oversampling technique. In this technique, instead of creating simple replications of the minority class, it uses synthetic data points. These synthetic data points are examples of minority class data points. These synthetic data points are generated by interpolating new data points between several available minority classes and positive instances that exist together. This interpolation of the synthetic data points is generated using minority class data instances and the K-Nearest Neighbours (KNN) method (Amit Singh et al., 2022). Based on the requirement of synthetic data points, the KNN algorithm selects minority class instances interpolates between all K data points, and plots new data points randomly. According to the requirement of the number of synthetic data points, these points are added to the original data and train the machine-learning model. The Smote technique works efficiently when the size of the dataset is small. In the case of large enough datasets, it will take more time to create synthetic data points,

and it will not perform efficiently. The new synthetic data point, A, created using the Smote method is defined as:

$$A = X + c(XR–X) \quad (16)$$

Where X is a data point in the dataset, XR is the randomly selected data point among the K nearest neighbours of X, and c ($0 \leq c \leq 1$) is a constant.

2.3. Synthetic sampling (AdaSyn)

AdaSyn, proposed by (He et al., 2008), is an oversampling approach like Smote. In the Smote over-sampling technique, there is a limitation to creating synthetic data points for minority classes that it can increase the possibility of overlapping data points. To avoid this limitation, AdaSyn, an adaptive oversampling approach, was proposed. The AdaSyn generates data samples for those minority class samples that are complicated to learn. The generation of synthetic data samples, based on the minority class density distribution, improves the decision boundary of the original data. The synthetic data points generated by AdaSyn not only improve the dataset by balancing it (Amit Singh et al., 2022), but they also reduce the learning bias of the original dataset. Let a dataset with a class imbalance problem consist of minority class samples, Smin, and majority class samples, Smax. The number of synthetic minority class samples, SN, generated using the AdaSyn is defined as:

$$|S_N| = (S_{max} - S_{min}) * b \quad (17)$$

where $b \in [0, 1]$ is the balance factor. In the case of total balance between two classes, choose b equal to 1.

2.4. Random oversampling (ROS)

Random oversampling (Ling & Li, 1998) is an effective and widely used oversampling approach to handle the class imbalance problem. It selects minority class samples randomly and replicates them. Then add this new subset to the training data for the machine learning models. The random oversampling method creates the same new data points from the original dataset that can increase the possibility of overfitting the model (Amit Singh et al., 2022). Let us assume a dataset S has a class imbalance problem consisting of minority class samples,

Smin, and majority class samples, Smax. Let A be the randomly generated replicated data points of minority class samples then the number of samples in the new dataset is given as

$$|S| = Smax + (Smin + |A|) \quad (18)$$

2.5. Undersampling

Undersampling is a sampling technique that removes instances from the majority class for balancing the dataset, and a new subset of original data is used for training the learning models (Fernández et al., 2018). In the undersampling technique, data will be eliminated from the majority class until the dataset gets balanced. The undersampling technique is further divided into two methods: Random undersampling and Informative undersampling (Ramyachitra & Manikandan, 2014)(Amit Singh et al., 2022)s. The Random undersampling method randomly selects a subset from the original dataset and starts to eliminate it until the dataset gets to balance. The Informative undersampling method selects a subset from the original dataset based on predefined criteria, and the subset will be eliminated based on these criteria. Some basic undersampling methods are described in the coming sections.

2.6. Random undersampling (RUS)

Random undersampling is the most common and effective undersampling technique used to balance the class distribution of the imbalanced dataset (Fernández et al., 2018). The RUS is faster than other techniques but has the major drawback of losing valuable information. There is a chance that we might lose some instances from the dataset that can affect the accuracy of the model. Let us assume a dataset S has a class imbalance problem. In the dataset, let S_{min} be the number of examples of the minority class samples and S_{max} be the number of instances of the majority class of samples.

3. Adopting Architectural and Technical models

3.1. Model 1: SMOTE-enhanced Fraud Detection Model

A multilayer perceptron is a deep learning technique; it is a feed-forward network connecting at least three layers of nodes, an input layer, a hidden layer, and an output layer in an oriented graph. The input layer contains the vector features of the Credit card transaction. The first hidden layer receives the weighted inputs from the input layer and sends data from the previous layer to the next one. Finally, the output layer contains the classification result. Where the output '0' indicates a normal transaction and, '1' indicates a fraud transaction. MLP is used for supervised learning problems, which need a set of training data and desired outputs to adjust the weights and biases of the model to minimize mean squared error using the backpropagation algorithm.

The steps of the backpropagation algorithm are: 1-Initialize the weighted inputs wij randomly 2-For all the training dataset: Give the input to the model to get the output;

$$y_i = f\left(\sum_j w_{ij}\ x_j\right) \quad (19)$$

Calculate the error by comparing the output

$$\delta_k = f'\left(\sum_k w_{kj}\ x_j\right)(y_{dk} - y_k) \quad (20)$$

result with the desired result;

Propagate the error from layer to layer in a backward direction;

$$\delta_i = f'\left(\sum_j w_{ij}\ x_j\right)\sum_k w_{ki}\ \delta_k \quad (21)$$

Update all network weights;

$$w_{ij}(t+1) = w_{ij}(t) + \lambda\frac{\partial E}{\partial wij} \quad (22)$$

λ is the learning rate.

$$\frac{\partial E}{\partial wij} = \delta_i\, x_j \quad (23)$$

Extreme Learning Machine is a machine learning method developed by (Guang Bin Huang et al., 2004) It is a single feedforward neural network. Its structure consists of a single hidden layer feedforward artificial neural network (SLFN), where the weights between inputs and hidden nodes and hidden layer biases are chosen arbitrarily and it remains constant during training and predicting phases. The weights that connect hidden nodes to outputs are determined analytically. ELM uses activation functions such as sigmoid, hard limit, Multiquadric, and Gaussian in the hidden layer, and linear functions in the output layer (Gao Huang et al., 2019). ELM algorithm can be generalized in 3 steps as shown below (Ding et al., 2014). Given a training set [Xi,Ti] Xi $\in R^p$; $T_i \in R^c$; i $\in\{1, \dots, N\}$, p input features, and q Hidden node number, c output node. F is the activation function. (1) Phase: W(q,p) input weights and bias values b $\in R^q$ are chosen randomly; (2) Phase: H(q,n) hidden layer output is calculated;

$$H(W_{(q,p)}, b, X_{(p,n)}) = \begin{bmatrix} F(w1j * x1j + b1 & \cdots & F(w1p * x1n + b1 \\ \vdots & \ddots & \vdots \\ F(wqj * xqj + bq & \cdots & F(wqp * xpn + bq \end{bmatrix} \quad (24)$$

(3) Phase: output weights are calculated with: Hβ= T (25)

β can computed via the Moore–Penrose pseudo-inverse method as follows:

$$\beta = (H^T H)^{-1} H^T T \quad (26)$$

The class label for a new input is defined as below:

$$Y = \text{argmax}\{o_i\},\ i = 1,\ldots,c \text{ Where } o_i = m_i^T h(t) \quad (27)$$

3.2. Model 2: Autoencoder-based Fraud Detection Model

This contribution introduces a pioneering approach that amalgamates two architectures: the AE and the MLP, forming a cohesive FCDN. Leveraging the AE to extract pertinent features from input transactions serves to reduce data complexity while retaining essential information. This dimensionality reduction strategy significantly augments the performance of the FCDN in detecting fraud. Our experiments were conducted on two imbalanced credit datasets and compared against recent studies.

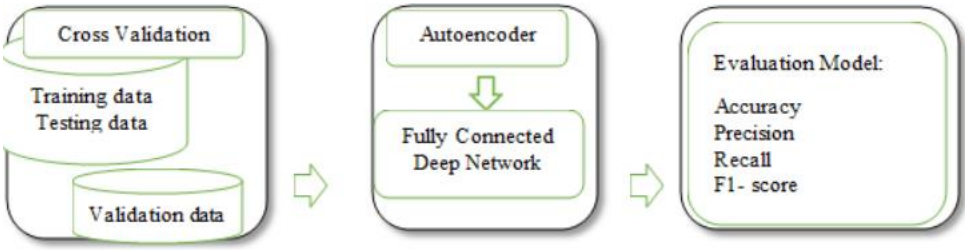


Figure12. Overall steps for detecting Fraudulent transactions from an imbalanced dataset

Data pre-processing plays a critical role in preparing data for machine learning tasks, and one commonly used technique is normalization, specifically min-max scaling. Normalization aims to standardize the range of values across all features to prevent the model from giving undue importance to features with larger values.

The dataset is divided into K subsets or folds. Each fold contains an equal number of samples. In our case, K is set to 5. During each iteration, one of the K folds is used as the testing set, and the rest of the K-1 folds are used as the training set. The model is trained on the training set and then evaluated on the testing set. This process is repeated K times, with each fold serving as the testing set once. The model's accuracy is calculated for each iteration, and the overall accuracy is obtained by taking the mean of these scores. To further validate the model, a separate validation dataset, not used during the K-Fold Cross Validation, is employed to measure its accuracy on unseen data. K-Fold Cross Validation is preferred over a single train-test split as it provides a more reliable estimate of the model's performance and ensures better generalization to new data. The validation accuracy obtained on the separate validation dataset is a more

realistic measure of how well our model will perform on new, unseen data. The mean accuracy calculated during K-Fold Cross Validation can be considered an estimate of the model's general performance.

Additionally, AE can be utilized to address the issue of imbalanced datasets by transforming the input data into a lower-dimensional latent space representation. This process enables the AE to compress the information from the original dataset into a more balanced representation, effectively capturing the underlying patterns and distributions of both the majority and minority classes. By doing so, the performance of subsequent classification models can be improved as the impact of class imbalance is reduced, leading to more accurate and robust predictions. Furthermore, incorporating AE as a pre-processing step can also reduce the computational complexity and training time required for deep learning models by reducing the dimensionality of the input data while retaining crucial features and information. In the training phase, the AE learns how to shrink the input data into a latent representation with few dimensions while minimizing the reconstruction error. It does this by getting the code (B) from the hidden layer (Figure 1). The code (B) consists of fewer neurons than the input layer, resulting in a representation of the input features with fewer dimensions.

Following the AE training, the famous Fully Connected Neural Network (FCDN) algorithm, plays a crucial role in learning from the training data.

Several tests are conducted to obtain the optimal parameters for the model, such as the ideal number of hidden layers, the number of neurons per layer, and the optimization function. The model's configuration is refined through this iterative process to achieve optimal performance on the given task. Finally, in the test phase, FCDN declares whether or not each testing input vector is fraudulent.

3.3. Model 3: Multi-phase Preprocessed Fraud Detection Model

There are three steps in the process: using fraudulent transactions to train an AE, using another AE to reduce the number of dimensions, and then using the encoded representations as input for FCDN classification. To further enhance the model's performance, we introduce an additional FCDN trained on the preprocessed data using the Synthetic Minority Oversampling Technique (SMOTE). The predictions from both AE, AE-FCDN, and the FCDN are combined using a majority voting approach.

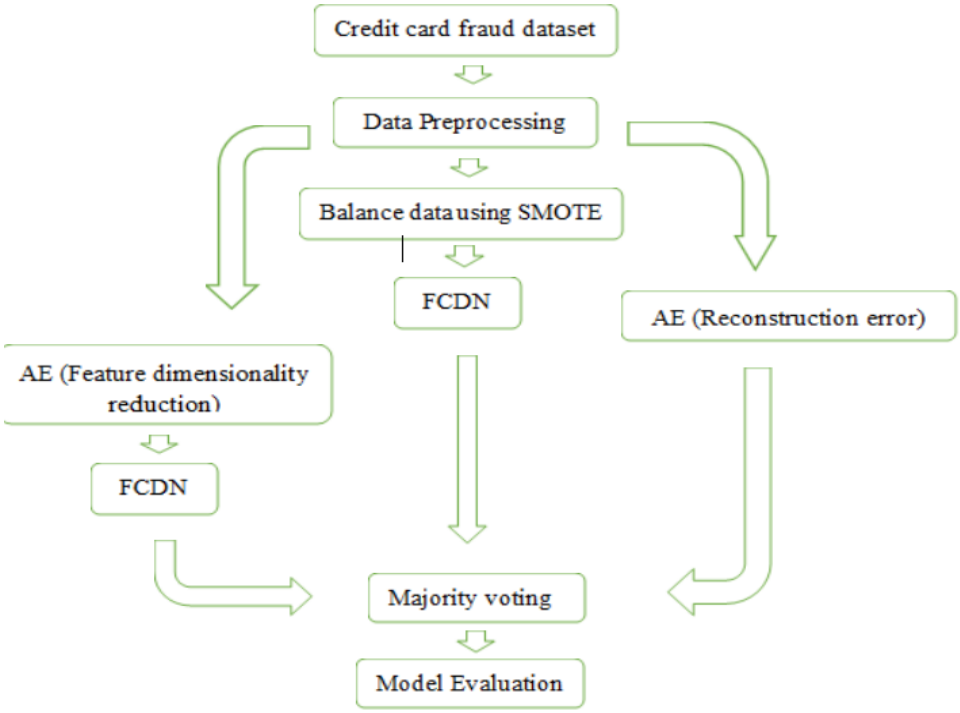


Figure13. Overall steps for detecting Fraudulent transactions from the imbalanced dataset

The overall process described in our architecture is depicted in Figure 9 and can be summarized as follows:

3. Data Pre-processing: Normalize the data to bring all features to a similar scale using techniques like min-max scaling. The dataset is split into two subsets: one for cross-validation during model training and hyperparameter tuning, and the other for final testing.
4. Reconstruction Error Model: Train an AE model using only fraudulent transactions as input data. The AE learns to reconstruct the input transactions and calculates a reconstruction error for each transaction. In the testing phase, transactions with lower reconstruction errors are more likely to be fraudulent since they align well with the patterns learned by the model.
5. Dimensionality Reduction Model: Train another AE model on the (k-1) fold dataset to reduce its dimensionality. The encoded representations obtained from the AE are then fed into FCDN for classification.
6. Data balance: The preprocessed data, including both fraudulent and non-fraudulent transactions, is subjected to SMOTE. The preprocessed and balanced data is then used to train another FCDN. This additional FCDN enhances the model's ability to accurately identify and classify fraudulent transactions.

7. Prediction and Combining: The outputs from all three models (the AE models and the FCDN) are combined using a majority voting approach. This ensemble technique selects the most frequent prediction among the models, resulting in a more accurate final prediction. The combination of these models and the majority voting scheme improves the overall accuracy of fraud detection.
8. After making predictions using the model, it is crucial to evaluate its performance to understand its effectiveness and identify its strengths and weaknesses.

The provided pseudocode outlines the steps involved in the proposed model. It covers data pre-processing, training the reconstruction error model, training the dimensionality reduction model, training the third model on balanced data using Smote, and the prediction and combining phase.

Pseudo code of the proposed model

```
Input:  X_train, X_test, y_train, y_test
#Train Model 1 on the training set using only fraudulent transactions
  X_train_fraud = select_fraudulent_transactions (X_train, y_train);
  model_1 = train_autoencoder(X_train_fraud);
# Predict Model 1 on the testing set
  y_pred1 = predict_model1(model_1, X_ test);
#Train Model 2 on the training set using all transactions
  X_Autenco = Encode(X_train) ;
  model_2 = train_autoencoder_FCDN(X_Autenco, y_train);
#Use Model 2 to predict the class of transactions in the testing set
  y_pred2 = predict_model2 (model_2, X_test);
#Train Model 3 on the training set using all transactions
  model_3 = train_FCDN (X_train_SMOTE, y_train_SMOTE);
#Use Model 3 to predict the class of transactions in the testing set
  y_pred3 = predict_model3(model_3, X_test);
#Combine the predictions from three models using majority voting
Output:  y_pred_majority = majority_voting(y_pred1, y_pred2, y_pred3);
```

Incorporating a multi-stage architecture in our approach empowers our fraud detection model with advanced capabilities. However, this complexity necessitates a comprehensive understanding of each component's role and collective impact on model performance. Implementing such a complex architecture introduces several challenges, including selecting and fine-tuning hyperparameters, substantial training times, and allocating computational resources. Maintenance of this intricate model structure in the face of evolving financial landscapes and new data requires careful consideration. To address these challenges, we have adopted strategies to streamline implementation and maintenance, organize our codebase modularly for manageability and allocate computational resources efficiently. While

acknowledging the complexity introduced by our model, it is crucial to recognize that these intricacies are balanced against the substantial gains in fraud detection accuracy, driven by our pursuit of optimal security measures and considered trade-offs to achieve superior performance.

4. Preprocessing Data Models for Enhanced Predictive Analytics

4.1. Datasets and Features

In this study, three distinct datasets were employed, each offering valuable insights into credit card transactions and fraud detection. The primary dataset, known as the Credit Card Fraud Dataset (CCFD), comprises transactions made by European cardholders in September 2013. This dataset features 30 numerical input features, with 'Time' and 'Amount' remaining unaltered by Principal Component Analysis (PCA) transformation. The 'V' features have undergone PCA for dimensionality reduction, while confidentiality constraints limit the disclosure of original features and comprehensive background information. Remarkably, the absence of 'Null' values streamlines data preprocessing, eliminating the need for value replacement techniques. The PaySim simulator produced the second dataset, which is a synthetic dataset with 8 numerical attributes. This dataset includes transaction types such as cash-in, cash-out, debit, payment, and transfer, coupled with transaction-specific details like customer identifiers, initial and new balance information, and a binary 'isFraud' class indicating fraudulent transactions. Additionally, the study incorporates the German credit card dataset sourced from the UCI Machine Learning Repository. This dataset provides critical numerical data and a binary class, further enriching the study's comprehensive analysis of fraud detection mechanisms across multiple datasets.

Table.1 Information about the datasets

Dataset	Total Transaction	Normal Transaction	Fraudulent Transaction	Number of features	Ratio
CCFD	284 807	284 315	492	30	0.17%
Synthetic Financial	6 362 620	6 354 407	8213	10	0.13%
German numerical	1000	700	300	4.1.1.1.	30%

For addressing imbalanced datasets prevalent in fraud detection, three distinct propositions were considered:

4.2. Synthetic Minority Over-Sampling Technique (SMOTE) approach

Machine learning algorithms have difficulty learning when classification transactions are not equitably distributed. For that, we have used SMOTE (Synthetic minority oversampling

technique), it's a popular oversampling method for solving imbalanced dataset problems (Sahayasakila et al., 2019)

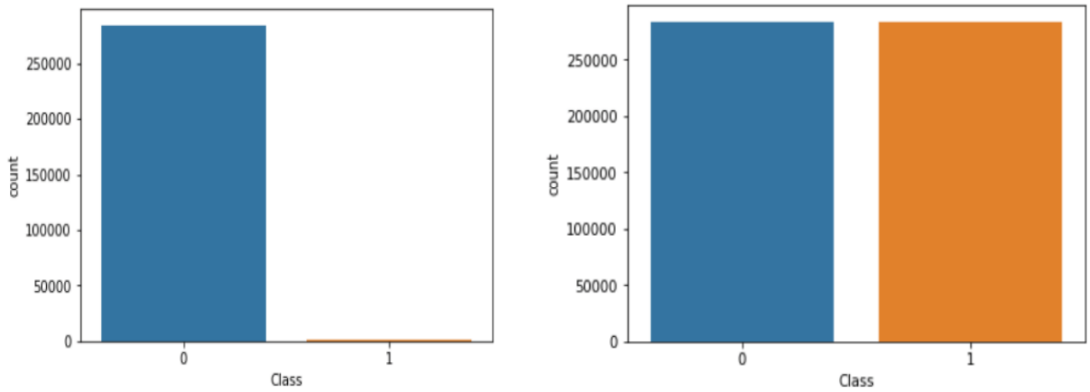


Figure10. Percentage of fraudulent transactions before and after oversampling the original dataset

We have 492 frauds out of 284,807 transactions. It accounts for 0.172% of all transaction datasets.

Our CCFD dataset contains 30 features as mentioned above, as well as the analysis of the least relevant ones in the dataset, requiring visualization techniques in this operation. For this purpose, we used the feature selector technique, by Will Koehrsen. This method searches for the most important features and uses the gradient boosting algorithm. Therefore, the results in Figure 7 showed that all 30 features were selected when the cumulative importance was 99%.

Data pre-processing plays a critical role in preparing data for machine learning tasks, and one commonly used technique is normalization, specifically min-max scaling. Normalization aims to standardize the range of values across all features to prevent the model from giving undue importance to features with larger values.

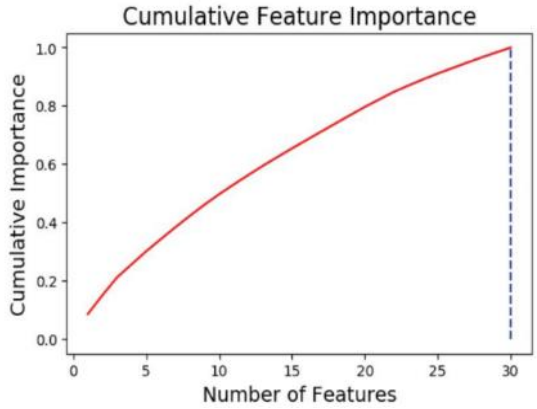


Figure11.Cumulative Feature Importance

The dataset is divided into K subsets or folds. Each fold contains an equal number of samples. In our case, K is set to 5. During each iteration, one of the K folds is used as the testing set, and the rest of the K-1 folds are used as the training set. The model is trained on the training set and

then evaluated on the testing set. This process is repeated K times, with each fold serving as the testing set once. The model's accuracy is calculated for each iteration, and the overall accuracy is obtained by taking the mean of these scores. To further validate the model, a separate validation dataset, not used during the K-Fold Cross Validation, is employed to measure its accuracy on unseen data. K-Fold Cross Validation is preferred over a single train-test split as it provides a more reliable estimate of the model's performance and ensures better generalization to new data. The validation accuracy obtained on the separate validation dataset is a more realistic measure of how well our model will perform on new, unseen data. The mean accuracy calculated during K-Fold Cross Validation can be considered an estimate of the model's general performance.

4.3. Autoencoder-based Approach

AE can be utilized to address the issue of imbalanced datasets by transforming the input data into a lower-dimensional latent space representation. This process enables the AE to compress the information from the original dataset into a more balanced representation, effectively capturing the underlying patterns and distributions of both the majority and minority classes. By doing so, the performance of subsequent classification models can be improved as the impact of class imbalance is reduced, leading to more accurate and robust predictions. Furthermore, incorporating AE as a pre-processing step can also reduce the computational complexity and training time required for deep learning models by reducing the dimensionality of the input data while retaining crucial features and information. During the training phase, the AE learns how to shrink the input data into a latent representation with few dimensions while minimizing the reconstruction error. This is done by getting the code (B) from the hidden layer (Figure 6). The code (B) consists of fewer neurons than the input layer, resulting in a representation of the input features with fewer dimensions.

4.4. Multi-phase Preprocessing approach

To address the complex, non-linear relationships inherent in transaction data, we integrate FCDN, which excels at modeling intricate dependencies. Moreover, we apply the Synthetic Minority Over-Sampling Technique (SMOTE) to enhance our model's performance. SMOTE generates synthetic samples to balance the class distribution, mitigating the challenge of imbalanced datasets.

The synergy between AE, FCDN, and SMOTE enhances our model's ability to identify and classify fraudulent transactions. Additionally, we employ ensemble majority voting to aggregate predictions from these models, reducing errors and enhancing prediction accuracy. Our architecture remains adaptable, allowing for fine-tuning based on empirical data and dataset characteristics.

Incorporating a multi-stage architecture into our approach empowers our fraud detection model with advanced capabilities. However, this complexity necessitates a comprehensive understanding of each component's role and collective impact on model performance. Implementing such a complex architecture introduces several challenges, including selecting and fine-tuning hyperparameters, substantial training times, and allocating computational resources. Maintenance of this intricate model structure in the face of evolving financial landscapes and new data requires careful consideration. To address these challenges, we have adopted strategies to streamline implementation and maintenance, organizing our codebase modularly for manageability and allocate computational resources efficiently. While acknowledging the complexity introduced by our model, it is crucial to recognize that these intricacies are balanced against the substantial gains in fraud detection accuracy, driven by our pursuit of optimal security measures and considered trade-offs to achieve superior performance.

5. Comparative Analysis and Effectiveness

5.1.Evaluation metrics

The confusion matrix, which lists the predictions of the model in terms of true positives, true negatives, false positives, and false negatives, was used to assess the performance of the classification model. The confusion matrix offers a thorough overview of these counts to calculate several performance measures, including accuracy, precision, recall (sensitivity), and F1 score. These metrics provide valuable information about the model's performance in accurately categorizing occurrences and identifying fraudulent transactions. They offer insights into the model's ability to distinguish between legitimate and fraudulent events.

Accuracy is the ratio of samples that are successfully classified to all other samples and is frequently used to evaluate the performance of classification models. Accuracy by itself, though, could not be sufficient to evaluate model performance in the case of imbalanced datasets. As a result, new evaluation criteria like sensitivity, precision, and F1-score are used.

Recall: Recall that the metric of sensitivity, also known as sensitivity or true positive rate, quantifies the percentage of positive samples that a model correctly recognizes.

Rec = TP / (TP + FN) (28)

Precision: The precision metric measures the percentage of positive samples that a model correctly identifies out of all samples predicted to be positive.

Pre = TP / (TP+ FP) (29)

F1-score: is a statistic that combines recall and precision into one number that represents the harmony of the two metrics.

F1-score = 2 * ((Pre * Rec) / (Pre + Rec)) (30)

5.2. Results found

5.2.1. SMOTE-enhanced Model

This work was executed in Google Collaboratory using GPU, implemented with Python. Based on the precision results for different ratios of the dataset, we have opted to divide our dataset as follows: 80% for learning and 20% for the testing phase. Data pre-processing was a very necessary step to get rid of useless attributes. For this purpose, we have used Will Koehrsen's famous method "Feature selector tool". However, all 30 features were selected for the experiment when the cumulative importance was 99%.

Since the dataset was very unbalanced, fraudulent transactions represented just 0.173% of all datasets. MLP and ELM algorithms have difficulty learning and give incorrect results. Therefore, we have solved this problem using Smote (Synthetic minority oversampling technique). A total set of data test samples is 113502; including 56914 of data are fraud transactions. To select the appropriate activation function for the ELM, Table 2 shows the training and testing accuracy for each activation function.

Table.2 Accuracy rates for ELM with different activation functions

Activation function	**Training Acc**	**Testing Acc**
Tanh	50.04%	49.91%
Sine	51.09%	50.80%
Tribas	60.75%	60.61%
Hardlim	51.53%	51.38%
RBF	95.58%	95.46%
Sigmoid	50.07%	49.94%

Then RBF (radial basis function) activation function gives the best classification results observed in Table 2.

Table3. A confusion matrix for CCFD using ELM.

	Predicted		
Actual		0	1
	0	55938	618
	1	4528	52418

ELM model in (Table 3) achieved:

Accuracy: 95.46%

Recall: 92.04%

Precision: 98.83 %

Also, the MLP model in (Table 4) achieved an accuracy of 97.84% using four hidden layers with (20,10,20,20) neurons:

Table.4 A confusion matrix for CCFD using MLP.

		Predicted	
Actual		0	1
	0	56182	374
	1	2073	54873

MLP model in (Table 4) achieved:

Accuracy: 97.84%

Recall: 96.35%

Precision: 99.32%

ELM detected only 52418 out of 56914 fraudulent transactions, while MLP detected 54873 fraudulent transactions. Also, ELM considered 4528 out of 56914 fraudulent transactions as normal transactions, while MLP declared just 2073 out of 56914 fraudulent transactions as normal transactions. Besides the accuracy of MLP is 97.84% is highly outperforming the accuracy of ELM 95.46%. The durations of training and testing classifiers were demonstrated in Figures 10 and 11.

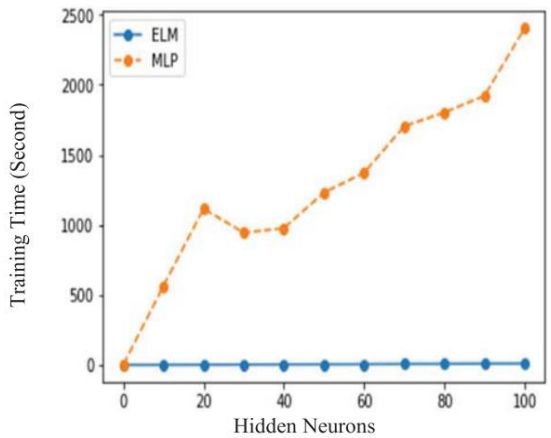


Figure14. The training time of MLP and ELM

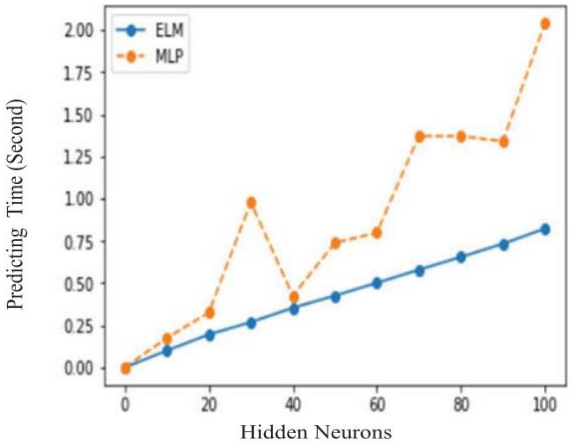


Figure15. The predicting time of MLP and ELM

In Figure 13, ELM required a few seconds to predict transactions as fraudulent or legitimate, while the MLP required several minutes.

So, the presentation of the prediction time curve takes the form of a straight line, however, its prediction speed is very fast compared to the MLP and generally is fast compared to other classical classifiers (Manju Parkavi et al., 2017). Also, compared to the results given in [7], MLP with an accuracy of 97.84%, outperforms Naïve Bayes, k-Nearest Neighbor, and Logistic Regression. Like in (Awoyemi et al., 2017a) MLP gave the best results compared to other classical algorithms. Consequently, deep networks work best with a large dataset, as they can be used to detect fraudulent transactions.

5.2.2. Autoencoder-based Model

To better visualize the results of the proposed model performance, the best training and testing accuracy on the two datasets are shown respectively in Figures 16 and 18. Thus their training and testing loss functions are shown in Figures 17 and 19.

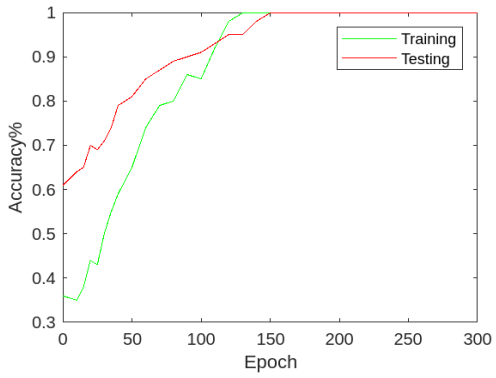


Figure16. Training and validation accuracy of the proposed model on CCFD.

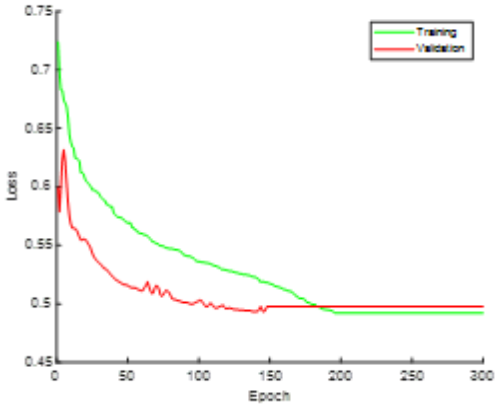


Figure17. Loss Function of training and validation proposed model on CCFD.

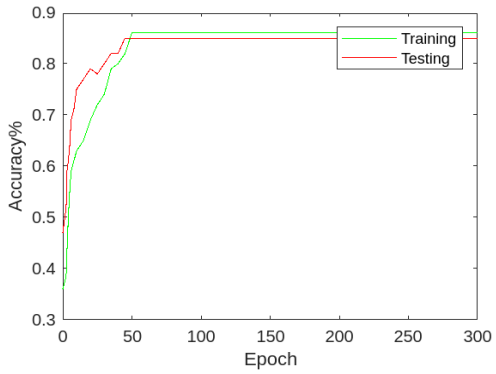


Figure18. Training and validation accuracy of the proposed model on the German dataset.

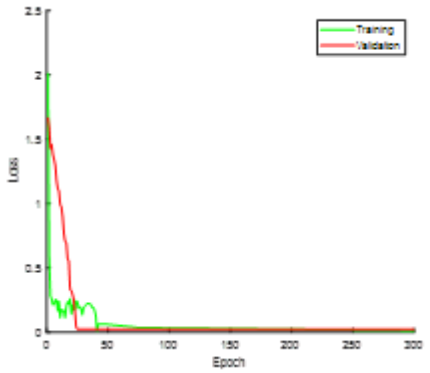


Figure19. The loss function of training and validation proposed model on German dataset.

The analysis of Figures 14-15 provides valuable insights into the performance of our proposed model on both the Credit Card Fraud Detection (CCFD) and the German datasets.

In Figure 14, our model's learning performance on the CCFD dataset peaks at the 150th epoch and remains stable throughout the rest of the training and validation period. The fact that the learning performance stabilizes indicates that the model effectively learns the patterns and features specific to the CCFD dataset. It translates to consistently high accuracy on training and validation sets, showcasing the model's ability to generalize well to unseen data. The minimum value of the loss function in Figure 15 further supports the successful optimization of the model's parameters, effectively minimizing the error during training. Stabilizing the validation accuracy after the 50th epoch in Figure 15 reaffirms our model's effectiveness and robustness on the German dataset.

In summary, Figures 14-15 provide valuable evidence of our model's ability to achieve excellent generalization performance on the German dataset. The convergence of training and validation accuracy, along with the stable and decreasing trend of training and validation loss, showcases the model's effectiveness in learning from the dataset, generalizing to new data, and avoiding overfitting. It reinforces the reliability and accuracy of our proposed model in detecting credit card fraud instances in the German dataset.

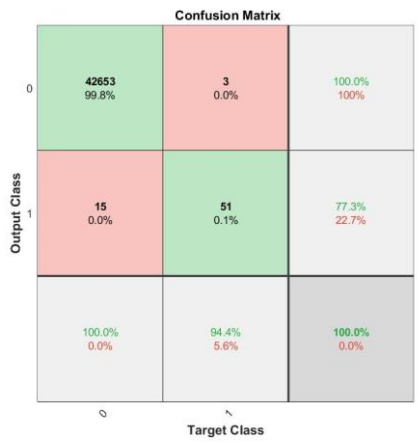


Figure20. Confusion matrix of CCFD

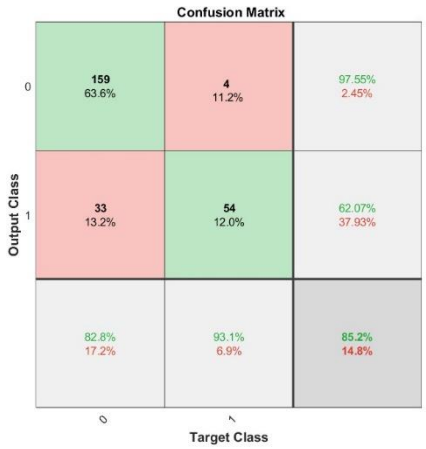


Figure21. Confusion matrix of a German dataset

For CCFD, the accuracy is 99.96%, which is an impressive result. It shows that our model has a high overall accuracy in classifying fraudulent and legitimate transactions. The recall is 94.4%. Our model successfully detects 94.4% of all the fraudulent transactions in the dataset, minimizing the number of false negatives. The precision is 77.3%. It indicates that when our model predicts a transaction as fraudulent, it is correct in 77.3% of cases, reducing the number

of false positives. While the F1-score is 84.99%, indicating a good trade-off between precision and recall.

Overall, the confusion matrix and performance metrics in Figure 17 demonstrate the robustness and effectiveness of our model on the CCFD dataset. The high accuracy, recall, precision, and F1 score indicate that our model accurately detects credit card fraud instances while minimizing false negatives and positives. It makes our model a reliable and robust credit card fraud detection solution with significant potential for real-world applications. The confusion matrix and performance metrics for the German dataset, as shown in Figure 18, demonstrate that our model performs reasonably well in detecting credit card fraud. The high accuracy of approximately 83.81% indicates that a significant portion of transactions is classified correctly as fraudulent or legitimate. The high recall of approximately 93.10% signifies that our model effectively identifies most fraudulent transactions, and minimizes false negatives.

5.2.3. Multi-phase Preprocessed Model

Experiments were executed on Python 3 (Jupyter Notebook) on a PC with a 2.5 GHz CPU (Intel Core i5-7200), 12 GB RAM, and Windows 7(64-bit). The average of the 5-fold cross-validation represents the findings, while each fold was repeated 10 times.

In our experiments, the first AE algorithm is applied to train the network effectively. In the training phase, only fraudulent transactions are used as input, allowing the network to learn and generate a compressed representation that can distinguish fraud from legitimate transactions. Experimental evaluation determines the number and size of layers, resulting in a five-layer architecture: 30-15-10-15-30 from left to right. The network training continues until the reconstruction error is minimized. The Euclidean distance between the input and its reconstruction is used to calculate the reconstruction error. After 30 epochs, it is observed that the reconstruction error is significantly reduced, indicating the network's effectiveness in capturing essential information and minimizing redundancy. Figure 3 visually represents the decrease in the loss value as the number of epochs increases, demonstrating the improvement in reconstruction accuracy.

In the training phase, the reconstruction error reaches its minimum value of 0.35 when the epoch is set to 30, indicating that the model has successfully learned to reconstruct the training data with a low error rate. However, in the testing phase, we used a specific fraction of the fraudulent transactions dataset set aside exclusively for testing and not utilized in training or validation steps.

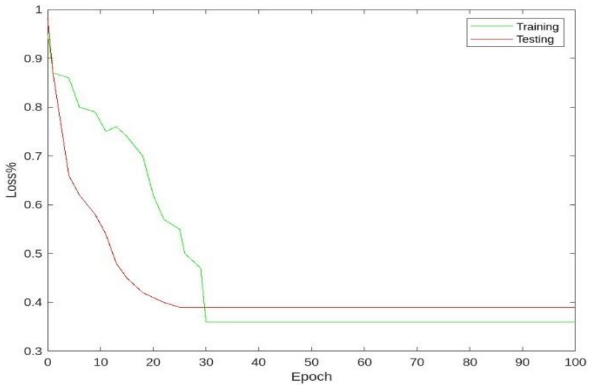


Figure22. Evaluation of AE Loss Function vs. Epochs

For the FCDN architecture, we explored different configurations to determine the optimal number of fully connected hidden layers. Each addition of a hidden layer was evaluated based on metrics and testing time. In summary, the choice of the optimized Fully Connected Deep Network (FCDN) architecture, characterized by four hidden layers with 20, 10, 20, and 20 neurons, resulted from a methodical and iterative process. The architecture balances the complexity necessary to capture intricate fraud patterns with practicality and computational efficiency considerations. It underwent hyperparameter tuning, cross-validation, and regularization to ensure robust performance while preventing overfitting. The architecture's selection was validated through empirical performance assessments and comparative analyses against alternative approaches. Ultimately, it represents a well-considered choice to maximize accuracy and effectiveness in fraud detection.

Eventually, we selected a three-layer for FCDN with hidden layers of 100, 100, and 2 neurons, respectively, as illustrated in Table 5 for AE and FCDN architecture. The model utilized an adaptive learning rate of 0.001 and employed the Adam optimization algorithm as the loss function.

Table.5 Optimized parameters

Proposed Model	Parameters
FCDN	(30,20,10,20,20,1), learning rate = 0.001, Adam optimizer.
AE(Dimensionality reduction) & FCDN	(30,20,30) + (20,100,100,2,1), learning rate = 0.001, batch size=32, Adam optimizer
AE(Reconstruction error)	(30,15,10,15,30), learning rate = 0.001,batch size=32, RMSE loss function, Threshold =0.4

The dimensionality reduction model, coupled with the FCDN, successfully reduced the dimensionality of the input data and extracted crucial features for classification. During the experimentation process, it was determined that the dimensionality reduction model should be constructed with a single hidden layer comprising twenty hidden neurons. For optimization, a

batch size of 32 was employed, and the Adam optimizer was utilized. However, we further optimize the number of fully connected hidden layers in our model by systematically evaluating the addition of each layer. This evaluation process considers the impact on evaluation metrics and testing time, allowing us to find the optimal configuration. Finally, optimized FCDN architecture is illustrated in Table 5 with four hidden layers, where the hidden layers have 20, 10, 20, and 20 neurons, an adaptive learning rate of 0.001, and an Adam optimization loss function.

The rest of this section has two parts: the first analyzes the results of the two datasets, and the second compares our method with existing classification methods.

Table.6 Computing Results on the Datasets

	CCFD				Synthetic financial dataset			
Algorithm	Accu	Recall	Precision	F1-score	Accu	Recall	Precision	F1-score
FCDN	97.84	96.32	99.16	97.72	95.84	92.17	96.41	94.24
AE(Dimensionality Reduction)+FCDN	98.2	96.78	98.83	97.79	97.32	96.67	97.13	96.9
AE(Reconstruction error)	97.64	98.46	94.57	96.48	96.64	97.26	90.65	89.1
Proposed model	98.96	98.81	99.71	99.26	97.71	98.31	97.54	97.93

The performance metrics of different algorithms for CCFD are summarized in Table 6. The FCDN algorithm achieves an accuracy of 97.84%, a recall of 96.32%, a precision of 99.16%, and a F1-score of 97.72%. When combining the AE (dimensionality reduction) with FCDN, the performance significantly improves, with an accuracy of 98.2%, a recall of 96.78%, a precision of 98.83%, and an F1-score of 97.79%. The AE (reconstruction error) algorithm achieves an accuracy of 97.64%, a recall of 98.46%, a precision of 94.57%, and a F1-score of 96.48%. However, the proposed model outperforms all other algorithms, achieving an accuracy of 98.96%, a recall of 98.81%, a precision of 99.71%, and a F1-score of 99.26%. These results demonstrate the superiority of the proposed model in credit card fraud detection.

On a synthetic financial dataset, the results obtained were as follows: The FCDN algorithm achieved an accuracy of 95.84%, indicating its ability to classify financial data accurately. It demonstrated a recall rate of 92.17%, ensuring that it correctly identified a high percentage of relevant instances. The precision of 96.41% indicated its ability to minimize false positives, making it reliable in detecting true positives. The F1-score of 94.24 reflected a balanced performance between precision and recall.

The AE (dimensionality reduction) combined with the FCDN algorithm showed promising results, with an accuracy of 97.32%. It exhibited a recall rate of 96.67%, indicating its

effectiveness in capturing a high proportion of relevant instances. The precision of 97.13% suggested a low rate of false positives. The F1 score of 96.9% showcased a balanced performance between precision and recall.

The AE (reconstruction error) algorithm achieved an accuracy of 96.64%. It displayed a recall rate of 97.26%, indicating its ability to correctly identify relevant instances. However, its precision of 90.65% indicated a higher rate of false positives compared to other algorithms. The F1-score of 89.1% reflected a lower balance between precision and recall.

The proposed model outperformed other algorithms with an accuracy of 97.71%. It demonstrated a high recall rate of 98.31%, indicating its strong ability to identify relevant instances. The precision of 97.54% suggested a low rate of false positives. The F1-score of 97.93% showcased a balanced performance between precision and recall, indicating the model's effectiveness in classifying the synthetic financial dataset accurately.

Overall, these results highlight the performance of different algorithms in handling the synthetic financial dataset, with the proposed model exhibiting the highest accuracy and a well-balanced precision-recall trade-off.

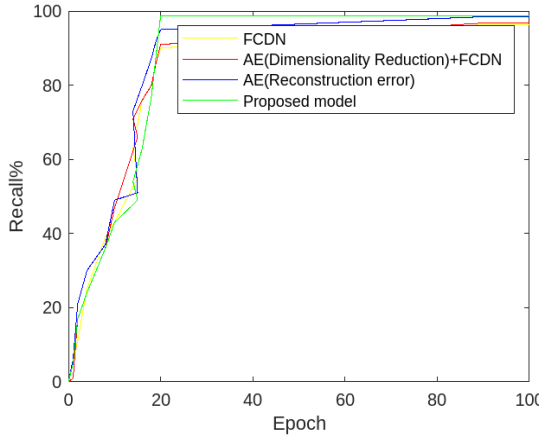


Figure23. Performance Evaluation of Recall using the Proposed Model on CCFD

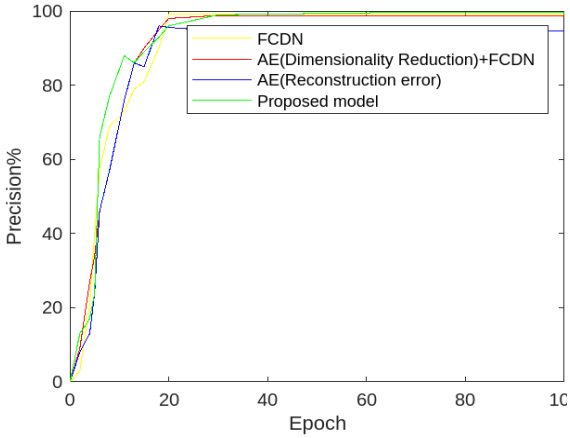


Figure24. Performance Evaluation of Precision using the Proposed Model on CCFD

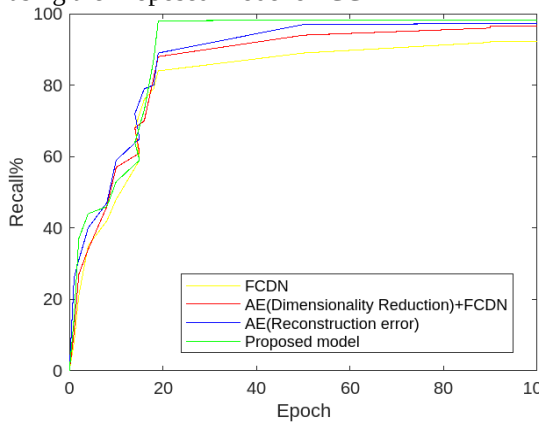


Figure 25 Performance evaluation of recall using the Proposed Model on Synthetic financial dataset

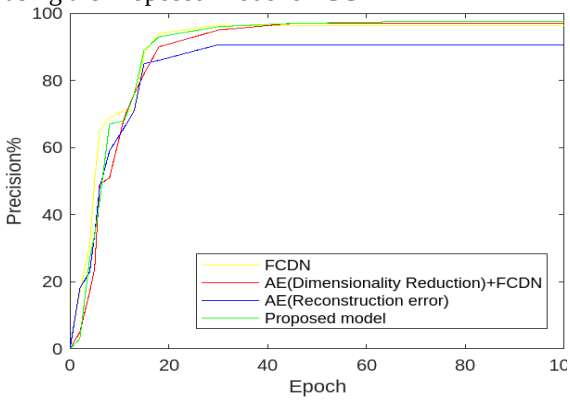


Figure 26 Performance evaluation of precision using the Proposed Model on Synthetic financial dataset

Figure 21 provides valuable insights into the performance of the proposed model in detecting credit card fraud during the testing phase at epoch N.20. It focuses on the evaluation of recall, demonstrating the model's ability to correctly identify fraudulent transactions. Higher values in the graph indicate better recall rates and a reduced number of false alarms. On the other hand, Figure 22. presents the evaluation of precision during the same testing phase using the proposed model. The graph showcases the model's accuracy in classifying transactions as fraudulent, with higher values representing higher precision levels. This highlights the model's capability to accurately identify fraudulent cases and minimize false positives. These figures collectively offer essential information regarding the effectiveness of the proposed model in detecting credit card fraud, emphasizing its ability to achieve both high recall and precision rates while minimizing false alarms and false positives.

Figure 23 provides valuable insights into the performance of the proposed model in detecting credit card fraud on synthetic financial datasets during the testing phase at epoch N.45. It focuses on the evaluation of recall, demonstrating the model's ability to correctly identify fraudulent transactions. Higher values in the graph indicate better recall rates and a reduced number of false alarms. On the other hand, Figure 24 presents the evaluation of precision during the same testing phase using the proposed model. The graph showcases the model's accuracy in classifying transactions as fraudulent, with higher values representing higher precision levels. This highlights the model's capability to accurately identify fraudulent cases and minimize false positives. These figures collectively offer essential information regarding the effectiveness of the proposed model in detecting credit card fraud, emphasizing its ability to achieve both high recall and precision rates while minimizing false alarms and false positives.

5.3. Comparative Analysis

Table 7 encompasses and evaluates the results of the three models on the utilized datasets, including results from other works on the same datasets. in terms of accuracy, recall, precision, and F1 score. These metrics indicate the model's performance in fraud detection on these specific datasets.

Table.7 Results of the proposed models compared to previous studies.

Dataset	Paper Citation	Algorithm	Accuracy	Recall	Precision	F1 score

CCFD	(Itoo et al., 2020)	Logistic regress	95.9	83.9	99.1	90.9
		Naïve Bayes	91.5	71.8	100	83.6
		KNN	75.1	40.5	48.3	44.1
	(SP Maniraj, A Saini, 2019)	Isolation Forest	99.76	-	-	-
		Local Out Factor	99.67	-	-	-
	(Dubey et al., 2020)	KNN	99.82	3.93	71.42	7.45
		ANN	99.92	76.19	81.15	78.59
	(A. A. Taha & Malebary, 2020)	Optimized light gradient boosting machine	98.40	-	97.34	56.95
	(Al-shabi, 2020)	AE (Threshold=0.7)	80.00	91	60	40
	(Zou et al., 2019)	Denoising AE with Neural Network	97.93	84	-	-
	(Xie et al., 2023)	LSTM	-	56.2	50.1	53
		GRU	-	60.8	50.1	55
		Bi-RNN	-	77.3	50.1	60.8
		TAH-RNN	-	89	50.3	64.3
		TLSTM	-	82.7	50	62.3
		HAInt-LSTM	-	68.1	50.1	57.7
		Ug-RNN	-	67.5	50.1	57.5
		STA-NN	-	66.8	50.2	57.3
		TAI-LSTM	-	99.6	50.7	67.2
	(Xie et al., 2022)	STA-NN	-	66.4	50.2	57.1
		LSTM	-	56	50.1	52.9
		GRU	-	60.8	50.1	55
		Bi-RNN	-	77.1	50	60.7
		UGRNN	-	67.5	50.1	57.5
		HAINT-LSTM	-	68.2	50.1	57.8
		T-LSTM	-	83	50	62.4
		TAH-LSTM	-	88.8	50	64.1
		TH-LSTM	-	**99.6**	50.4	66.9
	Method 1		**97.84**	**96.35**	**99.32**	**97.81**
	Method 2		**99.96**	**94.4**	77.3	**84.99**
	Method 3		**97.96**	**97.81**	**98.71**	**98.25**
German dataset	(Tripathi et al., 2020)	Evolutionary ELM	80.57	**94.42**	48.33	63.93
	(Tao et al., 2020)	ACFSVM	79.7	-	-	63.5
	Method 2		**83.81**	**93.1**	62.07	**74.48**
Synthetical financial	(Hajek et al., 2022)	KNN	90.89	98.51	11.31	20.29
		SVM	99.41	54.51	92.08	68.49
		RF	99.57	79.66	83.32	81.45
		XGBoost	**99.63**	82.4	82.40	83.91
		XGBOD	99.53	70.84	80.18	78.93
	(Cochrane et al., 2021)	Decision Tree	70.84	41.7	**99.95**	58.85
		Linear Regression	54.75	100	9.5	17.35
		Logistic Regression	88.31	**99.77**	76.8	86.79
	Method 3		**97.71**	**98.31**	**97.54**	**97.93**

The results obtained from the analysis demonstrate the remarkable efficacy of the proposed models in credit card fraud detection. Model 1, utilizing Multilayer Perceptron (MLP) and Extreme Learning Machine (ELM), outperforms classical algorithms such as Naïve Bayes, k-Nearest Neighbor, and Logistic Regression, achieving an accuracy of 97.84%. This highlights the superior performance of deep neural networks, particularly in handling large datasets and effectively detecting fraudulent transactions.

Model 2 emerges as a standout performer with an impressive accuracy of 99.96%, surpassing previous works and showcasing a significant reduction in false alerts. The integration of Autoencoder (AE) in this model proves instrumental, enabling unsupervised feature learning and dimensional reduction of credit card transactions. This learned representation enhances the subsequent Fully Connected Deep Network (FCDN), contributing to improved decision-making capabilities.

The success of Model 2 can be attributed to several key factors. Firstly, the AE facilitates unsupervised feature learning, reducing the dimensionality of transactions while capturing essential patterns. This, in turn, acts as a robust feature extractor for the FCDN, enhancing the model's decision-making prowess. Secondly, the AE's noise elimination capability ensures the model's robustness to noisy data, contributing to improved generalization. Additionally, the AE aids in addressing imbalanced data challenges, reducing biases, and improving fraudulent transaction detection.

The third model further advances fraud detection by employing a combination of adopting strategy, focusing on precision and recall, and incorporating SMOTE and ensemble methods for robust performance., resulting in enhanced capacity and expressiveness. The ensemble of learned representations contributes to improved discrimination between fraudulent and legitimate transactions, showcasing superior predictive power without succumbing to overfitting.

Comparisons with other algorithms, such as linear regression, XGBoost, decision tree, and logistic regression, reveal the well-balanced performance of the proposed models. While certain algorithms may excel in precision or recall, the proposed models achieve a harmonious blend of high precision, recall, and overall performance.

6. Conclusion

Credit card fraud detection poses a significant challenge, leading companies to invest substantial resources in developing algorithms to prevent and identify fraudulent transactions. This study

aims to detect credit card fraud using two artificial neural network algorithms, Multilayer Perceptron (MLP) and Extreme Learning Machine (ELM). Evaluation based on metrics such as accuracy, recall, precision, true positive rate, false positive rate matrix, and classification time revealed MLP's outperformance over ELM. However, ELM showcased faster prediction times owing to its simpler architecture when predicting new fraudulent transactions.

Advantages:

Multilayer Perceptron (MLP) exhibited superior performance across various metrics compared to Extreme Learning Machine (ELM).

Extreme Learning Machine (ELM) demonstrated quicker prediction times due to its simpler architecture compared to MLP.

Limitations:

MLP's computational intensity might affect its time efficiency despite its superior performance metrics. ELM, while faster in prediction, might lag behind MLP in specific fraud detection-related metrics.

Moreover, this study introduces a novel approach for detecting fraudulent credit card transactions by employing Autoencoder (AE) in conjunction with a fully connected deep learning network algorithm. The proposed model's evaluation on two benchmarked credit card datasets CCFD and German-numerical showcased its superiority in accuracy, recall, precision, and F1 score, surpassing prior models. Additionally, the model effectively reduced false alarms by minimizing false positive and false negative rates.

To enhance the proposed model's practicality, future research avenues should focus on broader validation across varied credit card datasets. Collaborative real-world implementations with financial institutions or credit card companies would offer valuable insights into its scalability and effectiveness in detecting fraudulent transactions.

However, the integration of Autoencoder (AE) and optimized Fully Connected Deep Network (FCDN) presents several challenges for future research:

It increased computational complexity and training time due to AE training and FCDN fine-tuning, especially for larger datasets.

Tuning multiple hyperparameters for optimal performance could be time-consuming.

The model's effectiveness heavily depends on the quality and representativeness of the credit card transaction dataset used for training, potentially limiting its performance if the dataset lacks adequate fraud pattern representation.

Overfitting risks in deep learning models, including FCDN, particularly with small or noisy datasets, require careful regularization and validation strategies.

Advantages:

Autoencoder (AE) and Fully Connected Deep Network (FCDN) demonstrated superiority in accuracy, recall, precision, and false alarm reduction compared to previous models.

The integration of AE and FCDN enhanced the model's capability to effectively detect fraudulent credit card transactions.

Limitations:

The combined AE and FCDN architecture's complexity might demand increased computational resources and training time.

Time-consuming hyperparameter tuning due to numerous parameters in the model architecture.

In conclusion, this research proposes a novel approach combining AE and FCDN models for credit card fraud detection, effectively addressing challenges posed by imbalanced datasets and displaying high recall and precision in detecting fraudulent transactions. However, the model's integration complexity and computational demands, along with data quality and quantity dependencies, present significant considerations for future research and practical application. Our research “Credit Card Fraud Detection: Addressing Imbalanced Datasets with a Multi-Phase Approach ”proposes an innovative approach integrating Autoencoders (AE) and Fully Connected Deep Networks (FCDN) to combat credit card fraud. This comprehensive model effectively addresses the challenges posed by imbalanced datasets, achieving remarkable precision and recall in identifying fraudulent transactions. The system's ability to learn intricate features through AE models trained on fraudulent transactions significantly enhances fraud detection. Additionally, techniques such as dimensionality reduction, SMOTE-based preprocessing, and majority voting for predictions contribute to the model's robustness and accurate detection. Despite its commendable performance, the architecture faces challenges. Its computational demands, including prolonged training periods and extensive hyperparameter tuning, limit its scalability in resource-constrained environments. The model's reliance on high-quality and diverse training data poses potential constraints on its effectiveness in scenarios with limited labeled data. Notably, while the architecture offers promising outcomes, its complexity and resource dependency necessitate further optimizations for broader applicability and practical implementation in real-time fraud detection scenarios. As we conclude the exploration of 'Detecting Fraudulent Transactions Using Stacked Autoencoder Kernel ELM Optimized by the Dandelion Algorithm,' the promising outcomes underscore the potential of our proposed method in addressing challenges associated with credit card fraud detection. Building upon these findings, the next chapter, titled 'Optimization Model and Performance Improvement,' delves into further enhancements and refinements. We transition from the innovative AE-

KELM architecture by introducing a comprehensive optimization model, aiming to elevate the overall performance and efficiency of our fraud detection system. This chapter intricately explores methods to streamline computational complexity, improve training times, and fine-tune hyperparameters for optimal results. Join us in the next chapter as we delve into the intricacies of optimizing our fraud detection model to ensure its effectiveness in real-world scenarios.

Chapter III: Optimization Model and Performance Improvement

1. Introduction

Detecting credit card fraud in real-time demands highly efficient and accurate machine learning models that can swiftly analyze massive transaction data while minimizing false positives and negatives. To achieve this, continuous refinement and optimization of fraud detection models are imperative.

The landscape of fraud detection has advanced significantly through the integration of artificial intelligence techniques. Traditional algorithms like Multilayer Perceptron (MLP) and Extreme Learning Machines (ELM) have solutions for fraud detection systems. However, they face challenges in adaptability, learning speed, and scalability, particularly when handling intricate or imbalanced datasets.

In response, researchers have delved into innovative methodologies and optimization techniques. These advancements encompass the integration of autoencoder (AE), kernel Extreme Learning Machine (KELM), and optimization algorithms. Our novel approaches aim to enhance the efficiency, accuracy, and adaptability of the optimization model and performance improvement has led to the formulation of more robust architectures entitled “Stacked AE-KELM optimized by the dandelion algorithm” (S-AEKELM-DA). This sophisticated approach strives to tackle the limitations of conventional models, offering heightened accuracy, improved generalization, and enhanced computational efficiency.

In addition, the continuous development and improvement of ELM show superior performance in the following areas: image classification (Cao et al., 2013), text classification (Conference & Technologies, 2016), and bioinformatics (Z. Wang et al., 2014). The input weights and bias are generated randomly, which directly contributes to the instability of the ELM model and has made parameter optimization for this model a popular topic.

Researchers have suggested different optimization techniques to boost ELM performance further and ensure its stability (Tripathi et al., 2020)(Shariati et al., 2022)(X. Li et al., 2017). However, previous work has had some limitations if the original dataset has been complicated or highly imbalanced.

Kernel ELM (KELM) was proposed by (Guang-bin Huang & Siew, 2014) to resolve this problem. They discovered that the link weights between the hidden and input layers are unnecessary, and KELM outperforms ELM in terms of performance and learning time.

Our study proposed a stacked AE kernel ELM optimized by the dandelion algorithm (S-AEKELM-DA). This study aimed to solve the specified machine learning objective by splitting an extensive network into multiple AE-KELM. The various AE-KELMs are stacked on different levels and connected serially. While C and σ are, respectively, the regularization factor and the kernel parameter of AE-KELM, they are randomly generated and optimized in the training phase without considering hidden nodes. Our architecture contributes to the target output for each level. The proposed stacked AE-KELM optimized by the DA presents the following advantages:

Employing the stacked autoencoder kernel ELM makes the number of hidden nodes in each layer unimportant. It eliminates the subjective and time-consuming process of selecting the appropriate number of hidden nodes, making the model more efficient and reducing the danger of overfitting or underfitting.

The best values of the AE-KELM parameters were adjusted using an optimization technique based on the dandelion algorithm, which was more optimal than that established using particle swarm optimization and the bat algorithm.

Our proposed method unifies all transformation matrices into only two matrices. This unification simplifies the model's structure and reduces storage requirements. It can also shorten the model's execution time by minimizing computational complexity.

According to the simulation results, the S-AEKELM-DA can achieve good testing accuracy results, including learning large amounts of data and detecting problems without causing memory problems.

2. Optimization Techniques for Deep Learning Architectures in Fraud Detection

This section is dedicated to exploring and dissecting the array of optimization techniques meticulously employed in the realm of deep learning architectures, specifically focusing on their application in bolstering fraud detection mechanisms in credit card transactions. Within the landscape of sophisticated neural network architectures, the quest for refining performance metrics remains paramount. This section aims to delve into a spectrum of optimization methodologies meticulously tailored to enhance the efficiency and efficacy of these intricate architectures. By scrutinizing techniques such as Stacked Autoencoders, Kernel Extreme Learning Machines (KELM), and their integration with innovative optimization algorithms like the Dandelion Algorithm, this section seeks to uncover their profound impact on refining the

precision, recall, and overall accuracy of deep learning models in detecting fraudulent activities within credit card transactions.

(Fanai & Abbasimehr, 2023) proposed a two-stage framework comprised of a deep AE model as the dimensionality reduction technique, and three deep learning-based classifiers including DNN, RNN, and CNN_RNN to improve fraud detection accuracy. The Bayesian optimization algorithm is utilized to select the best hyperparameters of the employed models.

Hierarchical neural networks, such as multilayer perception and deep learning techniques, are currently very effective, but the learning process takes a long time (Almuteer et al., 2021)(El Hlouli et al., 2020)(Ramzan & Ahmed, 2022)(X. Yu et al., 2020). As a result, our proposition that incorporates the ELM into deep architecture as an AE leads to a high rate and quick learning speed. Due to hidden biases and input weights that are randomly chosen, ELM necessitates many hidden neurons, which may create a loss condition problem.

ELM does not update a weight iteratively. However, it attributes input weights and bias randomly without being adjusted during the training phase and then automatically determines the output weights the reason why ELM learns rapidly (Gao Huang et al., 2019).. Additionally, based on benchmarking classification and regression issues (El Hlouli et al., 2020), it was shown that in comparison to feedforward network-based gradient-descent, the testing and validating part of many applications may now be finished quickly because the ELM algorithm attempts to offer high generalization performance in a very short time. Therefore, ELM architecture has attracted the interest of many researchers (C. Li et al., 2022).

To further boost the performance of ELM, (Q. Zhu et al., 2005)proposed a differential evolution-based ELM algorithm (Y. Yu et al., 2019), which, to determine the output weights, principally uses the optimization capability of a differential evolution (DE) algorithm based on population and has a robust global optimization capability.

The EELM algorithm that (Y. Wang et al., 2011) proposed should theoretically result in the entire column of the hidden layer H by correctly choosing the input weights and bias before computing the output weights. It enhances the resilience property of the networks as well as the learning rate (testing accuracy, prediction accuracy, and learning time). The experimental outcomes based on both the benchmark function approximation and actual problems, such as applications for classification and regression, demonstrate the effectiveness of EELM.

(Jiang et al., 2018) suggested an advanced learning algorithm called IPSO-ELM that utilizes the benefits of both PSO and ELM. The modified PSO includes both the output weights norm and the RMSE on the validation set when choosing the input weights. Compared to the E-ELM, ELM, and LM algorithms, the proposed algorithm performs better in generalization.

(Tripathi et al., 2020) introduced an original activation function, and four benchmark credit-scoring datasets where a range of activation functions is employed in the simulations by employing the bat optimization technique to obtain optimum weights and biases. The findings demonstrate that the integration of bat with ELM to pre-train ELM by selecting optimum weights (between input and hidden layer) and hidden biases is better for assessing credit risk.

A novel swarm intelligence optimization algorithm called dandelion optimization DO was proposed by (Zhao et al., 2022). Unconstrained benchmark functions (CEC2017) were used to evaluate the optimization performance of DO, which was applied to four real-world problems and compared with many nature-inspired metaheuristic algorithms (nine famous algorithms) where DO verifies the constrained programmability performance.

(Gong et al., 2018) developed an optimization technique that was dandelion-inspired. To compare the proposed algorithm's performance in terms of exploration, exploitation, local optima avoidance, and convergence, twelve test functions were used. Regarding optimization accuracy and convergence speed, the results demonstrated that the DA typically finds solutions correctly and outperforms the BA, the EFWA, the GA, and the PSO on 12 benchmark functions. The outcomes of the ELM optimization further demonstrated that the DA performs well in different search regions.

ELM possesses certain limitations that require resolution. One significant drawback is the potential amplification of outcome variances acquired by classifiers in multiple trials due to the randomly generated input weights and bias.

(Guang-Bin Huang, 2004) proposed an enhanced iteration of ELM called kernel ELM (KELM). The researchers discovered that the connection weights between the hidden and input layers are unnecessary. KELM exhibits superior performance and quicker training speed when compared to ELM. As a result of its exceptional qualities, KELM has been successfully applied to address various practical problems (Chen et al., 2014).

Applying the kernel ELM model to the detection of credit card fraud was the primary goal of this study. This chapter suggests a SAE-KELM generating optimum parameters of kernel ELM, such as cost parameter C and kernel parameter σ optimized by the DA. A more profound overview is provided in the next part to demonstrate the success of the suggested approach.

3. Adaptation of Dandelion Optimization Algorithm to ELM Architecture

3.1. Dandelion Optimizer Algorithm

A novel swarm intelligence algorithm was created to study dandelion sowing behavior (X. Li et al., 2017) and proposed for solving continuous optimization problems. Dandelion

populations in DA are divided into two subpopulations: those that can be sown and those that cannot. Then, for each subgroup, different sowing strategies are applied. Meanwhile, to avoid sliding into the local optimum, another way of sowing is to conduct a subpopulation that is suited for sowing. With a dandelion, its seeds will be dispersed all around it. Dandelion seeding can be considered a hunt for an ideal in a specific area surrounding a point. This consists of the following three stages:

When seeds are in the rising stage, eddies from above cause them to rise spirally, or they may float locally in communities depending on the weather. Flying seeds continually alter their path in outer space as they fall during the descending stage. Seeds are placed in randomly chosen locations during the landing stage for them to grow.

Each dandelion in the population can be initialized as below:

$$Di = rand \times (U - L) + L \quad (31)$$

where i is an integer between 1 and the population size, $rand$ denotes a random number between 0 and 1, dim is the dimension of the dandelion vector, and L and U are expressed as follows:

$$L = [l1,\ldots,ldim],\ U = [u1,\ldots,udim] \quad (32)$$

$$fbest = \min (f(Di)) \quad (33)$$

$$D_{elite} = D_i \quad (34)$$

Dandelion seeds must reach a certain height to separate from their parent plant during the dispersal stage. The height they rise varies depending on wind direction and air humidity. The two main parameters that influence the dispersal of dandelion seeds are wind velocity and climate. A seed's ability to fly long or short distances depends on the wind speed (Purschke et al., 2014). Dandelion growth in nearby or remote areas is influenced by the weather, which determines whether dandelion seeds can fly. The two categories of weather that apply here are as follows:

Category 1: Wind speed distributions on a clear day can be considered lognormal distributions $\ln Y \sim N(u, \sigma 2)$. This distribution enhances the likelihood that dandelion seeds will spread to distant areas because random numbers are more distributed along the Y-axis. DO, therefore, prioritizes exploration in this instance. The wind scatters dandelion seeds around the search area in different directions. A dandelion seed's increasing height is dependent on the wind speed. The higher the dandelion flies, and the farther the seeds are dispersed, the stronger the

wind. The wind speed constantly changes the vortexes above the dandelion seeds, causing them to rise in a spiral shape as follows:

$D_{t+1} = D_t + \alpha * v_x * v_y * \ln Y * (D_s - D_t)$ (35)

$D_s = \text{rand}(1, Dim) * (U-L) + L$ (36)

$$\ln Y = \begin{cases} \frac{1}{y\sqrt{2\pi}} \exp\left[\frac{-1}{2\sigma^2}(lny)^2\right] & y \geq 0 \\ 0 & y < 0 \end{cases}$$ (37)

$\alpha = \text{rand}() * (\frac{1}{T^2}t^2 - \frac{2}{T}t + 1)$ (38)

$r = \frac{1}{e^\theta}$; $v_x = r * \cos\theta$; $v_y = r * \sin\theta$ (39)

where θ is a number between $[-\pi, \pi]$.

Category 2: Because of air resistance, humidity, and other conditions, dandelion seeds cannot rise appropriately with the wind on a rainy day. Dandelion seeds are being used in this particular case in local neighborhoods, and the equivalent mathematical formula is given below:

$D_{t+1} = D_t * k$ (40)

$k = 1 - \text{rand}() * q$ (41)

$q = \text{rand}() * (\frac{1}{T^2-2T+1}t^2 - \frac{1}{T^2-2T+1}t + 1 + \frac{1}{T^2-2T+1})$ (42)

For dandelion seeds in their ascending stage, the mathematical expression is as follows:

$$D_{t+1} = \begin{cases} D_t + \alpha * v_x * v_y * \ln Y * (D_s - D_t) & \text{if rand n} < 1.5 \\ D_t * k & \text{else} \end{cases}$$ (43)

The average position data following the rising stage reflects the dandelion descent. It helps the population grow and develop into promising communities. The corresponding mathematical formula is depicted below:

$D_{t+1} = D_t - \alpha * \beta_t * (D_{mean-t} - \alpha * \beta_t * D_t)$ (44)

In the ith iteration, D_{mean-t} represents the population's average position and its mathematical expression is given by the following:

$D_{mean-t} = \frac{1}{pop}\sum_{i=1}^{pop} D_i$ (45)

Exploitation is the focus of this section of the DO algorithm. Based on the first two stages, the dandelion seed randomly decides where to land. Hopefully, the algorithm will arrive at the optimal solution as the iterations advance gradually. The approximate location where dandelion seeds will survive the most readily is the best solution that can be found. Search agents use the expert knowledge of the current elite to their advantage in their local communities to converge accurately to the global optimum. The optimal global solution will eventually be discovered through population evolution. The expression of this behavior is as follows:

$$D_{t+1} = D_{elite} + \text{levy}(\delta)*\alpha*(D_{elite} - D_t*\partial) \quad (46)$$

The dandelion seed in the ith iteration was placed in the ideal location represented by *Delite*. The function of Levy flight is represented by Levy (δ), which is calculated as below:

$$\text{levy}(\delta)=s*\frac{w*\sigma}{t^{\frac{1}{\beta}}} \quad (47)$$

β is a random number between (0, 2) (β = 1.25 in this paper). s is a fixed constant of 0.01, and w and t are random numbers between [0, 1]. The mathematical expression of σ is given below:

$$\sigma = \left(\frac{\Gamma(1+\beta)*\sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2})*\beta*2^{(\frac{\beta-1}{2})}} \right) \quad (48)$$

$$\partial = \frac{2t}{T} \quad (49)$$

Dandelion Optimizer Algorithm
Input: popsize*, pop sizeim is the dimension of the variable, maximum number of iterations T, and calculate the population fitness values.* ***Output:*** *The best dandelion seed (D_{best}) and its fitness value (best)* *a. Randomly initialize a population pop.* *b. Determine each dandelion seed's fitness value f.* *c. According to fitness values, choose the best dandelion seed D_{elite}.* *d.* ***While*** *(t < T)* ***do*** ***Rise stage.*** *e.* ***if*** *randn () < 1.25* ***do*** *f. Use Equation (38) to produce adaptive parameters.* *g. Use Equation (35) to update dandelion seeds.* *h.* ***else if do*** *i. Use Equation (41) to generate adaptive parameters.* *j. Use Equation (40) to update dandelion seeds.* *k.* ***end if*** ***Decline stage.*** *l. Use Equation (43) to update dandelion seeds.* ***Land stage*** *m.Use Equation (45) to update dandelion seeds.*

n. Sort dandelion seeds according to fitness values, from good to bad.
o. Update D_{elite}
p. ***if*** $f(D_{elite}) < f(D_{best})$
q. $D_{best} = D_{elite}$, $f_{best} = f(D_{elite})$
r. ***end if***
s. ***end While***
t. ***return*** D_{best} *and* f_{best}

3.2. Kernel Extreme Learning Machine

As described by (Ding et al., 2014) ELM is a single-hidden-layer neural network that operates feed-forwardly. The input weights and hidden biases are randomly initialized, whereas the output weights are determined within a single iteration. ELM achieves good performance and fast learning speed. To this end, ELM is used in several areas of artificial intelligence. The classical ELM algorithm is based on three steps with N training data, L hidden layer neurons, input samples X, and target matrix Y. For i = {1, 2, ……, N}, Xi ϵ R^n and Yi ϵ R^c, activation function f:

(1) Initialize randomly input weight P and hidden biases where the input nodes to the hidden layer neuron i are connected by the vector Pi = [Pi1, Pi2, …, Pin].

$$P = \begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & \ddots & \vdots \\ P_{L1} & \cdots & P_{Ln} \end{bmatrix}, s = \begin{bmatrix} s_1 \\ \vdots \\ s_L \end{bmatrix} \quad (50)$$

(2) Calculate matrix H of output hidden layer

$$H = \begin{bmatrix} f(P_1 * X_1 + s_1) & \cdots & f(P_L * X_1 + s_L) \\ & \vdots \ddots \vdots & \\ f(P_1 * X_N + s_1) & \cdots & f(P_L * X_N + s_L) \end{bmatrix}_{N*L}$$

(3) Compute output matrix weight with

$$O = H^{\dagger} * Y \quad (51)$$

The Moore Penrose inverse of matrix H is denoted as $H^{\dagger}$ (Serre, 2002)(Guang Bin Huang et al., 2012).

The Karush–Kuhn–Tucker theory asserts that the output weight *O* is computed as follows where C is the regularization factor:

$$O = H^T \left(\frac{I}{C} + H^T H\right)^{-1} Y \quad (52)$$

The output of an RBF network (Avenue, 2004) for input data Xi ϵ R^n, with R kernels will be as below:

$F(Xi)= \sum_{j=1}^{R} \emptyset_j(Xi)O_j$ (53)

For N arbitrary distinct samples (Xi, Yi) where Xi = [Xi1, Xi2, ..., Xin]^T Xi ϵ R^n and Yi = [Yi1, Yi2,…,Yic]^T ϵ R^c, RBF can be mathematically modeled as follows:

$\sum_{j=1}^{R} \emptyset_j(Xi)O_j$=T^i, i=l,...,N (54)

These N samples can be approximated with zero error using standard RBF and R kernels, as in the SLFN case. That means $Ti - Yi = 0$ for i = l, …., N

$H\ O = HH^T (\frac{I}{C} + HH^T)^{-1}\ T$ (55)

where $\Omega = HH^T$ (56)

The kernel matrix was defined using a kernel function K, where Ω (i,j) = h(Xi)h(Xj) = K(Xi, Xj). Replacing HH^T (9) with the kernel function, the solution can then be represented as below:

$$F(Xi)= \begin{bmatrix} K(X_i, X_1) \\ K(X_i, X_2) \\ \vdots \\ K(X_i, X_N) \end{bmatrix} \left(\frac{I}{C} + \Omega\right)^{-1}\ T \quad (57)$$

3.3. Multilayer Extreme Learning Machines

Let Xi = [*xi1*…, *xin*], where *xki* is the ith data representation of *xk* for k = 1 to n. Let Θi = [*ɣi1*, …, *ɣin*] be the i th transformation matrix, where *ɣki*, is the transformation vector used for representation learning concerning *xki*.

Then, Θi is learned by, replacing O with Θi, and Y with Xi in (51):

$H^i\ \Theta^i = X^i$ (58)

The output matrix of the ith hidden layer denoted as H^i, can be obtained concerning X^i using a similar approach as described in Equation (52):

$\Theta^i = H^i\ (\frac{I}{C} + H^i\ H^{iT})^{-1}\ X^i$ (59)

In Equation (59) , Θi is utilized for representation learning, and its learning process involves multiplying X^i with Θ^i:

$X^{final} = f^N(X^N\ (\Theta^N)^T)$ (60)

The final data representation of X, denoted as X^{final}, serves as the output of the hidden layer and is used to compute the output weight β in a similar manner as described in Equation (58):

$$X^{final} O = T \quad (61)$$

and O is calculated by (59)

$$O = X^{final}\left(\frac{I}{C} + X^{final}(X^{final})^{T}\right)^{-1} T \quad (62)$$

AE-KELM Algorithm
Input: *input matrix* $\boldsymbol{X}^{i-1}$*, regularization* C_{i-1}*, kernel parameters of* σ_{i-1}*, and activation* f'_{i-1} ***Output:*** X^{i} , Ω^{i}, Θ^{i} ***Method :*** *a.* $X^{i} = f'_{i-1}(X^{i-1}(\Theta^{i-1})^{T})$ *b.* $\Omega^{i}, = k(x_j,x_k,\sigma_{i-1})$ *c.* $\Theta^{i} = (\frac{I}{C^{i-1}}+\Omega^{i-1})^{-1} X^{i-1}$

3.4. S-AE-KELM-DA

Our study proposes a stacked autoencoder kernel ELM optimized by the dandelion algorithm (S-AEKELM-DA). This study aims to solve the specified machine learning objective by splitting an extensive network into multiple AE-KELM. The various AE-KELMs are stacked on different levels and connected serially. While C and σ are, respectively, the regularization factor and the kernel parameter of AE-KELM, they are randomly generated and optimized in the training phase without considering hidden nodes.

In the SAE-KELM approach, every AE-KELM module is responsible for learning the data transformation from the hidden layer to the output layer, as illustrated in Figure 27 while our architecture contributes to the target output for each level. The input matrix, Xi, is transformed into a kernel matrix, Ωi, using a kernel function. In AE-KELM, the kernel function commonly employed is the Gaussian radial basis function (RBF), expressed as follows:

$$K(Xi, Xj) = \exp(-\left\|X_i - X_j\right\|^2 / 2\sigma^2) \quad (63)$$

where σ is the kernel parameter.

The transformation matrix Θ^i, denoting the ith matrix in AE-KELM, can be learned similarly to ML-ELM (multi-layer extreme learning machine) in (58):

$$\Omega^i \Theta^i = X^i \quad (64)$$

The final step of the transformation procedure involves obtaining the data representation X^{i+1}, which is achieved similarly in Equation (60):

$$X^{i+1} = f'(X^i (\Theta^i)^T) \quad (65)$$

The activation function, denoted as f', can be chosen arbitrarily. In the case where the ith layer has the same dimension as the $(i + 1)^{th}$ layer, a linear piecewise activation function can be selected. In ML-KELM, all Θi have the dimension of $n \times n$ (a square matrix). As a result, the linear piecewise activation function can be applied to all Θ^i except for Θ^1.

Therefore, it is possible to merge the transformation matrices Θ^i for $i > 1$ into a single matrix Θunified $(i > 1)$ without compromising the ability to represent the data accurately. As a result, all transformations can be represented using just two matrices. A detailed explanation of the S-AEKELM-DA approach can be found in the stacked AE-KELM algorithm.

In the stage of representation learning, each pair of Θ^i and X^i can be calculated using (60) and (61), and the X^{final} is calculated as input to train a KELM model as follows:

$$\Omega^{final} O = T \quad (66)$$

and β is computed by (65)

$$O = (IC+\Omega^{final})^{-1} T \quad (67)$$

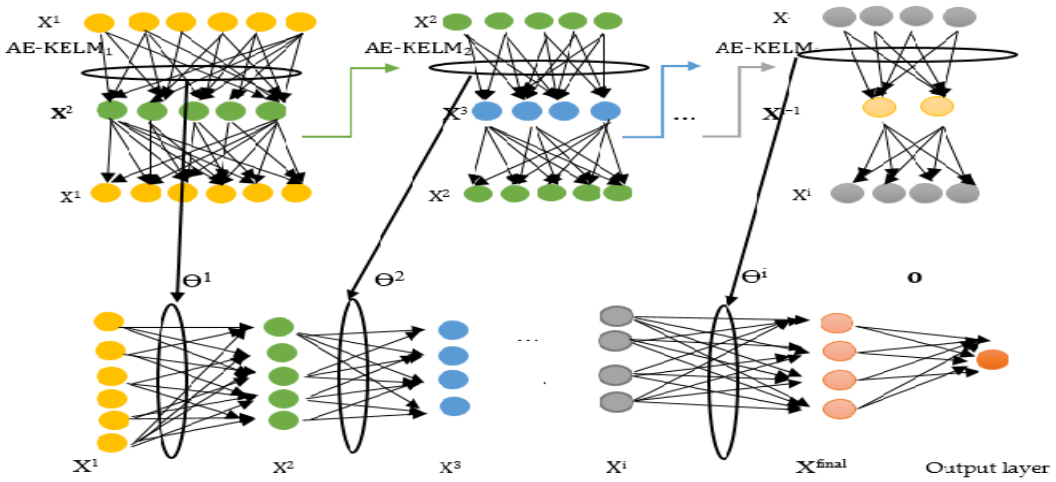


Figure27. Stacked AE-KELM

Stacked AE-KELM Pseudo code Algorithm
Input: *input matrix* $\boldsymbol{X}^1$*, output vector* $\boldsymbol{T}$*, regularization* C_i*, kernel parameters of* σ_i *and the number of layers L.* ***Output:*** $\boldsymbol{X}^{final}$ *matrix and output weight O* ***Method:*** *a. Compute the first kernel matrix* $\Omega^1_{(i,k)} = k(x_j,x_k,\sigma_1)$ *for j, k = 1 to N* *b. Calculate* $\Theta^1 = (I/C^1+\Omega^1)^{-1}X^1$ *c. Calculate new data representation* $X^2 = f(X^1(\Theta^1)^T)$ *d. Compute the second kernel matrix* $\Omega^2_{(i,k)} = k(x_j,x_k,\sigma_2)$ *e. Calculate* $\Theta^2 = (I/C^2+\Omega^2)^{-1}\ X^2$ *f.* $\Theta^{unified} = \Theta^2$ *g. for i = 2: L do* *h.* $X^i, \Omega^i, \Theta^i = AE\text{-}KELM\ (X^{i-1}, \Omega^{i-1}, \Theta{-}1^i)$ *i. Update* $\Theta^{unified} = \Theta^i\Theta^{unified}$ *j.* $X^{final} = X^N$ *k. Calculate the Nth kernel matrix* $\Omega^N_{(i,k)} = k(x_j,x_k,\sigma_N)$ *l. Calculate the output weight O =* $(I/C^N+\Omega^N)^{-1}T$

The S-AEKELM-DA learning network is made up of numerous small AE-KELM at various levels. The outputs from the hidden layers of each small AE-KELM are transmitted to the following layer via serial connections. Here, only the "optimized" kernel hidden layer outputs are passed to the following layer rather than the complete kernel hidden layer outputs. Therefore, the S-AEKELM-DA-based method can be explained as follows.

The first step in Figure 28 consists of using DA to choose the optimal C and σ in each label of the complete network. The training process of our proposed method in Figure 28 serially connects various AE-KELMs, so the optimal output weight of the first AE-KELM1 passes to the second AE-KELM2 and so on until the optimal β with the best metrics is found.

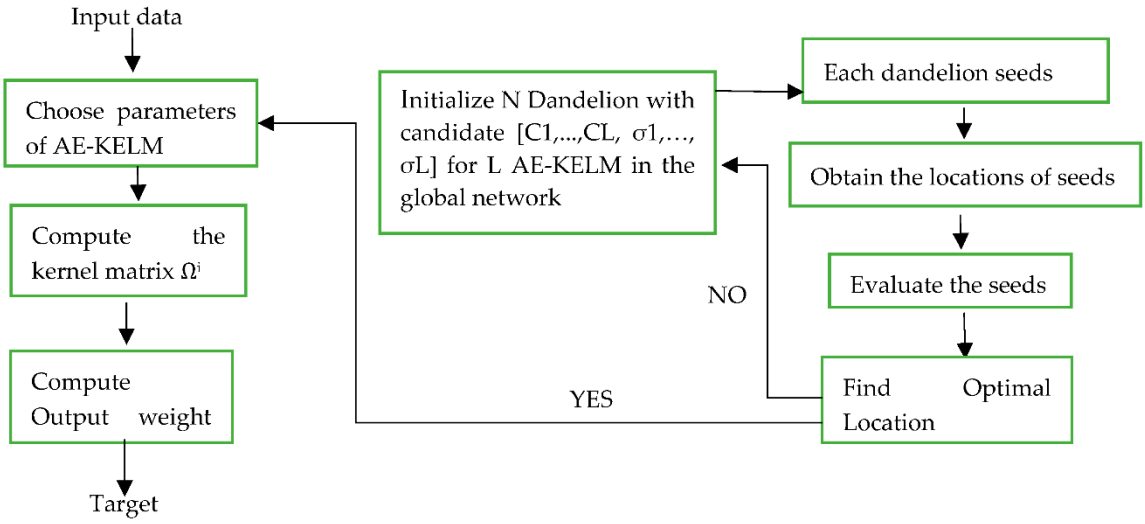


Figure28. Flow chart of optimized Stacked AE-KELM with DA.

As a result, a theoretically ideal reconstruction of Xi is obtained (i.e., Xi maintains all information of its previous data representation Xi−1), which explains why KELM-AE learns a superior representation for Xi. Ci is introduced to Θi in (33) to minimize overfitting, which leads to greater generalization (Guang-bin Huang & Siew, 2014). As a result of stacking numerous KELM-AEs, S-AEKELM-DA can learn a better data representation (i.e., fewer reconstruction errors are collected and transferred to the final layer). As a result, the generalization model is improved.

Figure 28 shows the different steps of our method. First, we initialize our population with N dandelion that is chosen randomly. Each dandelion has 2L parameters to optimize, while L is the number of AE-KELM in the global network, passing with the different steps mentioned in Section 3.1.

After obtaining the best dandelion, we can compute the kernel matrix Ωi for each AE-KELM to find the optimal output weight O and the target that finally is designated if the input data (transaction) is normal or not.

4. Evaluation of Optimized ELM Model Performance

4.1.Datasets

To check the effectiveness of our suggested algorithms, we select four publicly accessible datasets. They are Australian, German numerical, loan prediction, and default of credit card clients. Table 8 briefly shows the different information of each dataset.

Table.8 Dataset information.

Dataset	**Abbr.**	**#S**	**#NS**	**#FS**	**#NA**	**#RFS**
Australian	D1	690	307	383	14	55%
German numerical	D2	1000	700	300	21	30%
Loan prediction	D3	614	422	192	12	31%
Default of credit card clients	D4	30,000	23,364	6636	24	22%

Within Table 8, the column labeled “Abbr.” displays the designated codes assigned to the datasets. The “#S” entry denotes the total number of samples, while “#NS” represents the count of normal samples, and “#FS” indicates the number of fraudulent samples. Furthermore, “#NA” signifies the number of attributes present in the samples, and “#RFS” represents the ratio of fraudulent samples within a dataset.

4.2. Experiment results

For the D1 to D4 datasets in Figure 3, the results depict the obtained values with different activation functions, namely, rectified linear unit (Relu), radial basis function (RBF), sigmoid (Sig), and sine (Sin). According to the findings, ELM with RBF kernel function outperforms other models with various numbers of neurons. ELMs with RBF kernel consist of two user-specified parameters, the constant C and the kernel parameter σ. Also, σ is searched in a wide range $\{2^{-20}, 2^{-9}, ..., 2^{20}\}$ and C on $\{2^{-10}, 2^{-9}, ..., 2^{30}\}$.

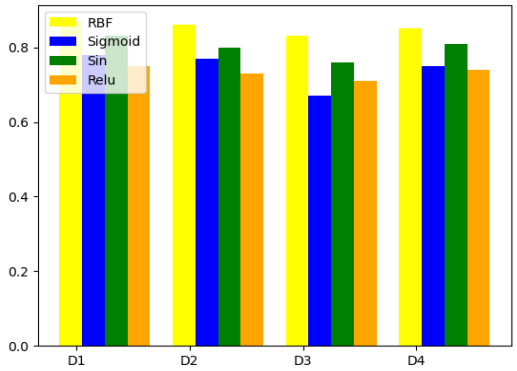
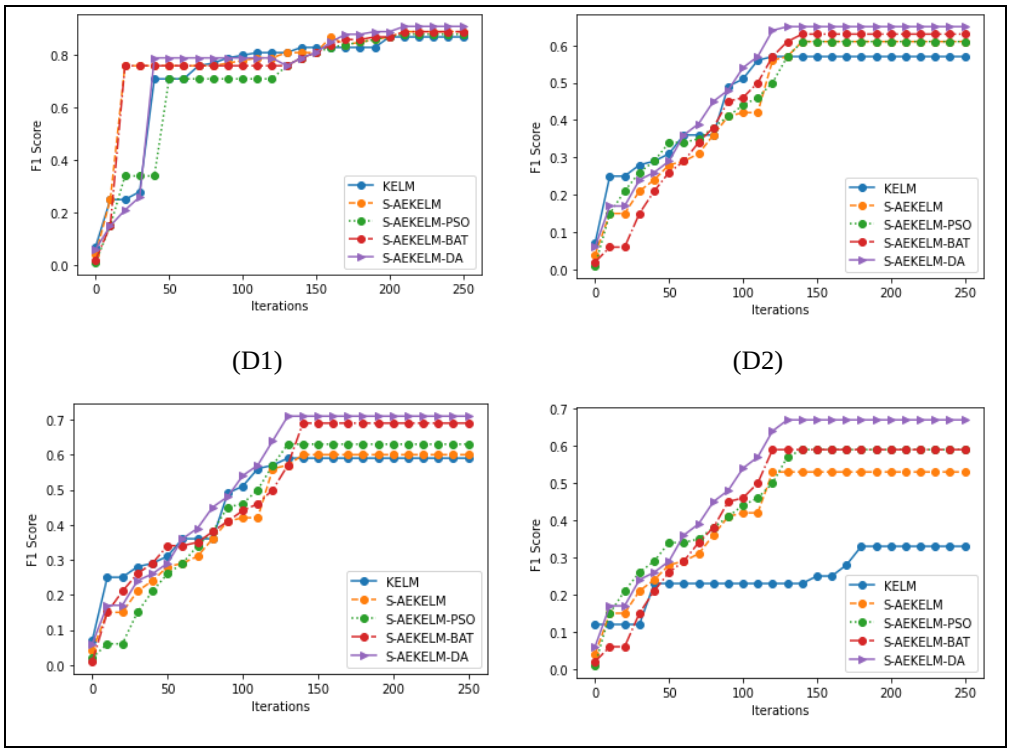


Figure29. Training ELM accuracy rates with on datasets various activation functions

The training classification performance on the fourth dataset with KELM, S-AEKELM, S-AEKELM-PSO, S-AEKELM-BAT, and S-AEKELM-DA with an F1-score is shown in Figure 27.



(D3)	(D4)

Figure30. Performance of KELM, S-AEKELM, S-AEKELM-PSO, S-AEKELM-BAT, and S-AEKELM-DA on fourth datasets (F1-Score).

Figure 31 shows that the best performance for S-AEKELM-DA is achieved when the number of AE-KELM components included in the model is greater than 10. Adding more than 10 AE-KELM components improves the training accuracy and potentially enhances the model's ability to capture complex patterns and relationships in the data.

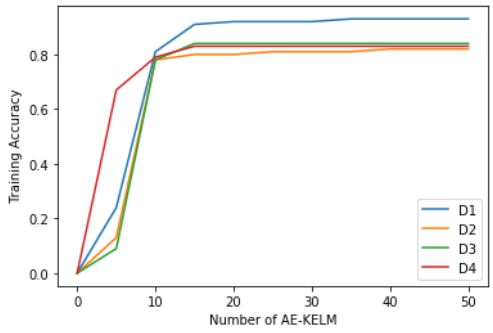


Figure31. Training the accuracy of S-AEKELM-DA in proportion to the number of AE-KELM.

The results tabulated in Table 9 represent the results obtained by the traditional KELM, S-AEKELM, S-AEKELM-PSO, S-AEKELM-BA, and S-AEKELM-DA. by applying the respective evolutionary approaches PSO, BA, and DA. From the results, it is observed that stacked-AEKELM improves classification performances as compared to KELM. It also performs better while integrating the cited evolutionary approaches, wherever S-AEKELM-DA proves exceptional results. Additionally, Table 4 shows the time cost of the AE-KELM using different evolutionary algorithms: PSO, BAT, and DA. The time cost of a model depends on multiple factors, including the size of the dataset, the correlation between its features, and the model architecture itself. In the case of the S-AEKELM-DA approach, including multiple hidden layers in the stacked AE-KELM contributes to a slightly higher time cost than other methods. However, it is essential to note that despite the high time cost, the S-AEKELM-DA approach demonstrates superior performance when compared to other methods such as KELM, S-AEKELM, S-AEKELM-PSO, and BAT-KELM.

Table.9 Computing results of dataset.

Dataset	**Metric**	**KELM**	**S-AEKELM**	**S-AEKELM-PSO**	**S-AEKELM-BA**	**S-AEKELM-DA**
D1	Acc	0.8732	0.8911	0.8889	0.8997	0.9121
	Prec	0.8747	0.8878	0.8720	0.8861	0.8945

	Recall	0.8808	0.8923	0.9047	0.9216	0.9409
	AUC	0.9129	0.9263	0.9221	0.9314	0.9568
	F1-score	0.8777	0.89	0.8873	0.9035	0.9171
D2	Acc	0.7811	0.7934	0.7916	0.8102	0.8217
	Prec	0.7331	0.7524	0.7603	0.78	0.8023
	Recall	0.9216	0.9335	0.9321	0.9347	0.9461
	AUC	0.8817	0.8864	0.8941	0.9008	0.9031
	F1-score	0.8166	0.8332	0.8374	0.8503	0.8682
D3	Acc	0.7602	0.8142	0.8211	0.8245	0.8447
	Prec	0.7809	0.7634	0.8444	0.8532	0.8602
	Recall	0.4618	0.5057	0.5132	0.5122	0.6118
	AUC	0.7913	0.7942	0.801	0.8015	0.8227
	F1-score	0.5803	0.6083	0.6384	0.6401	0.715
D4	Acc	0.7221	0.7548	0.7633	0.772	0.8322
	Prec	0.481	0.5451	0.5821	0.6016	0.753
	Recall	0.3341	0.6036	0.6009	0.5964	0.6164
	AUC	0.7011	0.7166	0.7207	0.7241	0.7633
	F1-score	0.3943	0.5729	0.5914	0.599	0.6779

Table.10 Time cost (seconds) of testing different algorithms.

	KELM	**S-AEKELM**	**S-AEKELM-PSO**	**S-AEKELM-BAT**	**S-AEKELM-DA**
D1	25	34	37	34	38
D2	46	51	59	52	60
D3	29	33	35	32	39
D4	1417	1808	1759	1770	1783

Table 10 displays the time costs in seconds for testing different algorithms on four datasets (D1, D2, D3, and D4). Each column represents a different algorithm, and each row represents a different dataset. While there might be slight variations in the time costs, it appears that the three algorithms, S-AEKELM-PSO, S-AEKELM-BAT, and S-AEKELM-DA, have similar time costs for testing across the different datasets.

Table.11 Parameter settings.

Algorithm	Parameters
PSO	N = 50, c_1 = 1.5, c_2 = 1.5, w = 0.7311
BA	N = 50, A = 1, R = 1, $\alpha = \gamma = 0.9$
DA	N = 50, X = 20, max = 100

N is the population size in Table 11, and the parameters are introduced as follows. PSO: w stands for the inertia weight, and c1 and c2 are acceleration factors. BA: R is the pulse emission rate, A is loudness, and α and ξ are constants. DA: max is the number of seeds, and X is the number of the best dandelions.

4.3. Comparative results

The average of 10-CV represented the experimental findings, and each fold was repeated 20 times. All algorithms and these simulations were implemented on Python 3 (Jupyter Notebook) on a PC with a 2.5 GHz CPU (Intel Core i5-7200), 12 GB RAM, and Windows (64-bit).

In Table 12 our study compared the performance of our proposed method with several existing machine learning and ensemble learning techniques on various datasets. The results consistently demonstrated that our method outperformed the other approaches regarding accuracy, recall, precision, F1-score, and AUC.

Table.12 Comparing results with previous work.

Dataset	Reference	Method	Accuracy	Recall	Precision	F1-Score	AUC
Australian	[48]	BPN	0.8683	-	-	-	-
		GP	0.87	-	-	-	-
		C4.5	0.859	-	-	-	-
		SVM + GA	0.869	-	-	-	-
	[10]	ANN	0.94	-	-	-	0.93
		KNN	0.836	-	-	-	0.893
		SVM-L	0.874	-	-	-	0.923
		SVM-R	0.861	-	-	-	0.929
		CART	0.859	-	-	-	0.894
		J48	0.845	-	-	-	0.881
		LR-R	0.862	-	-	-	0.940

	[49]	XGBoost	0.8633	0.8487	0.8449	0.8468	0.9394
		LightGBM	0.8624	0.8422	0.8476	0.8449	0.9371
		AugBoost-RP	0.8633	0.8487	0.8449	0.8468	0.9416
		AugBoost-PCA	0.8681	0.8497	0.8533	0.8515	0.9415
		AugBoost-NN	0.8645	0.8539	0.8435	0.8487	0.9424
		AugBoost-ELM	0.8635	0.8462	0.8485	0.8462	0.9422
		Our study	**0. 9121**	**0.9409**	0.8945	**0.9171**	**0.9568**
	[48]	BPN	0.7783	-	-	-	-
		GP	0.781	-	-	-	-
		C4.5	0.736	-	-	-	-
		SVM + GA	0.7792	-	-	-	-
	[10]	ANN	72.80	-	-	-	75.60
		KNN	66.90	-	-	-	70.20
		SVM-L	74.80	-	-	-	79.30
		SVM-R	75.90	-	-	-	80.20
		CART	55.90	-	-	-	64.30
German numerical		J48	64.10	-	-	-	65.20
		LR-R	75.40	-	-	-	78.50
	[49]	XGBoost	0.7582	0.9208	0.7757	0.842	0.7811
		LightGBM	0.7615	0.9007	0.7887	0.841	0.7776
		AugBoost-RP	0.7519	0.8867	0.7862	0.8335	0.7707
		AugBoost-PCA	0.7605	0.8852	0.7956	0.838	0.7735
		AugBoost-NN	0.7604	0.9237	0.7766	0.8438	0.7843
		AugBoost-ELM	0.7617	0.9245	0.7775	0.8446	0.7861
		Our study	**0.8217**	**0.9461**	**0.8023**	**0.8682**	**0.9031**
	[16]	WELML	0.7883	0.5782	0.6951	0.6285	0.7311
Loan prediction		WELMB	0.7834	0.5834	0.6792	0.6244	0.729

		WELME	0.7867	0.5837	0.7016	0.6263	0.7317
	[50]	KNN	0.837	-	-	-	-
		SVM	0.831	-	-	-	-
		Decision Tree	0.831	-	-	-	-
		Logistic Regres	0.824	-	-	-	-
		RF	0.798	-	-	-	-
		Our study	**0.8447**	**0.6118**	**0.8602**	**0.715**	**0.8227**
	[16]	WELML	0.7687	0.6032	0.5407	0.5683	0.7139
		WELMB	0.7657	0.6003	0.5359	0.5645	0.7110
		WELME	0.7679	0.5973	0.5391	0.5649	0.7009
Default of credit card clients dataset	[51]	Logistic Regres	0.8139	-	-	-	-
		Decision Tree	0.8218	-	-	-	-
		Boosting	0.8235	-	-	-	-
		RF	0.8212	-s	-	-	-
		Our study	**0.8322**	**0.6164**	**0.753**	**0.6779**	**0.7633**

Our study compared the performance of our proposed method with several existing machine learning and ensemble learning techniques on various datasets. The results consistently demonstrated that our method outperformed the other approaches regarding accuracy, recall, precision, F1-score, and AUC.

When comparing our results with machine learning methods such as BPN, GP, C4.5, SVM, KNN, and logistic regression, our method achieved higher accuracy and improved performance across all datasets. Results indicate that our approach is more effective at accurately predicting outcomes and producing reliable results.

Furthermore, our method showcased superior performance when compared to ensemble learning techniques like XGBoost, LightGBM, and AugBoost. It consistently outperformed these methods regarding accuracy, recall, precision, F1-score, and AUC, thus indicating its superiority in ensemble-based prediction tasks.

These findings highlight our proposed method's advantages over traditional machine learning and ensemble learning techniques. Our method offers a more accurate and robust solution for

various prediction and classification tasks, making it a promising approach for real-world applications.

5. Conclusion

This chapter proposed a method called S-AEKELM-DA for credit card fraud detection. This study aimed to solve the specified machine learning objective by splitting an extensive network into multiple AE-KELM. The various AE-KELMs are stacked on different levels and connected serially. While C and σ are, respectively, the regularization factor and the kernel parameter of AE-KELM, they are randomly generated and optimized in the training phase without considering hidden nodes. Several tests were conducted using four credit card fraud detection datasets to evaluate its effectiveness. The simulation results indicated that the suggested architecture gives the best findings. Furthermore, compared to KELM, S-AE-KELM, S-AE-KELM-PSO, and S-AE-KELM-BAT, S-AE-KELM-DA demonstrated higher accuracy, recall, precision, F1-score, and AUC performance criteria. Notably, S-AEKELM-DA achieved this performance advantage in a few seconds during both model training and testing phases. Thus, the proposed method was highlighted as a superior approach to detecting credit card fraud on four publicly accessible datasets with different sizes and different percentages of class imbalance. The suggested method demonstrated the effectiveness of our approach, showcasing its potential for addressing challenges in imbalanced datasets and achieving better performance compared to the literature methods. Eliminating manual tuning and optimizing the kernel function activation parameters can significantly improve the model's ability to capture complex patterns and relationships in the data.

The integration of data imbalance resolution techniques and model optimization strategies represents a significant advancement in the field of credit card fraud detection. By addressing the challenges posed by imbalanced datasets while simultaneously optimizing model performance, the next chapter offers a comprehensive solution for enhancing the accuracy and reliability of fraud detection systems. Building upon the insights gained from our integrated framework, the subsequent chapter will delve into practical implementations and case studies, shedding light on the real-world efficacy of this approach and paving the way for future advancements in fraud detection methodologies

Chapter IV: Integrating Data Imbalance Resolution and Model Optimization for Credit Card Fraud Detection

1. Introduction

Artificial neural networks (ANNs) are powerful machine learning models widely used in many fields because they handle nonlinear problems and map complex input data to output classes or values. ANNs can have multiple hidden layers, which allow them to learn increasingly complex features and hierarchies of information. However, traditional ANNs can suffer from overfitting, making generalizing new, unseen data challenging. Additionally, training ANNs can be computationally expensive, requiring large amounts of data and computing resources.
Moreover, traditional networks also need help with generalization and slow learning (Ma et al., 2022).
Overcoming ANN limitations, one feedforward hidden layer neural network called ELM was adopted (Avenue, 2004). With little human assistance, it is a non-iterative approach, and compared to other neural network methods, ELM is special (Guang-Bin Huang, 2004)(G.-B. Huang et al., 2011)(Gao Huang et al., 2019). In contrast to the backpropagation approach, it does not update a weight iteratively. However, it randomly attributes input weights and biases without being adjusted during training and then automatically determines the output weights, which is why ELM learns rapidly. Input weights are the link between input and hidden neurons, and the connections between the hidden layer and output layer neurons are known as output weights. Based on benchmarking classification and regression issues (Yahia et al., 2020)(Sun et al., 2016)(Tang, J.; Deng, C.;Huang, 2016)(El Hlouli et al., 2020) showed that in comparison with feedforward network-based gradient-descent, the testing and validating part of many applications may now be finished quickly because the ELM algorithm's attempt to offer high generalization performance in a very short time. Therefore ELM architecture has attracted the interest of many researchers (Ma et al., 2022)(Kardani et al., 2021)(Kardani et al., 2022)(Li et al., 2022), and superior performance has been shown via continuous development and improvement in the following areas: image classification (Lu et al., 2019)(Cao et al., 2013), text classification (Roul, n.d.)(Conference & Technologies, 2016), and bioinformatics (Shreya, n.d.)(Z. Wang et al., 2014). The input weights and biases are generated at random, This directly affects how unstable the ELM model is. Huang et al. (Guang Bin Huang & Slew, 2004) offer kernel ELM (KELM), an expanded version of ELM. The connection weights between the

hidden and input layers are discovered to be unneeded. KELM outperforms ELM in terms of performance and training speed. Because of its advantages, it has been effectively used to address various real-world issues. However, ELM and KELM could be more effective for problems with unbalanced classification (Gong et al., 2018)(Shariati et al., 2022). Additionally, previous work has some limitations if the original dataset is very complicated or highly imbalanced.

To balance the data distribution (L. Wang et al., 2021), two basic sampling methods are undersampling and oversampling, which replicate the minority samples and eliminate some of the samples from the majority. Using the oversampling approach presents a problem in separating

meaningful samples from redundant ones. Each sample is given additional weight to increase the relative influence of the minority class while decreasing the relative impact of the majority class to reduce the performance bias brought on by an unbalanced class distribution. Weighed ELM (WELM) is suggested (Zong et al., 2013). It gives various sample kinds distinct weights. To handle binary classification problems, which are resilient to balanced and unbalanced data distributions, WELM is implemented in this study based on KELM. Similar to KELM, parameter optimization for WELM is a hot topic since its two parameters, the bandwidth of RBF (radial basis function) kernel σ and regularization coefficient C, are essential to its functionality.

According to (Abualigah et al., 2022), the reptile search algorithm (RSA) is a natural metaheuristic optimizer influenced by crocodiles' hunting techniques. Numerous classification problems have been effectively optimized using RSA due to its strong performance and ease of use, which has caught the interest of many academics.

The advantages offered by the proposed WB-ELM-RSA are as follows:

- WB-ELM-RSA offers a resolution to the challenges presented by imbalanced binary classification datasets.
- Comparative evaluations reveal the superior performance of the proposed method when contrasted with alternative intelligent optimization algorithms, including Particle Swarm Optimization, Bat Algorithm, and Dandelion Algorithm.
- WB-ELM-RSA has improved real-time detection performance when used with datasets for credit card fraud detection.
- The proposed approach introduces a more expedient and efficient optimization mechanism for the two essential parameters of WB-ELM. This quality positions it as a promising and viable solution for addressing imbalanced binary classification tasks.

Our paper is structured into four main sections. In Section 2, we provide a brief introduction to the Weighted Extreme Learning Machine (WELM) and the Reptile Search Algorithm (RSA) optimization. Section 3 offers a detailed explanation of the proposed approach, which involves the optimization of a WELM for binary classification using the RSA algorithm. The simulation experiments and results obtained using the credit card fraud datasets are presented in Section 4. Finally, Section 5 summarizes the conclusions drawn from this study.

2. Related Works

2.1. Reptile Search Algorithm

The Reptile Search Algorithm (RSA) is a meta-learning algorithm that can be used to optimize the parameters of any machine-learning algorithm (Abualigah et al., 2022). The best-obtained solution is regarded as approaching the optimum in each iteration of the optimization process, which starts with a collection of candidate solutions (A), as stated in Eq. (68), which is created stochastically.

$$A=\begin{bmatrix} a11 & \cdots & a1n \\ \vdots & \ddots & \vdots \\ aE1 & \cdots & aEn \end{bmatrix}_{E*n} \quad (68)$$

aij denotes to the jth position of the i_{th} solution, E is the number of candidate solutions, and n denotes to the dimension size of the given problem.

$$aij = rand \times (U - L) + L,\ j = 1, 2, ..., n \quad (69)$$

L and U stand for the lower and upper bounds of the given issue, respectively, where a random variable named rand. The position updating solutions for the exploration phase are provided in this study as in Eq(70).

$$aij(t+1) = \begin{cases} best_j(t)^* - \Gamma_{ij}(t)^*\beta - S_{ij}(t)^* rand, & t \le T/4 \\ best_j(t)^* a_{s1}j * ES(t) * rand, & T/4 < t \le 2T/4 \end{cases} \quad (70)$$

Where $best_j(t)$ is the jth position in the best-obtained solution so far, T is the maximum number of iterations, t is the current iteration number, and rand stands for a random number between 0 and 1.

Eq(71) establishes the hunting operator for the jth point in the i_{th} solution, indicated by the symbol Γ_{ij}. A sensitive parameter β affects the exploration accuracy (high walking) during the encircling phase during iterations with a fixed value of 0.1. The search region is reduced by utilizing the reduction function (S_{ij}) derived from Eq(72). S_1 refers to a random point of the i_{th} solution and is an integer between [1 E] and a (s_{1j}). The number of potential answers is E. The

Evolutionary sense ($ES(t)$), which is determined by Eq(73), is a probability ratio that randomly decreases values between 2 and -2 throughout the number of repetitions.

$$\Gamma_{ij}(t) = best_j(t) * P_{ij}, \quad (71)$$

$$S_{ij} = \frac{best_j(t) - a_{s2j}}{best_j(t) + \epsilon}, \quad (72)$$

$$ES(t) = 2 * s3 * \left(1 - \frac{1}{T}\right), \quad (73)$$

Where s2 represents a random number between [1 E] and ϵ a small value. In Eq. (73), two is utilized as a correlation value to give values between 2 and 0, while s3 represents a random integer number between 1 and 1. The difference in percentage between the jth place of the best solution and the jth position of the present solution, or P_{ij}, is determined by applying Eq. (74) to the problem.

$$P_{ij} = \alpha + \frac{aij - M(ai)}{bestj(t) * (Uj - Lj) + \epsilon}, \quad (74)$$

$M(a_i)$, as in Eq. (74), denotes the average positions of the i_{th} solution and is determined using Eq. (75). The upper and lower limits of the jth position are, respectively, Uj and Lj. α is held constant in this study at 0.1.

$$M(a_i) = \frac{1}{n}\sum_{j=1}^{n} aij, \quad (75)$$

The position updating equations are shown below for the exploitation phase in Eq. (76).

$$aij(t+1) = \begin{cases} best_j(t) * P_{ij}(t) * rand, & 2T/4 < t \leq 3T/4 \\ best_j(t) - \Gamma_{ij}(t) * \epsilon - S_{ij}(t) * rand, & 3T/4 < t \leq T \end{cases} \quad (76)$$

Where bestj(t) represents the jth position in the best solution thus far, The hunting operator for the jth position in the i_{th} solution is denoted by the symbol Γij, which is determined by Eq(71). P_{ij} is the ratio of the jth place of the best-obtained solution to the jth position of the present solution, and it is derived using Eq(74). ϵ a minor value. S is a number used to reduce the search region and is calculated using Eq(72).

To generate a random value at each iteration and continue exploring both during the first and last iterations, we carefully set two parameters (i.e., β and α).

2.2. Other optimization algorithms

The bat algorithm is a metaheuristic optimization method researchers have discovered using these bat-like characteristics (X. S. Yang, 2010). The two main characteristics of bats, decreasing loudness and increasing the rate of emitted ultrasonic sound when chasing prey, are

adopted in designing the BA. In the BA, each artificial bat within the population has several properties, including a position vector, a velocity vector, and a frequency vector. These properties are updated iteratively during the algorithm to guide the search for optimal or near-optimal solutions. BA aims to balance exploration and exploitation to explore the search space and find suitable solutions effectively.

Particle Swarm Optimization (PSO) (Eberhart & Kennedy, 1995) is a metaheuristic optimization algorithm widely used across various fields to solve optimization problems. In PSO, the optimization problem is tackled by simulating the movement of particles in a search space. Initially, the particles are distributed randomly throughout the search space. During each iteration, the particles update their positions and velocities based on their own best position (Pb) and the best position found by their neighboring particles (Pg). The velocity of a particle determines its direction and speed of movement, while the position corresponds to a potential solution to the problem. PSO combines exploration and exploitation to guide the search for optimal solutions. It is widely used for various optimization problems.

The DA was created due to studying dandelion sowing behavior (H. Zhu et al., 2020), and proposed for solving continuous optimization problems. Dandelion populations in DA are split into two subpopulations: those suitable for sowing and those not. Then, various sowing techniques are used for the various subpopulations. In the meantime, one more method of sowing is to conduct the suitable subpopulation to avoid slipping into the local optimum. If dandelion is sown, its seeds will be dispersed all around it. Dandelion sowing can be viewed as a search for an ideal in a specific area surrounding a point. The algorithm employs seed dispersal, seedling competition, and wind drift mechanisms to guide the search process. It aims to balance exploration and exploitation to find optimal or near-optimal solutions.

While the Dandelion Algorithm, PSO, and BA have distinct inspirations and design principles, they also have differences in their specific mechanisms and characteristics. Each algorithm employs unique strategies and techniques to explore and exploit the search space. They may differ regarding the update rules for positions and velocities, information exchange or dispersal mechanisms, and exploration-exploitation strategies.

2.3. Weighted Extreme Learning Machine

A balanced dataset is classified using ELM. Hence an imbalanced dataset is more suitable to classify using weighted ELM (Zong et al., 2013)(El Hlouli et al., 2020)(Ding et al., 2014)(H. Zhu et al., 2020). Weighted Extreme Learning Machine (ELM) is an extension of the traditional ELM algorithm that incorporates sample weights during training. In weighted ELM, each

sample in the training set is assigned a weight that reflects its importance or significance. These weights can be used to adjust each sample's contribution to the ELM model's training. With N training data, L hidden layer neurons, input samples X, and target matrix Y. Xi ϵ Rn, yi ϵ Rm, f is an activation function, P is the input weight, b is the single hidden layer bias, and O is an output weight. The output of ELM can be calculated as follows:

$$\sum_{j=1}^{L} O_j \,.\, f\left(p_j.X_i + b_j\right) = y_i \text{ . for i = \{1, 2, ……., N\}} \quad (77)$$

Which can be simply expressed as:

$$HL.O=T \quad (78)$$

Where $HL(p_1,p_2,\ldots, p_L, b_1,b_2,\ldots,b_L,X_l, X_2, \ldots, X_N) =$

$$\begin{bmatrix} f(p_1.X_1 + b_1) & \cdots & f(p_L.X_1 + b_L) \\ \vdots & \ddots & \vdots \\ f(p_1.X_N + b_1) & \cdots & f(p_L.X_N + b_L) \end{bmatrix}_{N*L} \quad (79)$$

The Karush-Kuhn-Tucker theory asserts (Guang Bin Huang et al., 2012), that the output weight O is computed as follows where C is the regularization factor.

$$O = HL^T\left(\frac{I}{C} + HL.HL^T\right)^{-1} T \quad (80)$$

In ELM, the kernel matrix was defined as $\Omega = HL.HL^T$ using a kernel function K, where $\Omega_{(i,j)} = hl(X_i)hl(X_j) = K(X_i, X_j)$. The kernel function of KELM is usually an RBF function and is typically Gaussian as follows:

$$K(X_i, X_j) = \exp(-\sigma\|Xi - Xj\|) \quad (14)$$

where σ is the kernel parameter.

Replacing $HLHL^T$ (13) with the kernel function, the output function of an ELM can then be represented as:

$$F(X_i)=\begin{bmatrix} K(X_i, X_1) \\ K(X_i, X_2) \\ \vdots \\ K(X_i, X_N) \end{bmatrix}\left(\frac{I}{C} + \Omega\right)^{-1} T = \begin{bmatrix} K(X_i, X_1) \\ K(X_i, X_2) \\ \vdots \\ K(X_i, X_N) \end{bmatrix}\left(\frac{I}{C} + P\Omega\right)^{-1} PT \quad (15)$$

Where P is a weight matrix. P has two weighting schemes (Zong et al., 2013):

$$p_{ii} = \frac{1}{\#(yi)} \quad (16)$$

$$p_{ii} = \begin{cases} \frac{0.618}{\#(yi)} & \text{, if } yi > yi^* \\ \frac{1}{\#(yi)} & \text{, otherwise} \end{cases} \quad (17)$$

Where #(yi) is the number of samples belonging to class yi . yi* is the average of the total samples of all classes.

3. Optimizing WB-ELM by RSA

Weighted Binary Extreme Learning Machine (WB-ELM) is a novel extension of the weighted ELM algorithm specifically designed for binary classification problems.

In the suggested algorithm, the weight matrix is crucial. In essence, it establishes how much rebalancing users desire and how far the limit should be pushed in favor of the dominant class. Users can specify Wii for each sample xi, making the suggested algorithm a member of the cost-sensitive learning family. We opt for an automated weighting system that is created from class data out of convenience, which is a unique instance of cost-sensitive learning:

$P = \text{diag}\{ p_{ii} \}$, i= 1, …,N (18)

While pii was defined in equation (16).

To apply RSA optimization to WB-ELM, we need a set of hyperparameters that will be optimized. These typically include the regularization parameter and σ the parameter of the RBF kernel activation function used in the hidden layer. We also need to specify the range of values that each hyperparameter can take.

Choose the candidate solution with the hyperparameters that resulted in the highest fitness function value.

The fitness value of each solution in the current set of candidates is defined as in Eq. (19) which is based on Root Mean Squared Error (RMSE) (Mohapatra et al., 2015), where Nv is the size of data samples.

$$\text{Fitness} = \sqrt{\frac{\sum_{j=1}^{Nv} \left\| F(Xj) - T_j \right\|_2^2}{Nv}} \quad (19)$$

Update hyperparameters using Eqs (4)(5)(7). This process is repeated for several iterations, allowing the meta-parameters to converge to values that optimize the performance of the WB-ELM model on the entire training dataset.

Update position solution using Eqs (7)(9), This process is repeated for several iterations until the criterion is satisfied.

Update the WB-ELM parameters for training the model on the training data, the output weight parameters are computed. The optimized WB-ELM model is then tested on a new testing data to evaluate its performance as shown is Figure 1.

By following this process, RSA can optimize the parameters of a WB-ELM model quickly. The key advantage of RSA is that it doesn't require a large amount of training data, as the model is trained on a variety of related parameters rather than a single parameter.

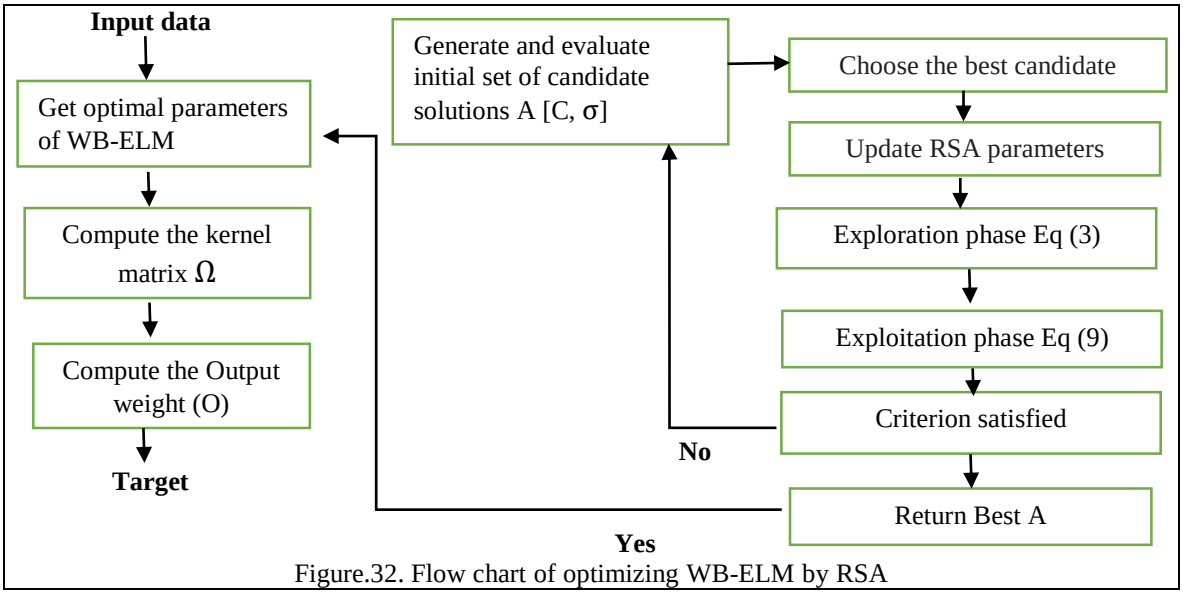


Figure.32. Flow chart of optimizing WB-ELM by RSA

WB-ELM-RSA Pseudo code

```
a.Initialize RSA and WB-ELM hyperparameters (α, β, etc.)
b.Define the range of values for each WB-ELM hyperparameter
c.Initialize candidate solutions
d.Set the number of iterations
e.for each iteration in the range of num_iterations do:
f.     for each candidate_solution in candidate_solutions do:
g.         Update hyperparameters of the candidate_solution using Eqs(71)
           (72)and(74)
h.         Update the position solution of the candidate_solution using Eqs
           (74) and (76)
i.     end for
j.     Calculate the fitness value for each candidate solution using Eq(16)
k.     Select the candidate solution with the highest fitness value as the
       best_candidate_solution
l.     Update WB-ELM parameters based on the best_candidate_solution
m.end for
n.Compute the output weight parameters for training the WB-ELM model on the
  training tasks
o.Test the WB-ELM model on the testing data to determine how well it
  performs
p.Return the optimized WB-ELM model parameters
q.End
```

4. Experiments and Comparative Results

The average of 10-CV represents the experimental findings, and each fold was repeated 20 times. This technique involves dividing the datasets into two global subsets, the first with nine subsets used for training, one for testing in each iteration, and the second subset for validating our model. All algorithms and these simulations are implemented on Python 3 (Jupyter Notebook) on a PC with a 2.5 GHz CPU (Intel Core i5-7200), 12 GB RAM, and Windows 7(64-bit).

4.1. Evaluation metrics

Various performance indicators were computed to evaluate the classification model on the datasets. True positives (TP) are instances that were correctly predicted to be positive; false negatives (FN) are instances that were correctly classified as positive; true negatives (TN) are instances that were correctly predicted to be negative; and false positives (FP) are instances that were correctly predicted to be negative.

Accuracy is the most current parameter used, and it is defined as the ratio of all correct transactions divided by the total number of predicted transactions (Itoo et al., 2020).

$$\text{Accuracy} = (T\_N+T\_P)/(T\_N+F\_N+F\_P+T\_P) \quad (20)$$

Although the dataset used is highly imbalanced, the accuracy alone will not be sufficient to calculate the performance of the model. Furthermore, other evaluation metrics were used to evaluate our model, such as:

Recall: fraudulent transactions divided by the number of all transactions that are positive instances.

$$\text{Recall(Sensitivity)} = T\_P/(T\_P+F\_N) \quad (21)$$

Precision: legitimate transactions divided by the number of all transactions that are predicted as positive instances.

$$\text{Precision (Specificity)} = T\_N/(T\_N+F\_P) \quad (22)$$

The G-mean combines both sensitivity and specificity, providing a balanced evaluation metric that considers the performance across both positive and negative classes.

$$\text{G-mean} = \sqrt{(\text{Recall}*\text{Precision})} \quad (23)$$

4.2. Dataset

To check the effectiveness of our suggested algorithms, we select four publicly accessible datasets. They are Australian, German numerical, Loan Prediction and Default of Credit Card Clients. Table 1 Briefly shows the different information of each dataset.

Table. 13. Datasets information

Dataset	Abbr	#S	#NS	#FS	#NA	#RFS
Creditcardcsvpresent	D1	3075	2627	448	11	14.55%
Default of Credit Card Clients	D2	30000	23364	6636	24	22%
German numerical	D3	1000	700	300	24	30%
Loan Prediction	D4	614	422	192	12	31%
Australian	D5	690	307	383	14	55%

Column Abbr in Table 1, lists the datasets' given identifiers; #S denotes the number of samples, #NS the number of normal samples, #FS the number of fraudulent samples, #NA the number of sample attributes, and #RFS the proportion of fraudulent samples in a dataset.ELMs with Gaussian Kernel, consist of two user-specified parameters, the constant C and the kernel parameter σ. Also, σ is searched in a wide range {2-20, 2-9, ..., 220} and C on {2-10, 2-9, ..., 230}.

To guarantee a fair comparison, the maximum number of evaluations for all evaluated algorithms is set at 20.

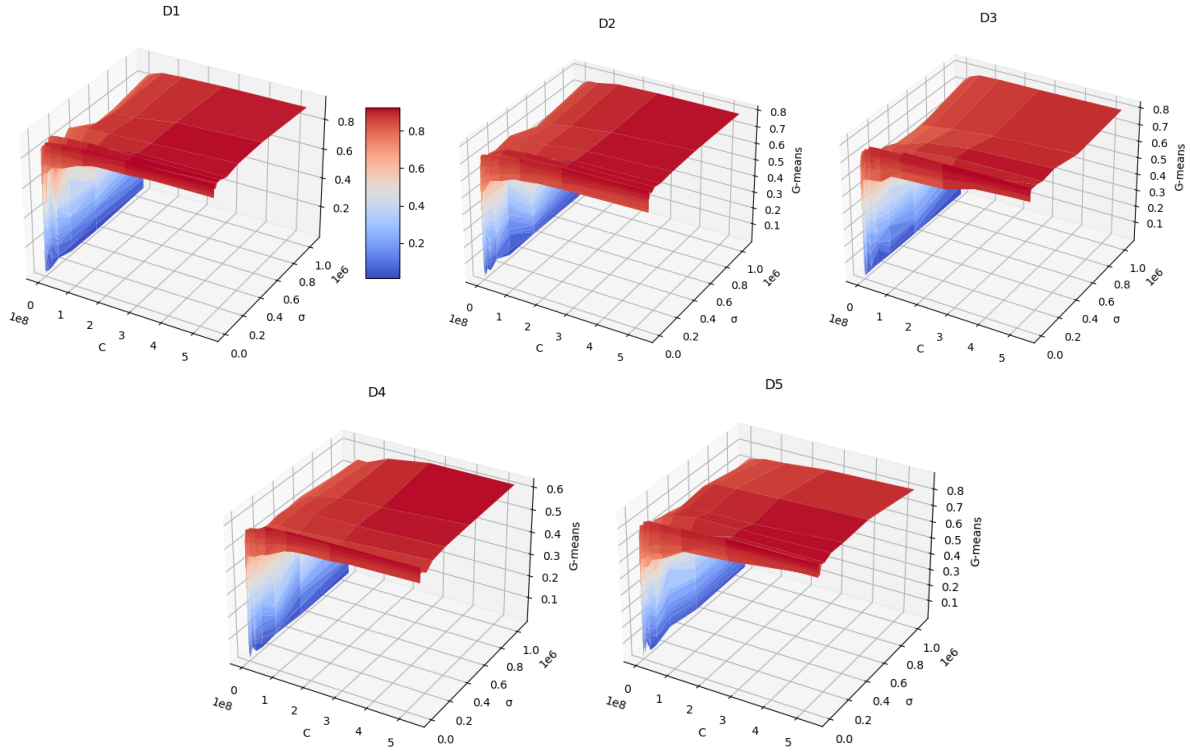


Figure. 33. Convergence of the G-mean for multiple choice of C and σ for Gaussian kernel on five benchmark datasets.

Table.14. Computing Results of Dataset

Dataset	Metric	KELM	WB-ELM	WB-ELM-PSO	WB-ELM-BA	WB-ELM-DA	WB-ELM-RSA
D1	Acc	0.8035	0.7821	0.9042	0.9312	0.9245	**0.9347**
	Prec	0.9001	0.811	0.8634	**0.9094**	0.8732	0.9051
	Recall	0.7222	0.7318	0.7332	0.7337	**0.7362**	0.7359
	G-mean	0.8062	0.7704	0.7956	0.8168	0.8018	**0.8161**
D2	Acc	0.7221	0.7249	0.7458	0.7736	0.7921	**0.8075**
	Prec	0.481	0.4918	0.5461	**0.5621**	0.5508	0.5535
	Recall	0.3341	0.3542	0.4103	0.5371	0.5646	**0.6078**
	G-mean	0.4008	0.4174	0.4732	0.5494	0.5576	**0.58**
D3	Acc	0.7811	0.7921	0.8111	0.8017	0.8022	**0.8178**
	Prec	0.7331	0.7515	0.7578	0.771	0.7943	0.8143
	Recall	0.9216	0.922	0.9426	0.9301	0.9474	0.9571
	G-mean	0.8219	0.8324	0.8452	0.8468	0.8675	**0.8828**
D4	Acc	0.7602	0.7621	0.7746	0.7817	0.7822	0.84
	Prec	**0.7809**	0.779	0.7782	0.7776	0.7641	0.78
	Recall	0.4618	0.4618	0.4833	0.4871	0.4957	0.5982
	G-mean	0.6005	0.5998	0.6133	0.6154	0.6154	**0.683**
D5	Acc	**0.8832**	0.8736	0.8742	0.8785	0.8814	**0.8811**
	Prec	**0.8747**	0.8631	0.8688	0.8609	0.8677	0.8625
	Recall	0.8608	0.8621	0.8671	**0.8782**	0.8707	0.8777
	G-mean	0.8677	0.8626	0.8679	0.8695	0.8692	**0.87**

The bolded measure is the best one. For D1, WB-ELM-BA has the best precision, WB-ELM-DA has the best recall, and WB-ELM-RSA has the best accuracy and G-mean. On Acc, Recall, and G-mean, WB-ELM-RSA performs the best for D2. Only accuracy is ideal for WB-ELM-BA, whereas the other characteristics are optimum for WB-ELM-RSA. For D3, WB-ELM-RSA has the best results on accuracy, recall, precision, and G-mean. For D4, WB-ELM-RSA has the best accuracy, recall, and precision. Finally, for D5, KELM has the best precision, and WB-ELM-BA has the best recall. WB-ELM-BA is the best for precision when the dataset is imbalanced. Additionally, KELM is the best on precision when the dataset is balanced, but WB-ELM-RSA performs better on accuracy and G-mean.

The tabulated results in Table 2 display the outcomes achieved by KELM, WB-ELM, and WB-ELM utilizing various evolutionary approaches: PSO, BA, DA, and RSA. The results indicate that the proposed WB-ELM exhibits superior classification performance compared to KELM. Furthermore, WB-ELM improves performance with evolutionary optimization approaches, particularly the RSA.

Table.15. Time cost (seconds) for different optimization algorithms during the testing phase

	WB-ELM	WB-ELM PSO	WB-ELM BAT	WB-ELM DA	WB-ELM RSA
D1	69	85	88	87	82
D2	117	135	142	159	124
D3	29	54	58	51	56
D4	13	21	23	24	20
D5	19	27	30	32	23

Table 3 provides information on the time costs in seconds for different optimization algorithms applied to the WB-ELM model.

Each row represents a specific dataset (D1, D2, D3, D4, and D5), and each column corresponds to a particular combination of the WB-ELM model with an optimization algorithm (WB-ELM PSO, WB-ELM BAT, WB-ELM DA, WB-ELM RSA). Based on the table results above, the time cost of WB-ELM-RSA is lower than the other optimization algorithms for some datasets. Considering our problem of detecting fraudulent transactions in real-time, focusing on time efficiency and performance, it is essential to balance the two factors. Additionally, the performance of optimization algorithms can be influenced by several factors, such as the size of the dataset and the specific characteristics of the optimization algorithm itself.

Table. 16: Parameter settings

Algorithm	Parameters
PSO	n = 20, c1 = 2, c2 = 2, w = 0,7311
BA	n = 20, A = 1, R = 1, $\alpha = \gamma = 0{,}9$
DA	n = 20, X = 20, max = 100
RSA	n =20 , α=0.1 , β= 0.1

n is the population size in Table 4, and the following parameters are introduced:

PSO: w stands for the inertia weight, and c1 and c2 are acceleration factors.

BA: R is the rate of pulse emission, A is loudness, and α and ξ are constants.

DA: max is the number of seeds, and X is the number of the best dandelions.

RSA: α, β are the hyperparameters of RSA

To evaluate the performance of our proposed model, we compared it with previous studies. The results are presented in Table 5. The proposed model achieved the highest accuracy and G-mean in all five datasets, outperforming the previous works as shown. Additionally, our model significantly reduced false alerts by minimizing false positives and false negatives, which is evident from the recall and precision values obtained. Further details can be found in Table 5.

Table. 17 Comparing Results with previous work

Dataset	Reference	Method	Acc	Recall	Precision	G-mean
Creditcardcsvpresent	(Liu et al., 2021)	mg-GBDT	0.6775	0.6375	0.6938	0.6650
	(H. Zhu et al., 2023)	Noisy-Sample-Removed Undersampling Scheme	-	-	-	0.698
		Our study	0.9347	0.7359	0.9051	0.8161
Default of Credit Card Clients Dataset	(H. Zhu et al., 2020)	$WELM_L$	0.7687	0.6032	0.5407	0.5711
		$WELM_B$	0.7657	0.6003	0.5359	0.5672
		$WELM_E$	0.7679	0.5973	0.5391	0.5674
	(H. Zhu et al., 2023)	Noisy-Sample-Removed	-	-	-	0.661

		Undersampling Scheme				
		Our study	.8075	0.6078	0.5535	0.58
German numerical	(C. L. Huang et al., 2007)	BPN	0.7783	-	-	-
		GP	0.781	-	-	-
		C4.5	0.736	-	-	-
		SVM + GA	0.7792	-	-	-
	(Tripathi et al., 2020)	ANN	72.80	-	-	-
		KNN	66.90	-	-	-
		SVM-L	74.80	-	-	-
		SVM-R	75.90	-	-	-
		CART	55.90	-	-	-
	(Tripathi et al., 2020)	J48	64.10	-	-	-
		LR-R	75.40	-	-	-
		ELM(new)		0.8928	0.4833	0.6569
		EELM(BA)	0.7591	0.9198	0.5166	0.6893
		EELM-1(BA)	0.7788	0.9361	0.4233	0.6295
	(I. Singh et al., 2021)	EELM-2(BA)	0.7893	0.9442	0.4833	0.6755
		MLCCE	0.8057	0.9671	0.8512	0.9073
		mg-GBDT	0.8580	0.9186	0.7847	0.8490
	(Liu et al., 2021)	XGBoost	0.7715	0.9208	0.7757	0.8451
		LightGBM	0.7582	0.9007	0.7887	0.8428
	(Zou & Gao, 2022)	AugBoost-RP	0.7615	0.8867	0.7862	0.8349
		AugBoost-PCA	0.7519	0.8852	0.7956	0.8392
			0.7605	0.9237	0.7766	0.847
			0.7604	0.9245	0.7775	0.8478
			0.7617			

		AugBoost-NN				
		AugBoost-ELM				
		Our study	0.8178	0.9571	0.8143	0.8828
Loan Prediction	(H. Zhu et al., 2020)	$WELM_L$	0.7883	0.5782	0.6951	0.6339
		$WELM_B$	0.7834	0.5834	0.6792	0.6295
		$WELM_E$	0.7867	0.5837	0.7016	0.64
	(Hasan et al., 2021)	KNN	0.837	-	-	-
		SVM	0.831	-	-	-
		Decision Tree	0.831	-	-	-
		Logistic Regres	0.824	-	-	-
		RF	0.798	-	-	-
	(H. Zhu et al., 2023)	Noisy-Sample-Removed Undersampling Scheme	-	-	-	0.698
		Our study	0.84	0.5982	0.78	0.6283

So, our method's superior performance can be attributed to several key factors. The Reptile Search Algorithm (RSA) possesses an inherent adaptability or sensitivity to the characteristics of the WB-ELM optimization problem in the context of imbalanced binary classification. This adaptability could lead to more effective parameter space exploration and better convergence to optimal solutions.

As a metaheuristic algorithm, RSA could strike a better balance between exploration (searching for new solutions) and exploitation (utilizing known solutions). This balance is crucial for optimization, especially in complex problems, and might give RSA an edge in finding optimal or near-optimal solutions.

Different optimization algorithms have different parameter settings, and RSA's tuning parameters may align more effectively with the nature of the WB-ELM algorithm.
RSA converges more quickly to reasonable solutions compared to other algorithms. It could be due to its unique approach to updating solution candidates, which allows it to narrow down promising solutions faster.
RSA introduces a novel algorithmic approach, which could bring diversity to the optimization landscape. This diversity allows it to escape local optima more effectively and discover more globally optimal solutions.
The study's empirical results show that RSA outperforms other algorithms, which suggests that it has an advantage in solution quality and convergence speed for detecting fraudulent transactions.

5. Conclusion

This study aimed to tackle the imbalanced classification problem by comparing various optimization techniques applied to the Weighted Binary Extreme Learning Machine (WB-ELM) and evaluating their performance using accuracy, precision, recall, and G-mean metrics. Among the optimization techniques examined, including kernel ELM, Weighted Binary ELM (WBELM), WB-ELM optimized by Particle Swarm Optimization (PSO), WB-ELM optimized by Bat Algorithm (BA), WB-ELM optimized by Dandelion Algorithm (DA), and WB-ELM optimized by Reptile Search Algorithm (RSA), the results consistently demonstrated that WB-ELM optimized by RSA achieved the best performance across all evaluated metrics.
The Reptile Search Algorithm, inspired by the hunting behavior of reptiles, exhibited superior efficiency in searching for optimal solutions within the WB-ELM problem space. RSA outperformed the other optimization methods, including PSO, BA, and DA, particularly in imbalanced binary classification tasks by striking a balance between performance and time efficiency.
The evaluation of the proposed techniques was also extended to credit card fraud detection, where real-time processing and accurate results are crucial. In this application domain, WB-ELM optimized by RSA showcased high detection performance while meeting the time constraints of real-time processing.

General Conclusion and Perspectives

In conclusion, this thesis has taken us on an engaging progress through the intricacies of credit card fraud detection, exploring various facets from the current landscape (state of the art) to concrete developments and future perspectives.

In the first chapter, we established a robust foundation by examining the state of the art in fraud detection and presented the CCF detection background by navigating through various techniques, from data mining to sophisticated approaches in deep learning and metaheuristics. Also, identifying key gaps and challenges that persist in the current research landscape. This thorough analysis situated our work within the broader research context, highlighting the need for innovative solutions.

The second chapter sheds light on an innovative approach to address the challenge of imbalanced resolution in credit card fraud detection. By specifically understanding and tackling the inherent imbalances in data, our proposal demonstrated a significant improvement in the accuracy of detection models, paving the way for more reliable real-world applications. These contributions can be summarized as follows:

- The first contribution aimed to detect credit card fraud using two artificial neural network algorithms, Multilayer Perceptron (MLP) and Extreme Learning Machine (ELM).
- The second contribution introduced a pioneering approach that combines two architectures: the AE and the MLP, forming a cohesive FCDN. Leveraging the AE to extract pertinent features from input transactions serves to reduce data complexity while retaining essential information. The extracted features are then classified using an optimized, fully connected deep network (FCDN) to obtain the model's optimal parameters.
- The third contribution presented a novel approach that combines AE and FCDN models to address this issue. The process involves three phases: training an AE on fraudulent transactions, utilizing another AE for dimensionality reduction, and using the encoded representations as input for FCDN classification. To further enhance the model's performance, we introduced an additional FCDN trained on the preprocessed data using the Synthetic Minority Oversampling Technique (SMOTE). The predictions from both AE, AE-FCDN, and the FCDN are combined using a majority voting approach. This approach offers superior performance to alternative optimization algorithms, particularly real-time credit card fraud detection. AE and FCDN have advanced the

capabilities of credit card fraud detection. These models address class imbalance issues, harness ensemble techniques, and outperform existing methods, setting new benchmarks for the field.

The third chapter focused on designing an optimization model to enhance overall system performance. Through rigorous methodology and meticulous implementation, we achieved promising results contributing to the efficiency and robustness of fraud detection systems.

In summary, this thesis has made substantial contributions to credit card fraud detection, surpassing current research boundaries. Striking a balance between theoretical understanding and practical application, we hope this study will serve as a solid foundation for future developments in the field.

We eagerly await the exciting advances that will follow from these solid foundations and are excited to apply these novel ideas in practical applications. Thank you for accompanying me throughout this academic journey, and we are confident that our contribution will have a lasting impact in the field of credit card fraud detection.

With excellent prospects and unexplored areas, the future horizons for credit card fraud detection appear optimistic. The research recommends that these models be continuously improved and evolved to address new fraud trends and advanced cyber threats. Investigating new directions including incorporating more characteristics, utilizing sophisticated anomaly detection methods, and adopting different machine learning algorithms may strengthen the robustness of fraud detection systems.

Furthermore, extending the application of these techniques beyond credit card transactions into diverse financial transactions and cybersecurity domains holds immense potential. Collaborative efforts with industry stakeholders and financial institutions to validate and implement these models in real-world settings could offer invaluable insights into their scalability, adaptability, and efficacy.

This chapter provides a perspective, describing the concrete effects and sketching the direction of future research on strengthening financial security.

By providing workable ways to improve fraud detection and opening the door for their inclusion in current or future systems, my study adds long-term value. My contributions could be used in the following ways to improve fraud detection:

- **Enhanced Detection Algorithms:** By combining advanced detection models like deep neural networks and machine learning algorithms, my research enables more effective identification of complex fraud patterns, thereby enhancing the reliability of detection systems.

- **Management of Imbalanced Data:** My work on handling imbalanced datasets provides methods to balance transaction data, improving the accuracy of detection models and reducing false alerts.
- **Optimization of Existing Models:** By proposing optimization models and performance techniques, my research provides means to enhance existing detection models, allowing faster and more precise fraud detection.
- **Application in Existing Systems:** My contributions can be integrated into existing fraud detection systems, offering incremental improvements without necessitating major changes in the current infrastructure.

1. Limitations and ways to improve

The limitations of my research provide crucial opportunities for future improvements. Here are some identified limitations along with associated prospects for enhancement:

Data Size and Quality:

- *Limitation:* Results may not be similarly applicable if there is a lack of data or if dataset quality varies.
- *Improvement Paths:* Collect larger and more representative datasets, and employ advanced data cleaning and augmentation techniques to enhance dataset quality.

Performance Evaluation:

- *Limitation:* Performance evaluation of models may sometimes not accurately reflect their real-world effectiveness.
- *Improvement Paths:* Use more robust evaluation metrics and conduct additional experiments in real-world environments to validate model reliability.

Model Complexity:

- *Limitation:* Overly complex models can be challenging to interpret or implement in real-time settings.
- *Improvement Paths:* Search for simpler but more efficient models that prioritize interpretability without compromising performance.

Model Generalization:

- *Limitation:* Developed models may be specific to certain data or contexts, limiting their generalizability.
- *Improvement Paths:* Explore transfer learning techniques to make models more adaptable to diverse scenarios.

System Interoperability:

- *Limitation:* Integrating new models or techniques into existing systems may be complex due to limited interoperability.
- *Improvement Paths:* Develop more compatible and adaptable solutions to facilitate integration into existing systems.

By addressing these limitations and exploring these avenues for improvement, it's possible to strengthen the quality, applicability, and effectiveness of future research in the field of credit card fraud detection.

2. Future developments

In the context of fraud detection, several areas require further improvements or additional research to advance knowledge and tools. Here are potential areas for further developments :

- **Model Interpretability:** Explore techniques to enhance understanding and explanation of complex models, especially in deep neural networks, to increase user and decision-maker confidence in algorithmic decisions.
- **Real-Time Fraud Detection:** Develop models and systems capable of detecting and preventing fraud in real-time, reducing the time between detection and preventive action.
- **Fraud Detection in New Contexts:** Extend research to detect fraud in emerging new contexts, such as blockchain-based transactions, advanced mobile payments, or AI-driven financial transactions.
- **Cybersecurity and Threat Prevention:** Consolidate cybersecurity tactics with fraud detection to create more comprehensive security techniques that can more effectively resist complex attacks and other threats.

- **Synthetic Data Development:** Explore methods to generate realistic synthetic data to address the lack of real-world data in fraud detection while ensuring the confidentiality of sensitive information.

3. Emerging technology

In the realm of fraud detection, my forthcoming research will center on harnessing emerging technologies to significantly enhance the efficacy of detection and prevention methods. The incorporation of advanced artificial intelligence models, including recurrent neural networks, convolutional neural networks, and natural language processing, will be a focal point. These models have the potential to detect complex fraudulent patterns, improving the accuracy and precision of fraud detection systems. Further, acknowledging the increasing popularity of blockchain transactions and the increasing usage of IoT devices in financial transactions, I want to modify current detection techniques to efficiently traverse unstructured and full-of-information scenarios. Embracing these emerging technologies, my research will not only refine current models to detect anomalies in evolving transaction systems but also explore behavior-based security technologies. By leveraging user behavior analysis, these technologies offer innovative avenues for fortifying fraud detection mechanisms.

4. Cooperation and collaboration

To propel and enrich the current research trajectory, potential collaborations with various researchers or institutions present pivotal opportunities. Partnering with universities and research centers specialized in artificial intelligence, data security, and financial transactions could expand the breadth of knowledge and perspectives within the domain of fraud detection. Additionally, close collaboration with entities in the financial industry, including financial institutions and fintech companies, offers practical insights into real-world fraud detection challenges, facilitating model testing and the integration of pragmatic solutions. Engaging with experts in blockchain, IoT, and data security could refine and adapt detection models for emerging and complex environments. Furthermore, interdisciplinary collaborations with professionals from behavioral psychology or sociology could offer unique insights into integrating behavioral perspectives within fraud detection frameworks. Such collaborations hold promise in not only enriching research but also fostering comprehensive model testing and practical implementation in real-world fraud detection setups.

5. Conclusion

In conclusion, this chapter has delved into the limitations and ways to improve and future perspectives of the study on credit card fraud detection. The implementation of advanced models and approaches, such as deep neural networks, imbalanced data handling, and optimized autoencoders, in our contributions, marks substantial progress in strengthening enhanced transaction security and protection of financial institutions against fraudulent activities.

Looking ahead, the prospects for credit card fraud detection appear promising, with unexplored areas and excellent prospects awaiting exploration. The study recommends a commitment to continuous improvement and evolving models to address emerging fraud trends and advanced cyber threats.

In the realm of future work, researchers are encouraged to investigate the seamless integration of blockchain and IoT into existing credit card fraud detection models. This interdisciplinary approach holds the promise of not only bolstering the security of financial transactions but also establishing a foundation for secure, transparent, and efficient financial ecosystems

Publication list

- Detection of SMS Spam Using Machine-Learning Algorithms. FZ El Hlouli, J Riffi, MA Mahraz, A El Yahyaouy, H Tairi. Embedded Systems and Artificial Intelligence, 429-440,2020. DOI: 10.1007/978-981-15-0947-6_41.

- Credit Card Fraud Detection Based on Multilayer Perceptron and Extreme Learning Machine Architectures FZ EL hlouli, J Riffi, MA Mahraz, AE Yahyaouy, H Tairi. Intelligent Systems and Computer Vision (ISCV), 1-5,2020. **DOI:** 10.1109/ISCV49265.2020.9204185.

- Detecting Fraudulent Transactions Using Stacked Autoencoder Kernel ELM Optimized by the Dandelion Algorithm FZ El Hlouli, J Riffi, M Sayyouri, MA Mahraz, A Yahyaouy, K El Fazazy, H Tairi.Journal of Theoretical and Applied Electronic Commerce Research 18 (4), 2057-2076,2023. https://www.mdpi.com/0718-1876/18/4/103.

- Credit Card Fraud Detection: Addressing Imbalanced Datasets with a Multi-Phase Approach. FZ El Hlouli, J Riffi, MA Mahraz, A El Yahyaouy, H Tairi. *SN COMPUT. SCI.* **5**, 173 (2024). https://doi.org/10.1007/s42979-023-02559-6.

References

Abd Elrahman, S. M., & Abraham, A. (2013a). A Review of Class Imbalance Problem. *Journal*

of Network and Innovative Computing, *1*, 332–340. www.mirlabs.net/jnic/index.html

Abd Elrahman, S. M., & Abraham, A. (2013b). A Review of Class Imbalance Problem. *Journal of Network and Innovative Computing*, *1*, 332–340.

Abualigah, L. (2021). Group search optimizer: a nature-inspired meta-heuristic optimization algorithm with its results, variants, and applications. *Neural Computing and Applications*, *33*(7), 2949–2972. https://doi.org/10.1007/s00521-020-05107-y

Abualigah, L., Elaziz, M. A., Sumari, P., Geem, Z. W., & Gandomi, A. H. (2022). Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer. *Expert Systems with Applications*, *191*(October 2021), 116158. https://doi.org/10.1016/j.eswa.2021.116158

Al-Hashedi, K. G., & Magalingam, P. (2021). Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019. *Computer Science Review*, *40*. https://doi.org/10.1016/j.cosrev.2021.100402

Alejo, R., Valdovinos, R. M., García, V., & Pacheco-Sanchez, J. H. (2013). A hybrid method to face class overlap and class imbalance on neural networks and multi-class scenarios. *Pattern Recognition Letters*, *34*(4), 380–388. https://doi.org/10.1016/j.patrec.2012.09.003

Alghofaili, Y., Albattah, A., & Rassam, M. A. (2020). A Financial Fraud Detection Model Based on LSTM Deep Learning Technique A Financial Fraud Detection Model Based on LSTM. *Journal of Applied Security Research*, *0*(0), 1–19. https://doi.org/10.1080/19361610.2020.1815491

Almuteer, A. H., Aloufi, A. A., Alrashidi, W. O., Alshobaili, J. F., & Ibrahim, D. M. (2021). Detecting Credit Card Fraud using Machine Learning. *International Journal of Interactive Mobile Technologies*, *15*(24), 108–122. https://doi.org/10.3991/IJIM.V15I24.27355

Anusandhan, S. O. (2018). Effect of Class Imbalanceness in Detecting Automobile Insurance Fraud. *2018 2nd International Conference on Data Science and Business Analytics (ICDSBA)*, 528–531. https://doi.org/10.1109/ICDSBA.2018.00104

Avenue, N. (2004). *Extreme Learning Machine: RBF Network Case*. *December*, 6–9.

Awoyemi, J. O., Adetunmbi, A. O., & Oluwadare, S. A. (2017a). Credit card fraud detection using machine learning techniques: A comparative analysis. *Proceedings of the IEEE International Conference on Computing, Networking and Informatics, ICCNI 2017*, *2017-Janua*, 1–9. https://doi.org/10.1109/ICCNI.2017.8123782

Awoyemi, J. O., Adetunmbi, A. O., & Oluwadare, S. A. (2017b). Credit card fraud detection using machine learning techniques: A comparative analysis. *Proceedings of the IEEE International Conference on Computing, Networking and Informatics, ICCNI 2017*, *2017-Janua*, 1–9. https://doi.org/10.1109/ICCNI.2017.8123782

Behera, T. K., & Panigrahi, S. (2015). Credit Card Fraud Detection: A Hybrid Approach Using Fuzzy Clustering & Neural Network. *Proceedings - 2015 2nd IEEE International Conference on Advances in Computing and Communication Engineering, ICACCE 2015*, 494–499. https://doi.org/10.1109/ICACCE.2015.33

Bejjanki, K. K., Gyani, J., & Gugulothu, N. (2020). Class imbalance reduction (CIR): A novel approach to software defect prediction in the presence of class imbalance. *Symmetry*, *12*(3). https://doi.org/10.3390/sym12030407

Benchaji, I., & Ouahidi, B. El. (2018). *Using Genetic Algorithm to Improve Classification of*

Imbalanced Datasets for credit card fraud detection. 2–6.

Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud : A comparative study. *Decision Support Systems*, *50*(3), 602–613. https://doi.org/10.1016/j.dss.2010.08.008

Boukerche, A., Zheng, L., & Alfandi, O. (2020). Outlier Detection: Methods, Models, and Classification. *ACM Computing Surveys*, *53*(3). https://doi.org/10.1145/3381028

Cao, F., Liu, B., & Sun, D. (2013). Neurocomputing Image classification based on effective extreme learning machine. *Neurocomputing*, *102*, 90–97. https://doi.org/10.1016/j.neucom.2012.02.042

Chandola, V., Banerjee, A., & Kumar, V. (2009). *Anomaly Detection : A Survey Technical Report Department of Computer Science TR 07-017 Anomaly Detection : A Survey Varun Chandola , Arindam Banerjee , and Vipin Kumar*. *May 2014*. https://doi.org/10.1145/1541880.1541882

Chen, C., Li, W., Su, H., & Liu, K. (2014). *Spectral-Spatial Classification of Hyperspectral Image Based on Kernel Extreme Learning Machine*. *June*. https://doi.org/10.3390/rs6065795

Cochrane, N., Gomez, T., Warmerdam, J., Flores, M., McCullough, P., Weinberger, V., & Pirouz, M. (2021). Pattern Analysis for Transaction Fraud Detection. *2021 IEEE 11th Annual Computing and Communication Workshop and Conference, CCWC 2021*, 283–289. https://doi.org/10.1109/CCWC51732.2021.9376045

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. *Proceedings of the 25th International Conference on Machine Learning*, 160–167.

Conference, I., & Technologies, C. (2016). *Text Classification Using KM-ELM Classifier*.

Darban, Z. Z., Webb, G. I., Pan, S., Aggarwal, C. C., & Salehi, M. (2022). *Deep Learning for Time Series Anomaly Detection: A Survey*. *1*(1). https://doi.org/10.48550/arXiv.2211.05244

Denil, M., Shakibi, B., Dinh, L., Ranzato, M., & De Freitas, N. (2013). Predicting parameters in deep learning. *Advances in Neural Information Processing Systems*, 1–9.

Ding, S., Xu, X., & Nie, R. (2014). Extreme learning machine and its applications. *Neural Computing and Applications*, *25*(3–4), 549–556. https://doi.org/10.1007/s00521-013-1522-8

Doersch, C. (2016). *Tutorial on Variational Autoencoders*. *August*. http://arxiv.org/abs/1606.05908

Dornadula, V. N., & Geetha, S. (2019). Credit Card Fraud Detection using Machine Learning Algorithms. *Procedia Computer Science*, *165*, 631–641. https://doi.org/10.1016/j.procs.2020.01.057

Doshi, S., Desai, K., & Shukla, D. (2023). Comparative Study of Fraudulent Activities and Various Fraud Detection Techniques. *International Journal for Research in Applied Science and Engineering Technology*, *11*(8), 1140–1148. https://doi.org/10.22214/ijraset.2023.55308

Dubey, S. C., Mundhe, K. S., & Kadam, A. A. (2020). Credit Card Fraud Detection using

Artificial Neural Network and BackPropagation. *Proceedings of the International Conference on Intelligent Computing and Control Systems, ICICCS 2020*, *Iciccs*, 268–273. https://doi.org/10.1109/ICICCS48265.2020.9120957

Eberhart, J., & Kennedy, R. (1995). Particle Swarm Optimisation. *Proceedings of ICNN'95 - International Conference on Neural Networks*, *927*, 1942–1948. https://doi.org/10.1109/ICNN.1995.488968

El Bouchti, A., Chakroun, A., Abbar, H., & Okar, C. (2017). Fraud detection in banking using deep reinforcement learning. *7th International Conference on Innovative Computing Technology, INTECH 2017*, *Intech*, 58–63. https://doi.org/10.1109/INTECH.2017.8102446

El Hlouli, F. Z., Riffi, J., Mahraz, M. A., El Yahyaouy, A., & Tairi, H. (2020). Credit Card Fraud Detection Based on Multilayer Perceptron and Extreme Learning Machine Architectures. *2020 International Conference on Intelligent Systems and Computer Vision, ISCV 2020*, *June*. https://doi.org/10.1109/ISCV49265.2020.9204185

Fanai, H., & Abbasimehr, H. (2023). A novel combined approach based on deep Autoencoder and deep classifiers for credit card fraud detection. *Expert Systems with Applications*, *217*(December 2022). https://doi.org/10.1016/j.eswa.2023.119562

Fernández, A., García, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary. *Journal of Artificial Intelligence Research*, *61*, 863–905. https://doi.org/10.1613/jair.1.11192

Filali, H., Riffi, J., Boulealam, C., Mahraz, M. A., & Tairi, H. (2022). Multimodal Emotional Classification Based on Meaningful Learning. *Big Data and Cognitive Computing*, *6*(3). https://doi.org/10.3390/bdcc6030095

Francis, C., Pepper, N., & Strong, H. (2011). Using support vector machines to detect medical fraud and abuse. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 8291–8294. https://doi.org/10.1109/IEMBS.2011.6092044

Freund, Y., Schapire, R. E., & Hill, M. (1996). *Experiments with a New Boosting Algorithm Rooms f 2B-428 , 2A-424 g*.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). *Bayesian Network Classifiers *. 163*, 131–163.

Fu, K., Cheng, D., Tu, Y., & B, L. Z. (2016). *Credit Card Fraud Detection Using Convolutional Neural Networks*. 483–490. https://doi.org/10.1007/978-3-319-46675-0

Galar, D., Parida, A., Kumar, U., Stenstrom, C., & Berges, L. (2011). Maintenance metrics: A hierarchical model of balanced scorecard. *2011 IEEE International Conference on Quality and Reliability, ICQR 2011*, *October*, 67–74. https://doi.org/10.1109/ICQR.2011.6031683

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., & Herrera, F. (2012). A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, *42*(4), 463–484. https://doi.org/10.1109/TSMCC.2011.2161285

Ghobadi, F., & Rohani, M. (2017). Cost sensitive modeling of credit card fraud using neural network strategy. *Proceedings - 2016 2nd International Conference of Signal Processing and Intelligent Systems, ICSPIS 2016*, 8–10.

https://doi.org/10.1109/ICSPIS.2016.7869880

Gong, C., Han, S., Li, X., Zhao, L., & Liu, X. (2018). A new dandelion algorithm and optimization for extreme learning machine. *Journal of Experimental and Theoretical Artificial Intelligence*, *30*(1), 39–52. https://doi.org/10.1080/0952813X.2017.1413142

Guang-Bin Huang, C.-K. S. (2004). Extreme Learning Machine : RBF Network Case. *Guang-Bin Huang, & Chee-Kheong Siew. (n.d.). Extreme Learning Machine: RBF Network Case. ICARCV 2004 8th Control, Automation, Robotics and Vision Conference, 2004. Doi:10.1109/Icarcv.2004.1468985*, *Icarcv*. https://doi.org/doi:10.1109/icarcv.2004.1468985

Guo, H., & Viktor, H. L. (2004). Learning from imbalanced data sets with boosting and data generation. *ACM SIGKDD Explorations Newsletter*, *6*(1), 30–39. https://doi.org/10.1145/1007730.1007736

Gyamfi, N. K., & Abdulai, J. D. (2019). Bank Fraud Detection Using Support Vector Machine. *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference, IEMCON 2018, May*, 37–41. https://doi.org/10.1109/IEMCON.2018.8614994

Hajek, P., Abedin, M. Z., & Sivarajah, U. (2022). Fraud Detection in Mobile Payment Systems using an XGBoost-based Framework. *Information Systems Frontiers*, *0123456789*. https://doi.org/10.1007/s10796-022-10346-6

Hamdi, M. (2010). *Improving a credit card fraud detection system using genetic algorithm*. 436–440.

Han, S., Pool, J., Tran, J., & Dally, W. J. (2015). Learning both weights and connections for efficient neural networks. *Advances in Neural Information Processing Systems*, *2015-Janua*, 1135–1143.

Haratinik, M. R., Akrami, M., Khadivi, S., & Shajari, M. (2012). FUZZGY: A hybrid model for credit card fraud detection. *2012 6th International Symposium on Telecommunications, IST 2012*, 1088–1093. https://doi.org/10.1109/ISTEL.2012.6483148

Hasan, N., Anzum, T., Hasan, T., & Jahan, N. (2021). Machine Learning Algorithm to Predict Fraudulent Loan Requests. *2021 12th International Conference on Computing Communication and Networking Technologies, ICCCNT 2021, November 2022*. https://doi.org/10.1109/ICCCNT51525.2021.9579517

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *Proceedings of the International Joint Conference on Neural Networks, July 2008*, 1322–1328. https://doi.org/10.1109/IJCNN.2008.4633969

Hilal, W., Gadsden, S. A., & Yawney, J. (2022). Financial Fraud : A Review of Anomaly Detection Techniques and Recent Advances. *Expert Systems With Applications*, *193*, 116429. https://doi.org/10.1016/j.eswa.2021.116429

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, Abdel-rahman Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., & Kingsbury, B. (2012). Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Processing Magazine, 29(6), 82–97,2012.*, 82–97. https://doi.org/10.1109/msp.2012.2205597

Huang, Guang Bin, Zhou, H., Ding, X., & Zhang, R. (2012). Extreme learning machine for

regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *42*(2), 513–529. https://doi.org/10.1109/TSMCB.2011.2168604

Huang, Guang Bin, Zhu, Q. Y., & Siew, C. K. (2004). Extreme learning machine: A new learning scheme of feedforward neural networks. *IEEE International Conference on Neural Networks - Conference Proceedings*, *2*(August 2004), 985–990. https://doi.org/10.1109/IJCNN.2004.1380068

Huang, C. L., Chen, M. C., & Wang, C. J. (2007). Credit scoring with a data mining approach based on support vector machines. *Expert Systems with Applications*, *33*(4), 847–856. https://doi.org/10.1016/j.eswa.2006.07.007

Huang, Gao, Huang, G. Bin, Song, S., & You, K. (2019). Trends in extreme learning machines: A review. *IEEE Access*, *7*, 108070–108089. https://doi.org/10.1016/j.neunet.2014.10.001

Huang, Guang-bin, & Siew, C. K. (2014). *Extreme Learning Machine with Randomly Assigned RBF Kernels*. *January 2005*.

Itoo, F., Meenakshi, & Singh, S. (2020). Comparison and analysis of logistic regression, Naıve Bayes and KNN machine learning algorithms for credit card fraud detection. *International Journal of Information Technology*, 1–9. https://doi.org/10.1007/s41870-020-00430-y

Jeragh, M., & Alsulaimi, M. (2019). Combining Auto Encoders and One Class Support Vectors Machine for Fraudulant Credit Card Transactions Detection. *Proceedings of the 2nd World Conference on Smart Trends in Systems, Security and Sustainability, WorldS4 2018*, 57–64. https://doi.org/10.1109/WorldS4.2018.8611624

Jiang, J., Han, F., Ling, Q. H., & Su, B. Y. (2018). An Improved Evolutionary Extreme Learning Machine Based on Multiobjective Particle Swarm Optimization. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10956 LNAI*, 1–6. https://doi.org/10.1007/978-3-319-95957-3_1

Jie, L., & Wei, L. V. (2009). *Research on Detecting Technique of Financial Statement Fraud Based on Fuzzy Genetic Algorithms BPN*. 1462–1468.

Kazemi, Z., & Zarrabi, H. (2017). *Using deep networks for fraud detection in the credit card transactions*. 630–633.

Khalilia, M., Chakraborty, S., & Popescu, M. (2011). Predicting disease risks from highly imbalanced data using random forest. *BMC Medical Informatics and Decision Making*, *11*(1), 51. https://doi.org/10.1186/1472-6947-11-51

Khan, A., Singh, T., & Sinhal, A. (2012). Implement credit card fraudulent detection system using observation probabilistic in hidden Markov model. *3rd Nirma University International Conference on Engineering, NUiCONE 2012*, 6–8. https://doi.org/10.1109/NUICONE.2012.6493206

Khateeb, A., & Alrabba, H. (2023). *How the frequency of monitoring financial statements affect the ability to detect fraud and financial irregularities in the financial industry Abstract*. *September*.

Kho, J. R. D., & Vea, L. A. (2017). *Credit Card Fraud Detection Based on Transaction Behavior*. *2*, 1880–1884.

Kucharski, D., Kleczek, P., Jaworek-Korjakowska, J., Dyduch, G., & Gorgon, M. (2020). Semi-supervised nests of melanocytes segmentation method using convolutional autoencoders. *Sensors (Switzerland)*, *20*(6), 1–19. https://doi.org/10.3390/s20061546

LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, *3361*(May), 255–258. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.32.9297&rep=rep1&type=pdf

Li, C., Zhou, J., Tao, M., Du, K., Wang, S., Jahed Armaghani, D., & Tonnizam Mohamad, E. (2022). Developing hybrid ELM-ALO, ELM-LSO and ELM-SOA models for predicting advance rate of TBM. *Transportation Geotechnics*, *36*(July), 100819. https://doi.org/10.1016/j.trgeo.2022.100819

Li, H. B., Wang, W., Ding, H. W., & Dong, J. (2009). *Trees Weighting Random Forest Method for Classifying High-Dimensional Noisy Data*. https://doi.org/10.1109/ICEBE.2010.99

Li, X., Han, S., Zhao, L., Gong, C., & Liu, X. (2017). New Dandelion Algorithm Optimizes Extreme Learning Machine for Biomedical Classification Problems. *Computational Intelligence and Neuroscience*, *2017*. https://doi.org/10.1155/2017/4523754

Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 1–21. https://doi.org/10.1109/tnnls.2021.3084827

Lin, S. Y., Chiang, C. C., Li, J. Bin, Hung, Z. S., & Chao, K. M. (2018). Dynamic fine-tuning stacked auto-encoder neural network for weather forecast. *Future Generation Computer Systems*, *89*, 446–454. https://doi.org/10.1016/j.future.2018.06.052

Ling, C. X., & Li, C. (1998). Data Mining for Direct Marketing: Problems and Solutions. *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, KDD 1998*, *98*, 73–79.

Liu, W., Fan, H., & Xia, M. (2021). Step-wise multi-grained augmented gradient boosting decision trees for credit scoring. *Engineering Applications of Artificial Intelligence*, *97*(October 2020), 104036. https://doi.org/10.1016/j.engappai.2020.104036

Lu, B. L., Wang, X. L., Yang, Y., & Zhao, H. (2011). Learning from imbalanced data sets with a Min-Max modular support vector machine. *Frontiers of Electrical and Electronic Engineering in China*, *6*(1), 56–71. https://doi.org/10.1007/s11460-011-0127-1

Lucas, Y., & Jurgovsky, J. (2020). Credit card fraud detection using machine learning: A survey. *Proceedings of the International Conference on Intelligent Computing and Control Systems, ICICCS 2020*, *October 2020*, 1264–1270. https://doi.org/10.1109/ICICCS48265.2020.9121114

Makhzani, A., & Frey, B. (2014). k-Sparse autoencoders. *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*.

Makki, S., Assaghir, Z., Taher, Y., Haque, R., Hacid, M. S., & Zeineddine, H. (2019). An Experimental Study With Imbalanced Classification Approaches for Credit Card Fraud Detection. *IEEE Access*, *7*, 93010–93022. https://doi.org/10.1109/ACCESS.2019.2927266

Manju Parkavi, R., Shanthi, M., & Bhuvaneshwari, M. C. (2017). Recent trends in ELM and MLELM: A review. *Advances in Science, Technology and Engineering Systems*, *2*(1), 69–

75. https://doi.org/10.25046/aj020108

Mareeswari, V., & Gunasekaran, G. (2016). Prevention of credit card fraud detection based on HSVM. *2016 International Conference on Information Communication and Embedded Systems, ICICES 2016, Icices*, 1–4. https://doi.org/10.1109/ICICES.2016.7518889

Mhamane, S. S., & Lobo, L. M. R. J. (2012). Internet banking fraud detection using HMM. *2012 3rd International Conference on Computing, Communication and Networking Technologies, ICCCNT 2012, July*. https://doi.org/10.1109/ICCCNT.2012.6395910

Mienye, I. D., & Sun, Y. (2021). Performance analysis of cost-sensitive learning methods with application to imbalanced medical data. *Informatics in Medicine Unlocked*, *25*(August), 100690. https://doi.org/10.1016/j.imu.2021.100690

Mirjalili, S., Mirjalili, S. M., & Yang, X. S. (2014). Binary bat algorithm. *Neural Computing and Applications*, *25*(3–4), 663–681. https://doi.org/10.1007/s00521-013-1525-5

Mishra, A., Nair, N. K. C., & Patel, N. D. (2018). Fault current characterisation of single phase inverter systems. *IEEE Power and Energy Society General Meeting*, *2018-Janua*, 1–5. https://doi.org/10.1109/PESGM.2017.8274697

Mohapatra, P., Chakravarty, S., & Dash, P. K. (2015). An improved cuckoo search based extreme learning machine for medical data classification. *Swarm and Evolutionary Computation*, *24*, 25–49. https://doi.org/10.1016/j.swevo.2015.05.003

N.Malini and Dr.M.Pushpa. (2017). *Analysis on Credit Card Fraud Identification Techniques based on KNN and Outlier DetectionThird IEEE International Conference on Advances in Electrical, Electronics, Information, Communication and Bio Informatics : AEEICB - 2017 : 27th & 28th February 20*. 3–6.

Naik, H., & Kanikar, P. (2019). Credit card fraud detection based on machine learning algorithms. *Computer Applications*, *182*(44), 8–12. https://doi.org/International Journal of Computer Applications

Nezhad, F. S., & Shahriari, H. R. (2013). Fuzzy logic and Takagi-Sugeno Neural-Fuzzy to Deutsche bank fraud transactions. *2013 7th Intenational Conference on E-Commerce in Developing Countries: With Focus on e-Security, ECDC 2013*, 1–15. https://doi.org/10.1109/ECDC.2013.6556727

Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in fi nancial fraud detection : A classi fi cation framework and an academic review of literature. *Decision Support Systems*, *50*(3), 559–569. https://doi.org/10.1016/j.dss.2010.08.006

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, & W. Philip Kegelmeyer. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *30*(2), 321–357.

Niu, X., Wang, L., & Yang, X. (2019). *A Comparison Study of Credit Card Fraud Detection: Supervised versus Unsupervised*. http://arxiv.org/abs/1904.10604

Paleologo, G., Elisseeff, A., & Antonini, G. (2010). Subagging for credit scoring models. *European Journal of Operational Research*, *201*(2), 490–499. https://doi.org/10.1016/j.ejor.2009.03.008

Pang, G., Shen, C., Cao, L., & Hengel, A. Van Den. (2021). Deep Learning for Anomaly

Detection: A Review. *ACM Computing Surveys*, *54*(2). https://doi.org/10.1145/3439950

Phua, C., Alahakoon, D., & Lee, V. (2004). Minority report in fraud detection. *ACM SIGKDD Explorations Newsletter*, *6*(1), 50–59. https://doi.org/10.1145/1007730.1007738

Popat, R. R., & Chaudhary, J. (2018). A Survey on Credit Card Fraud Detection using Machine Learning. *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, *Icoei*, 1120–1125.

Purschke, O., Sykes, M. T., Poschlod, P., Michalski, S. G., Römermann, C., Durka, W., Kühn, I., & Prentice, H. C. (2014). Interactive effects of landscape history and current management on dispersal trait diversity in grassland plant communities. *Journal of Ecology*, *102*(2), 437–446. https://doi.org/10.1111/1365-2745.12199

Qingshan, D. (2009). Application of support vector machine in the detection of fraudulent financial statements. *Proceedings of 2009 4th International Conference on Computer Science and Education, ICCSE 2009*, 1056–1059. https://doi.org/10.1109/ICCSE.2009.5228542

Quinlan, J. R. (1991). Improved Estimates for the Accuracy of Small Disjuncts. *Machine Learning*, *6*(1), 93–98. https://doi.org/10.1023/A:1022646118217

Ramyachitra, D., & Manikandan, P. (2014). Imbalanced Dataset Classification and Solutions: a Review. *International Journal of Computing and Business Research (IJCBR) ISSN (Online*, *5*(4), 2229–6166.

Ramzan, M., & Ahmed, M. (2022). Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms. *IEEE Access*, *10*, 39700–39715. https://doi.org/10.1109/ACCESS.2022.3166891

Randhawa, K., Loo, C. K., Seera, M., Lim, C. P., & Nandi, A. K. (2018). Credit Card Fraud Detection Using AdaBoost and Majority Voting. *IEEE Access*, *6*, 14277–14284. https://doi.org/10.1109/ACCESS.2018.2806420

Ravisankar, P., Ravi, V., Rao, G. R., & Bose, I. (2011). Detection of fi nancial statement fraud and feature selection using data mining techniques. *Decision Support Systems*, *50*(2), 491–500. https://doi.org/10.1016/j.dss.2010.11.006

Ridgeway, G., Madigan, D., Richardson, T., & O'Kane, J. (1998). Interpretable Boosted Naïve Bayes Classification. *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, KDD 1998*.

Rifai, S., Vincent, P., Muller, X., Glorot, X., & Bengio, Y. (2011). Contractive auto-encoders: Explicit invariance during feature extraction. *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, *1*, 833–840.

Roy, A., Sun, J., Mahoney, R., Alonzi, L., Adams, S., & Beling, P. (2018). Deep Learning Detecting Fraud in Credit Card Transactions. *2018 Systems and Information Engineering Design Symposium (SIEDS)*, 129–134. https://doi.org/doi:10.1109/sieds.2018.8374722

Roy, R., & K, T. G. (2017). *DETECTING INSURANCE CLAIMS FRAUD USING MACHINE LEARNING TECHNIQUES*.

Sadgali, I., Sael, N., & Benabbou, F. (2018). Detection and prevention of credit card fraud: State of art. *MCCSIS 2018 - Multi Conference on Computer Science and Information Systems; Proceedings of the International Conferences on Big Data Analytics, Data*

Mining and Computational Intelligence 2018, Theory and Practice in Modern Computing 2018 and Connected Sma, *18*(11), 129–136.

Sahayasakila, V., Aishwaryasikhakolli, D., & Yasaswi, V. (2019). Credit card fraud detection system using smote technique and whale optimization algorithm. *International Journal of Engineering and Advanced Technology*, *8*(5), 190–192.

Sahin, Y., & Duman, E. (2011). Detecting credit card fraud by ANN and logistic regression. *INISTA 2011 - 2011 International Symposium on INnovations in Intelligent SysTems and Applications*, 315–319. https://doi.org/10.1109/INISTA.2011.5946108

Sailusha, R., Gnaneswar, V., Ramesh, R., & Ramakoteswara Rao, G. (2020). Credit Card Fraud Detection Using Machine Learning. *Proceedings of the International Conference on Intelligent Computing and Control Systems, ICICCS 2020*, *October 2020*, 1264–1270. https://doi.org/10.1109/ICICCS48265.2020.9121114

Serre, D. (2002). *Matrices : Theory and Applications* (S. Axler, F. W. Gehring, & K. A. Ribet (eds.)).

Seyedhossein, L., & Hashemi, M. R. (2010). *Mining Information from Credit Card Time Series for Timelier Fraud Detection*. 619–624.

Shariati, M., Mafipour, M. S., Ghahremani, B., Azarhomayun, F., Ahmadi, M., Trung, N. T., & Shariati, A. (2022). A novel hybrid extreme learning machine–grey wolf optimizer (ELM-GWO) model to predict compressive strength of concrete with partial replacements for cement. *Engineering with Computers*, *38*(1), 757–779. https://doi.org/10.1007/s00366-020-01081-0

Singh, Amit, Ranjan, R. K., & Tiwari, A. (2022). Credit Card Fraud Detection under Extreme Imbalanced Data: A Comparative Study of Data-level Algorithms. *International Journal of Computer Applications*, *34*(4), 8–12. https://doi.org/10.1080/0952813X.2021.1907795

Singh, Arpit, & Purohit, A. (2015). A Survey on Methods for Solving Data Imbalance Problem for Classification. *International Journal of Computer Applications*, *127*(15), 37–41. https://doi.org/10.5120/ijca2015906677

Singh, I., Kumar, N., Srinivasa, K. G., Maini, S., Ahuja, U., & Jain, S. (2021). A multi-level classification and modified PSO clustering based ensemble approach for credit scoring. *Applied Soft Computing*, *111*, 107687. https://doi.org/10.1016/j.asoc.2021.107687

Srivastava, A., Kundu, A., Sural, S., & Majumdar, A. K. (2008). Credit card fraud detection using Hidden Markov Model. *IEEE Transactions on Dependable and Secure Computing*, *5*(1), 37–48. https://doi.org/10.1109/TDSC.2007.70228

Sundarkumar, G. G., Ravi, V., & Siddeshwar, V. (2016). One-class support vector machine based undersampling: Application to churn prediction and insurance fraud detection. *2015 IEEE International Conference on Computational Intelligence and Computing Research, ICCIC 2015*, *ii*. https://doi.org/10.1109/ICCIC.2015.7435726

Supraja, K., & Saritha, S. J. (2018). Robust fuzzy rule based technique to detect frauds in vehicle insurance. *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing, ICECDS 2017*, 3734–3739. https://doi.org/10.1109/ICECDS.2017.8390160

Thach, N. H., Rojanavasu, P., & Pinngern, O. (2008). Cost-sensitive XCS classifier system addressing imbalance problems. *Proceedings - 5th International Conference on Fuzzy*

Systems and Knowledge Discovery, FSKD 2008, *2*, 132–136. https://doi.org/10.1109/FSKD.2008.391

Tian, Y., & Liu, G. (2020). MANE: Model-Agnostic Non-linear Explanations for Deep Learning Model. *Proceedings - 2020 IEEE World Congress on Services, SERVICES 2020*, 33–36. https://doi.org/10.1109/SERVICES48979.2020.00021

Tripathi, D., Edla, D. R., Kuppili, V., Bablani, A., & Dharavath, R. (2018). Credit Scoring Model based on Weighted Voting and Cluster based Feature Selection. *Procedia Computer Science*, *132*(Iccids), 22–31. https://doi.org/10.1016/j.procs.2018.05.055

Tripathi, D., Reddy, D., Kuppili, V., & Bablani, A. (2020). *Engineering Applications of Artificial Intelligence Evolutionary Extreme Learning Machine with novel activation function for credit scoring*. *96*(September 2018).

Vasek, M., & Moore, T. (2019). *Analyzing the Bitcoin Ponzi Scheme Ecosystem* (Vol. 2). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-58820-8

Viaene, S., Derrig, R. A., Baesens, B., & Dedene, G. (2002). *A Comparison of the State of the Art Classification Techniques for Expert Automobile Insurance Claim Fraud Detection*. *69*(3), 373–421.

Viaene, S., Derrig, R. A., & Dedene, G. (2004). *A Case Study of Applying Boosting Naive Bayes to Claim Fraud Diagnosis*. *16*(5), 612–620.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P. A. (2010). Stacked denoising autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *Journal of Machine Learning Research*, *11*(June 2014), 3371–3408.

Wang, X., Wu, H., & Yi, Z. (2018). Research on Bank Anti-fraud Model Based on K-means and Hidden Markov Model. *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*, 780–784. https://doi.org/10.1109/ICIVC.2018.8492795

Wang, Y., Ã, F. C., & Yuan, Y. (2011). Neurocomputing A study on effectiveness of extreme learning machine $. *Neurocomputing*, *74*(16), 2483–2490. https://doi.org/10.1016/j.neucom.2010.11.030

Wang, Z., Yu, G., Kang, Y., Zhao, Y., & Qu, Q. (2014). *Neurocomputing Breast tumor detection in digital mammography based on extreme learning machine*. *128*, 175–184. https://doi.org/10.1016/j.neucom.2013.05.053

West, J., & Bhattacharya, M. (2016). Intelligent financial fraud detection: A comprehensive review. *Computers and Security*, *57*, 47–66. https://doi.org/10.1016/j.cose.2015.09.005

Winham, S. J., Freimuth, R. R., & Biernacka, J. M. (2013). *A Weighted Random Forests Approach to Improve Predictive Performance*. https://doi.org/10.1002/sam

Wu, G., & Chang, E. Y. (2005). KBA: Kernel boundary alignment considering imbalanced data distribution. *IEEE Transactions on Knowledge and Data Engineering*, *17*(6), 786–795. https://doi.org/10.1109/TKDE.2005.95

Xu, W., & Liu, Y. (2012). An optimized SVM model for detection of fraudulent online credit card transactions. *Proceedings - 2012 International Conference on Management of e-Commerce and e-Government, ICMeCG 2012*, 14–17. https://doi.org/10.1109/ICMeCG.2012.39

Xuan, S., Liu, G., Li, Z., Zheng, L., Wang, S., & Jiang, C. (2018). Random forest for credit

card fraud detection. *ICNSC 2018 - 15th IEEE International Conference on Networking, Sensing and Control*, 1–6. https://doi.org/10.1109/ICNSC.2018.8361343

Yang, C. Y., Yang, J. S., & Wang, J. J. (2009). Margin calibration in SVM class-imbalanced learning. *Neurocomputing*, *73*(1–3), 397–411. https://doi.org/10.1016/j.neucom.2009.08.006

Yang, P., Yoo, P. D., Fernando, J., Zhou, B. B., Zhang, Z., & Zomaya, A. Y. (2014). Sample subset optimization techniques for imbalanced and ensemble learning problems in bioinformatics applications. *IEEE Transactions on Cybernetics*, *44*(3), 445–455. https://doi.org/10.1109/TCYB.2013.2257480

Yang, X. S. (2010). A new metaheuristic Bat-inspired Algorithm. *Studies in Computational Intelligence*, *284*(April 2010), 65–74. https://doi.org/10.1007/978-3-642-12538-6_6

Yu, X., Li, X., Dong, Y., & Zheng, R. (2020). A Deep Neural Network Algorithm for Detecting Credit Card Fraud. *Proceedings - 2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering, ICBAIE 2020*, 181–183. https://doi.org/10.1109/ICBAIE49996.2020.00045

Yu, Y., Gao, S., Wang, Y., & Todo, Y. (2019). Global Optimum-Based Search Differential Evolution. *IEEE/CAA Journal of Automatica Sinica*, *6*(2), 379–394. https://doi.org/10.1109/JAS.2019.1911378

Yun, Z., & Bing, L. (2008). Parallel classifiers ensemble with hierarchical machine learning for imbalanced classes. *Proceedings of the 7th International Conference on Machine Learning and Cybernetics, ICMLC*, *1*(July), 94–99. https://doi.org/10.1109/ICMLC.2008.4620385

Zadrozny, B., & Elkan, C. (2001). Learning and making decisions when costs and probabilities are both unknown. *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *July 2001*, 204–213. https://doi.org/10.1145/502512.502540

Zareapoor, M., & Shamsolmoali, P. (2015). Application of credit card fraud detection: Based on bagging ensemble classifier. *Procedia Computer Science*. https://doi.org/10.1016/j.procs.2015.04.201

Zhai, S., Chang, K. H., Zhang, R., & Zhang, Z. M. (2016). DeepIntent: Learning attentions for online advertising with recurrent neural networks. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *13-17-Augu*, 1295–1304. https://doi.org/10.1145/2939672.2939759

Zhao, S., Zhang, T., Ma, S., & Chen, M. (2022). Dandelion Optimizer: A nature-inspired metaheuristic algorithm for engineering applications. *Engineering Applications of Artificial Intelligence*, *114*(January). https://doi.org/10.1016/j.engappai.2022.105075

Zhou, Q., Zhou, H., & Li, T. (2015). Cost-sensitive feature selection using random forest : Selecting low-cost subsets of informative features. *Knowledge-Based Systems*. https://doi.org/10.1016/j.knosys.2015.11.010

Zhu, H., Liu, G., Zhou, M., Xie, Y., & Abusorrah, A. (2020). Neurocomputing Optimizing Weighted Extreme Learning Machines for imbalanced classification and application to credit card fraud detection. *Neurocomputing*, *407*, 50–62. https://doi.org/10.1016/j.neucom.2020.04.078

Zhu, H., Liu, G., Zhou, M., Xie, Y., & Kang, Q. (2019). Dandelion algorithm with probability-based mutation. *IEEE Access*, *7*, 97974–97985. https://doi.org/10.1109/ACCESS.2019.2927846

Zhu, H., Zhou, M. C., Liu, G., Xie, Y., Liu, S., & Guo, C. (2023). NUS: Noisy-Sample-Removed Undersampling Scheme for Imbalanced Classification and Application to Credit Card Fraud Detection. *IEEE Transactions on Computational Social Systems*, 1–12. https://doi.org/10.1109/TCSS.2023.3243925

Zhu, Q., Qin, A. K., Suganthan, P. N., & Huang, G. (2005). *Evolutionary extreme learning machine*. *38*, 1759–1763. https://doi.org/10.1016/j.patcog.2005.03.028

Zojaji, Z., & Atani, R. E. (2016). *A Survey of Credit Card Fraud Detection Techniques : Data and Technique Oriented Perspective*. *November*.

Zong, W., Huang, G., & Chen, Y. (2013). Weighted extreme learning machine for imbalance learning. *Neurocomputing*, *101*, 229–242. https://doi.org/10.1016/j.neucom.2012.08.010

Zou, Y., & Gao, C. (2022). Extreme Learning Machine Enhanced Gradient Boosting for Credit Scoring. *Algorithms*, *15*(5). https://doi.org/10.3390/a15050149