الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة سعيدة د. مولاي الطاهر

كلية التكنولوجيا

قسم: الإعلام الآلي

MASTER THESIS

Major: Computer security and cryptography

Theme

Nadra Start-up Part II: Market Frauds Detection using Artificial Intelligence

Presented by :

Izzeddine BENAISSA

Mohamed Chakib SEDDIKI

Supervised by :

Hadj Ahmed BOUARARA

College year 2022-2023

Dedication

*To my loving **parents**, Thank you for always standing with me. Thank you for your endless love and support, which have always been my guiding light. Thank you for teaching me the importance of hard work, perseverance, and compassion. I am so grateful to have you as my parents.*

*To my dear **siblings**, Thank you for always being there for me. You have been my biggest supporters throughout my life. I am so grateful for your love, and support.*

*To all the **teachers** who shaped my education, thank you for your dedication, knowledge, and passion. Your efforts made a profound impact on my academic journey. I'm forever grateful for your guidance and invaluable lessons.*

*In unity we thrive, together we shine. Grateful for for the collaboration we made with **Oussama Amer** & **Lahcen Redouane Mekkaoui**, without their collaboration this project couldn't be accomplished on time , thank you for sharing this journey with us.*

I dedicate this thesis to the people who have loved and supported me throughout my life. Your guidance and encouragement have helped me to achieve my goals, and I am so grateful for your presence in my life.

- Izzeddine

Dedication

*In profound gratitude and love, I dedicate this thesis to the most cherished individuals in my life. To **my parents & siblings** , you have been my unwavering source of support, inspiration, and strength. Your belief in my abilities and constant encouragement have propelled me forward, and I am forever grateful for the countless sacrifices you have made to see me succeed.*

*To my dear **brother** , your generosity in providing the PC that became the cornerstone of this thesis is a testament to your selflessness and belief in my dreams. I am forever indebted to you for your incredible kindness and support.*

*To my esteemed **teachers**, you have played a pivotal role in shaping my intellectual growth and nurturing my passion for knowledge. Your guidance, mentorship, and dedication have opened new horizons for me, and I am immensely grateful for the wisdom and expertise you have shared.*

*I extend my deepest appreciation to our esteemed collaborators, **Oussama AMER & Lahcene Redouane MEKKAOUI**. Your collaborative spirit, expertise, and invaluable contributions have greatly enhanced the quality and impact of this thesis. Working alongside you has been an enriching experience, and I am grateful for the opportunity to learn from your brilliance and dedication.*

I dedicate this thesis to the people who have loved and supported me throughout my life. Your guidance and encouragement have helped me to achieve my goals, and I am so grateful for your presence in my life.

- Mohamed Chakib

ACKNOWLEDGEMENTS

First and foremost, We express our gratitude to the Almighty Allah for providing us with the strength and resilience to complete our research.

Secondly, We would like to give a big thank you to our supervisor, ***Dr. Hadj Ahmed BOUARARA****. He has been incredibly helpful and supportive. His advice and encouragement have motivated us to do our best and meet the high expectations we were given.*

We also want to express our appreciation to the jury members for taking the time to evaluate our work. We value your presence and insights.

We are grateful to all our teachers for sharing their knowledge and helping us grow intellectually.

Furthermore, we would like to extend our heartfelt gratitude to ourselves for demonstrating remarkable patience and putting in immense effort throughout this personal journey.

Lastly, we want to thank everyone who has contributed directly or indirectly to the completion of this work. Your contributions have been crucial.

May you all be blessed and rewarded for your kindness, support, and guidance throughout this journey.

Abstract

In today's marketplace, fraudulent activities such as fake reviews, counterfeit NFTs sale distribution of fake news have become pervasive issues globally, posing significant financial risks for investors clients alike while shaking the public's trust confidence in transactions. Hence it's imperative we address these problems by implementing robust security measures reliable systems, however unfortunately such safeguards do not yet exist in North Africa, especially Algeria. Our focus is on developing an advanced fraud detection system using cutting-edge technologies like computer vision natural language processing that can analyze visual content textual data, thus detecting instances of suspicious activity related to counterfeit NFTs or fraudulent gold sales via algorithms that enable users to make informed decisions mitigate risks. Through this project we aspire toward fostering a secure environment where individuals can transact confidently without worrying about being exposed to fraudulent schemes thus protecting their interests better.

Keywords:AI, deep learning, machine learning, NLP, computer vision, market frauds, nadra

---

Résumé

Sur le marché actuel, les activités frauduleuses telles que les fausses critiques, la vente de NFT contrefaits et la distribution de fausses nouvelles sont devenues des problèmes omniprésents à l'échelle mondiale, posant des risques financiers importants pour les investisseurs et les clients tout en ébranlant la confiance du public dans les transactions. Il est donc impératif de résoudre ces problèmes en mettant en œuvre des mesures de sécurité robustes et des systèmes fiables, mais malheureusement, de telles garanties n'existent pas encore en Afrique du Nord, en particulier en Algérie. Notre objectif est de développer un système avancé de détection de fraude utilisant des technologies de pointe telles que la vision par ordinateur et le traitement du langage naturel qui peuvent analyser le contenu visuel et les données textuelles, détectant ainsi les cas d'activité suspecte liée à des NFT contrefaits ou à des ventes d'or frauduleuses via des algorithmes qui permettent aux utilisateurs pour prendre des décisions éclairées et atténuer les risques. Grâce à ce projet, nous aspirons à favoriser un environnement sécurisé où les individus peuvent effectuer des transactions en toute confiance sans se soucier d'être exposés à des stratagèmes frauduleux, protégeant ainsi mieux leurs intérêts.

Mots clés:IA, apprentissage profond, apprentissage automatique, NLP, vision par ordinateur, fraudes du marché, nadra

---

ملخص

في سوق اليوم ، أصبحت الأنشطة الاحتيالية مثل المراجعات المزيفة ، وبيع NFTs المزيفة ، وتوزيع الأخبار المزيفة ، قضايا منتشرة على مستوى العالم ، مما يشكل مخاطر مالية كبيرة للمستثمرين والعملاء بينما يقوض ثقة الجمهور في المعاملات. لذلك من الضروري حل هذه المشاكل من خلال تنفيذ تدابير أمنية قوية وأنظمة موثوقة ، ولكن للأسف ، مثل هذه الضمانات غير موجودة حتى الآن في شمال إفريقيا ، وخاصة في الجزائر. هدفنا هو تطوير نظام متقدم للكشف عن الاحتيال باستخدام أحدث التقنيات مثل رؤية الكمبيوتر ومعالجة اللغة الطبيعية التي يمكنها تحليل المحتوى المرئي والبيانات النصية ، وبالتالي الكشف عن حالات النشاط المشبوه المتعلقة بـ NFTs المزيفة أو مبيعات الذهب الاحتيالية عبر الخوارزميات التي تمكن المستخدمين من اتخاذ قرارات مستنيرة وتخفيف المخاطر. من خلال هذا المشروع ، نطمح إلى تعزيز بيئة آمنة حيث يمكن للأفراد التعامل بثقة دون القلق بشأن التعرض لمخططات احتيالية ، وبالتالي حماية مصالحهم بشكل أفضل.

كلمات مفتاحية: الذكاء الاصطناعي ، التعلم العميق ، التعلم الآلي ، البرمجة اللغوية العصبية ، رؤية الكمبيوتر، عمليات الاحتيال في السوق

Contents

List of Figures

List of Tables

List of Acronyms

AI	*Artificial Intelligence*
DL	*Deep learning*
ML	*Machine Learning*
OF	*Overfitting*
UF	*Underfitting*
NN	*Neural Networks*
ANN	*Artificial Neural Networks*
SNN	*Spiking Neural Networks*
ASR	*Automated Speech Recognition*
DNN	*Deep Neural Networks*
DBN	*Deep Belief Network*
Adam	*Adaptive Moment Estimation*
AdamW	*Adam with Weight Decay*
RMSprop	*Root Mean Square Propagation*
SGD	*Stochastic Gradient Descent*
ReLu	*Rectified Linear Unit*
SVM	*Support Vector Machine*
CNN	*Convolutional Neural Network*
FCNN	*Fully connected Neural Network*
RNN	*Recurrent Neural Network*
GAN	*Generative Adversarial Network*
GPT-2	*Generative Pre-trained Transformer*

BERT *Bidirectional Encoder Representations from Transformers*

DINO *Distributed Input Neural Oracle*

SAM *Segment Anything Model*

NLP *Natural Language Processing*

GloVe *Global Vectors for Word Representation*

NFT *Non-Fungible Token*

Chapter 1
General Introduction

Chapter 1

General introduction

1.1 Introduction

1.1.1 Context

Frauds in the marketplace are a pervasive issue affecting global economies. This problem is particularly pronounced in North Africa, and specifically in Algeria, where there is a notable lack of effective tools to combat fraudulent activities. To address this critical challenge, our project places a strong emphasis on leveraging advanced technologies such as computer vision and natural language processing.

By harnessing the power of computer vision, we can detect and analyze fraudulent activities in visual content, such as counterfeit products or manipulated images. Additionally, employing natural language processing enables us to extract meaningful insights from textual data, including customer reviews, social media posts, and online conversations. These technologies combined allow us to develop a comprehensive platform that assists both investors and regular clients in mitigating potential losses and risks in the marketplace.

Our platform serves as a valuable resource, providing real-time fraud detection, price trend prediction, and risk assessment functionalities. By utilizing cutting-edge AI algorithms, we empower users to make informed decisions based on accurate data analysis and predictions. This helps them navigate the marketplace with increased confidence, minimizing the chances of falling victim to fraudulent schemes and reducing potential financial losses.

Through our commitment to combating fraud and protecting market participants, particularly in Algeria and across North Africa, we aim to contribute to the overall stability and trustworthiness of the marketplace. By offering a robust platform built on computer vision and natural language processing, we strive to empower investors and clients with the tools they need to mitigate risks and make secure and informed investment decisions.

1.1.2 Motivation

The motivation behind our project stems from the widespread prevalence of fraud in marketplaces worldwide, with a particular focus on addressing the issue in North Africa, especially Algeria. We are driven by the desire to provide a solution that effectively combats fraud and helps investors and regular clients safeguard their interests. By leveraging advanced technologies such as computer vision and natural language processing, we aim to empower users with the tools and insights necessary to reduce their losses and mitigate risks in the marketplace.

1.1.3 Problematic

The problem we aim to address is the lack of adequate tools and resources to combat fraud in the marketplace, specifically in North Africa and Algeria. The absence of robust fraud detection and risk prediction mechanisms leaves investors and clients vulnerable to financial losses and deceitful practices. The prevailing fraudulent activities undermine trust, hinder economic growth, and hinder the overall stability of the marketplace.

Moreover, traditional methods of fraud detection often fall short in identifying increasingly sophisticated fraudulent schemes, making it imperative to leverage advanced technologies to effectively combat these challenges. The absence of comprehensive platforms that integrate computer vision and natural language processing exacerbates the problem and leaves investors and clients without the means to make informed decisions.

By tackling this problematic, our project strives to fill the existing gap and provide a reliable and efficient solution that helps users navigate the marketplace with confidence. We aim to equip investors and clients with the necessary tools and insights to detect fraud, and assess risks, thereby reducing their exposure to fraudulent activities and enabling them to make informed investment decisions.

1.1.4 Objective

Our main objective is to develop an advanced fraud detection system that effectively identifies and prevents fraudulent activities within the marketplace. With frauds prevalent worldwide, we recognize the need to address this issue, especially in North Africa and Algeria in areas related to e-commerce and investment activities. By harnessing the power of computer vision and natural language processing technologies, we aim to create a robust and reliable platform that can accurately detect and flag instances of fraud in visual content and textual data.

Our goal is to enhance user security and trust by proactively addressing fraudulent activities. By leveraging AI algorithms and machine learning techniques, our system will analyze and interpret complex data patterns to identify potential fraudulent behavior. This will enable us to provide users with timely insights and alerts, empowering them to take proactive measures to protect their investments and minimize potential financial losses.

In pursuing this objective, we understand the importance of collaboration with industry experts. By establishing partnerships, we aim to stay updated on emerging fraud techniques, and best practices in fraud detection and prevention. This collaborative approach will help us continuously improve our fraud detection system and ensure that our platform adheres to industry standards and regulations.

Additionally, we are committed to educating our users about fraud prevention. Through informative resources, educational materials, and interactive content, we will raise awareness about common fraud schemes, red flags to watch out for, and best practices for avoiding fraudulent activities. By empowering our users with knowledge and resources, we aim to create a community of vigilant and informed individuals who can make sound decisions and protect themselves from falling victim to fraud.

Furthermore, we recognize the need for continuous innovation and adaptation to combat the evolving nature of fraud. We are dedicated to ongoing research and development to enhance our fraud detection system, leveraging emerging technologies and techniques to stay one step ahead of fraudsters. By staying at the forefront of technological advancements, we aim to provide our users with a cutting-edge platform that effectively safeguards their interests and fosters trust within the marketplace.

Overall, our objective is to create a secure and trustworthy marketplace by developing an advanced fraud detection system, collaborating with industry experts, educating users about fraud prevention, and continuously innovating to stay ahead of fraudulent activities. Through these efforts, we strive to empower investors and regular clients, enabling them to make informed decisions and minimize their risk exposure in the marketplace.

1.1.5 The roadmap of our work

- Designing new version models for sequential data and second for images data (NadraBERT., nadraGPT-2, Nadra Embedding Convolutional network, Incoherent Pixels Technique).
- Using benchmarks to validate our solutions and making comparisons with existing techniques in the literature.
- Self-learning through the use of data scraping and data community.
- Building our own language model by testing our solution in a general sequential context (fake news and fake reviews) which will automatically give good results face to downsampling problems such as: dropshipping fraud, Initial Coin Offering (ICO) scams, Manipulating cryptocurrency prices, Fake blockchain projects, Charity scams, counterfeit products fraud, ponzi fraud, Auction or sales fraud.
- Building of our own pretrained model by training our solution on complex problems such as fake NFT and fake gold which will automatically gives good results face to other types of fraud related to data image.
- Store our models in order to apply them to real cases in Algeria and Arab countries
- Integrate our solutions into a nadra application in two desktop and mobile versions

1.1.6 Organization of work

The thesis is meticulously structured into five distinct chapters, each dedicated to examining a specific aspect of the research in depth. The following provides a concise overview of the subjects covered in each chapter:

1.1.6.1 Chapter 1: General Introduction

The first chapter serves as a general introduction to the research topic, delving into the context of Frauds and their impact on the marketplace. It highlights the necessity for advanced computational approaches and outlines the research goals and objectives, which encompass the analysis of diverse datasets, the application of deep learning and machine learning algorithms, and the integration of models to enhance fraud detection accuracy.

1.1.6.2 Chapter 2: Marketplace Frauds

This chapter delves into the realm of marketplace frauds. It explores various methods and techniques employed in fraud detection. Additionally, this chapter presents compelling case studies and real-world applications to illustrate the practical implications of fraud detection methodologies.

1.1.6.3 Chapter 3: Artificial Intelligence

This chapter presents an extensive exploration of artificial intelligence, encompassing a wide range of vital topics. It begins by introducing the foundational principles of machine learning and deep learning, equipping readers with a solid grasp of key concepts and techniques. Within this chapter, valuable insights into neural networks, including CNNs, RNNs, and Transformers, are provided. Serving as a comprehensive guide, this chapter empowers readers to navigate through fundamental concepts, models, and architectures in the field of artificial intelligence.

1.1.6.4 Chapter 4: Contribution

This chapter highlights the contributions made by this study, focusing on the system architecture and developed models. The primary objective is to offer a comprehensive and theoretical explanation of the project, emphasizing the conceptual aspects rather than the intricate implementation details in code. Specifically, the techniques employed in each model, such as detecting fake reviews, fake news, fake gold, and fake NFTs, are introduced. By elucidating the framework and models, this chapter showcases unique contributions to the field and provides insights into the methodologies employed in this research endeavor.

1.1.6.5 Chapter 5: Result Discussion and Experimentation

This chapter of this research is dedicated to result discussion and experimentation. It centers around the core objective of conducting experiments to test, evaluate, and discuss the outcomes of the developed predictive models. This chapter offers a comprehensive description of the experimentation process and analyzes the performance of these models in terms of accuracy and other pertinent metrics. Furthermore, it serves as a comprehensive overview of the experimental results, providing valuable insights for future research in the domain of AI-based fraud detection.

1.1.6.6 Chapter 6: Nadra's Platform

Chapter 6 focuses on the development of Nadra's prototype, which is a platform designed to provide customers with subscription-based services. This chapter explores the key features, functionalities, and technologies employed in creating the prototype, with a particular emphasis on the integration of a chatbot.

1.1.6.7 Chapter 7: General Conclusion

The concluding chapter of this work offers a comprehensive overview and general conclusion. It reflects on the research challenges encountered throughout the study and outlines potential avenues for future exploration. Providing a concise summary of the main findings and contributions, this chapter underscores the significance of the research and establishes a foundation for further advancements in the field. By encapsulating the key takeaways from the thesis, it concludes the work on a strong note, highlighting the implications and potential for future developments.

Chapter 2
Marketplace Frauds

Chapter 2

Marketplace frauds

Deep learning has shown great promise in enhancing cybersecurity measures in various domains, such as network security, malware detection, and intrusion detection. Deep learning algorithms can analyze large amounts of data and identify complex patterns that traditional security measures may miss. For instance, deep learning models have been used to detect zero-day malware and phishing attacks with high accuracy. Furthermore, deep learning models can learn and adapt to evolving cybersecurity threats, making them well-suited for dynamic and complex environments. Deep learning techniques have also been applied in the financial industry to analyze market data and identify patterns for better investment decisions. For example, deep learning models have been used to predict stock prices and detect trading anomalies . The application of deep learning in the financial industry has the potential to improve market efficiency and reduce risk. However, the use of deep learning also poses ethical and security concerns, such as the potential for bias and the risk of cyber attacks on deep learning systems. Therefore, it is crucial to ensure that deep learning models are designed and implemented with appropriate safeguards and transparency measures.

2.1 Introduction

Market frauds can take many forms, from simple deception to complex schemes that can be difficult to detect. Fake gold is one such example, where gold plated items or alloys are sold as pure gold [24]. Similarly, fake news and rumors can be spread to manipulate markets or individual securities. In addition, e-commerce marketplaces have seen an increase in counterfeit and fake products, leading to customer dissatisfaction and financial losses. These types of frauds not only harm individual consumers but also have broader economic impacts, including reduced consumer confidence and increased regulatory costs. Therefore, market participants must remain vigilant and take appropriate measures to prevent and detect fraudulent activities in all forms.

2.2 Gold frauds

2.2.1 Counterfeit gold

One of the most counterfeited precious metals is gold. Copper has the same hue as gold. As a result, copper is one of the most commonly utilized materials in color counterfeiting. Where the chemical characteristics are concerned, wolfram is similar gold (density of gold and tungsten

are 19.30 g/ml and 19.25 g/ml, respectively), hence it may be used as a chemical counterfeit. Gold purity can be tested using X-ray, however this procedure is so expensive. [7]

2.2.1.1 Existing model

For the sound processing module, decision trees, KNN, neural networks, and SVM classifiers were used. In the image processing module the most popular data mining approaches for classification are described in this solution. Support Vector Machine (SVM), Decision Tree, a Deep Neural Network (one output, one input, and two hidden dense layers), and CNN were the four algorithms that were selected. In order to run the first three methods, 2D matrices must first be converted to 1D arrays. CNN is immediately applied on the 2D pictures. These algorithms were chosen because they each stand for a distinct category, here are the used architectures [7]:

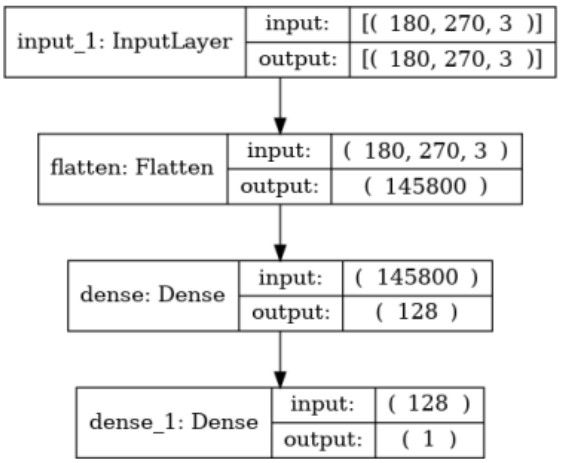


Figure 2.1: The CNN architecture used for classifying precious metals. Figure reproduced from [7]

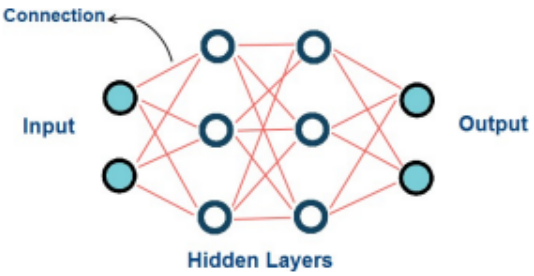


Figure 2.2: ANN Model Structure. Figure reproduced from [7]

2.3 Dropshipping frauds

2.3.1 Fake reviews

Fake reviews are a growing problem in e-commerce, where customers increasingly rely on product reviews to make purchasing decisions. These reviews can be written by individuals with no real experience of the product or by the product's competitors, seeking to damage its reputation. In some cases, companies may even pay people to write fake positive reviews to boost their ratings. These practices can deceive customers and harm the reputation of genuine products. Furthermore, they undermine the trust customers have in e-commerce platforms, making it harder for them to differentiate between genuine and fake reviews. It is crucial for e-commerce platforms to take steps to combat fake reviews, such as implementing algorithms that detect suspicious reviews and verifying the identity of reviewers. Customers should also be encouraged to leave honest reviews and report any suspected fake reviews they come across. [14]

2.3.1.1 Existing model

Because of the growing amount of Internet purchases, fake consumer review detection has gained a lot of attention in recent years. Current ways to detect false customer reviews rely on review content, product and reviewer information, and other factors. Yet, as evidenced by recent research, the semantic meaning of reviews may be very essential for text categorization. Furthermore, the emotions disguised in the evaluations might be another evidence of fraudulent material. We present two neural network models that use standard bag-of-words as well as word context and customer emotions to improve the effectiveness of false review identification. The models specifically learn document-level representation by combining three sets of features: (1) n-grams, (2) word embeddings, and (3) distinct lexicon-based mood indicators. A high-dimensional feature representation of this type is used to categorize false reviews into four areas. We compare the classification performance of the provided detection systems with numerous state-of-the-art approaches for fake review detection to illustrate their efficacy. The suggested methods outperform on all datasets, regardless of sentiment polarity or product category. [14]

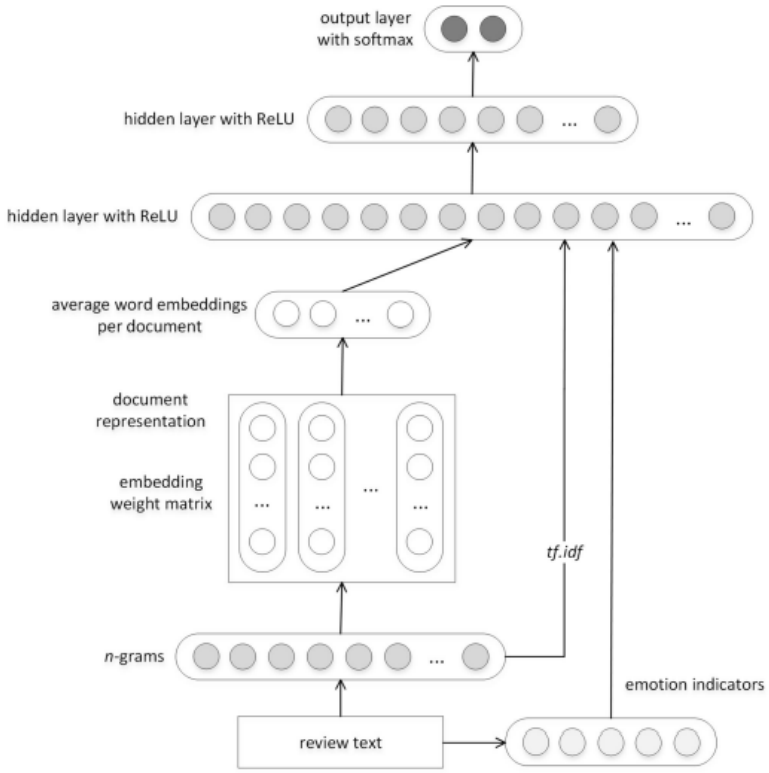


Figure 2.3: DFFNN model for fake review detection. Figure reproduced from [14]

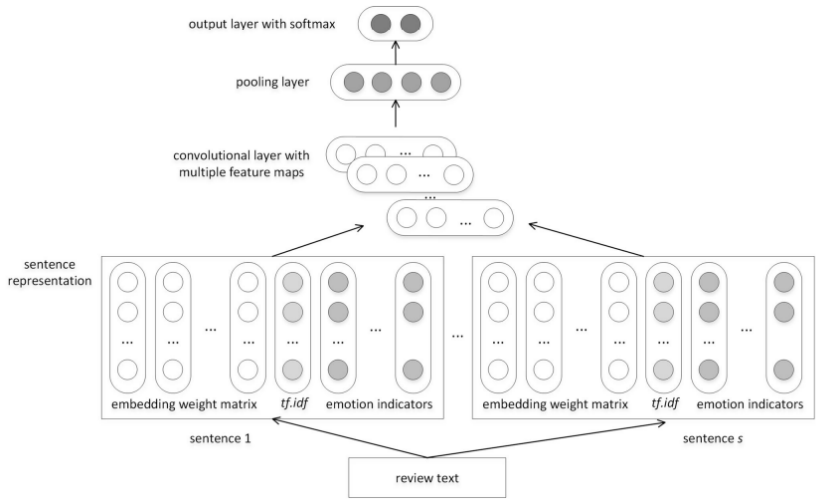


Figure 2.4: CNN model for fake review detection. Figure reproduced from [14]

2.3.1.1.1 Experimental results

Here are the results of the experimentation:

2.3.2 Fake comments

Fake comments are a common problem in e-commerce, where unscrupulous vendors or their affiliates attempt to manipulate customer perceptions by posting positive reviews that are not genuine. These reviews may be written by the vendors themselves or by third-party individuals

	Hotel dataset			Restaurant dataset		
	Acc (%)	AUC	F-score	Acc (%)	AUC	F-score
Baseline methods						
SVM[a]	80.75 ± 3.12	0.807 ± 0.031	0.808 ± 0.031	80.34 ± 7.22	0.803 ± 0.090	0.809 ± 0.069
NB	81.25 ± 3.29	0.850 ± 0.042	0.817 ± 0.031	80.58 ± 3.38	0.832 ± 0.042	0.813 ± 0.031
Bagging	78.19 ± 4.90	0.857 ± 0.041	0.781 ± 0.050	77.09 ± 6.68	0.828 ± 0.061	0.766 ± 0.069
k-NN	71.38 ± 2.99	0.772 ± 0.031	0.678 ± 0.047	72.14 ± 6.93	0.788 ± 0.074	0.692 ± 0.093
AdaBoost	77.06 ± 2.38	0.842 ± 0.028	0.771 ± 0.027	74.38 ± 6.64	0.837 ± 0.063	0.749 ± 0.060
RF	79.31 ± 2.91	0.873 ± 0.027	0.798 ± 0.028	76.62 ± 3.92	0.861 ± 0.037	0.770 ± 0.040
NN methods						
$DFFNN_{ngram}$ [10]	88.19 ± 2.15	**0.951 ± 0.014**	0.882 ± 0.022	88.31 ± 3.91	0.938 ± 0.038	0.887 ± 0.035
SCNN [43]	86.44 ± 2.41	0.939 ± 0.020	0.863 ± 0.023	89.30 ± 5.76	0.952 ± 0.041	0.898 ± 0.054
$DFFNN_{skipgram}$ [10]	83.00 ± 4.06	0.908 ± 0.025	0.831 ± 0.042	71.67 ± 7.14	0.788 ± 0.058	0.709 ± 0.081
CNN_{cbow} [61]	84.88 ± 3.25	0.911 ± 0.026	0.850 ± 0.032	79.61 ± 7.86	0.889 ± 0.055	0.803 ± 0.064
DFFNN (this study)	**89.56 ± 3.01**	**0.951 ± 0.018**	**0.896 ± 0.029**	88.31 ± 4.71	0.953 ± 0.030	0.884 ± 0.047
CNN (this study)	87.25 ± 1.70	0.945 ± 0.014	0.872 ± 0.015	**89.80 ± 6.16**	**0.965 ± 0.028**	**0.901 ± 0.057**

The best results are in bold
[a]Obtained for $C = 2^4$

Table 2.1: Results of the experiments for the hotel and restaurant datasets. Table reproduced from [14]

	Doctor dataset			Amazon dataset		
	Acc (%)	AUC	F-score	Acc (%)	AUC	F-score
Baseline methods						
SVM[a]	85.31 ± 6.65	0.838 ± 0.071	0.886 ± 0.053	76.25 ± 3.85	0.762 ± 0.038	0.760 ± 0.029
NB	81.02 ± 3.90	0.827 ± 0.054	0.853 ± 0.028	59.21 ± 0.92	0.633 ± 0.012	0.617 ± 0.009
Bagging	70.40 ± 7.72	0.752 ± 0.106	0.792 ± 0.051	80.41 ± 0.58	0.861 ± 0.007	0.793 ± 0.006
k-NN	71.13 ± 3.85	0.716 ± 0.049	0.786 ± 0.025	75.97 ± 0.83	0.804 ± 0.009	0.756 ± 0.009
AdaBoost	69.73 ± 5.71	0.726 ± 0.054	0.774 ± 0.049	79.22 ± 0.82	0.846 ± 0.010	0.782 ± 0.009
RF	75.07 ± 5.59	0.812 ± 0.061	0.831 ± 0.034	59.35 ± 0.98	0.624 ± 0.015	0.595 ± 0.011
NN methods						
$DFFNN_{ngram}$ [10]	86.19 ± 5.71	0.931 ± 0.034	0.894 ± 0.044	81.98 ± 0.79	0.881 ± 0.005	0.813 ± 0.010
SCNN [43]	87.81 ± 3.94	0.925 ± 0.036	0.906 ± 0.031	80.62 ± 0.62	0.863 ± 0.007	0.798 ± 0.007
$DFFNN_{skipgram}$ [10]	64.16 ± 0.89	0.646 ± 0.066	0.781 ± 0.006	78.85 ± 1.00	0.860 ± 0.009	0.777 ± 0.011
CNN_{cbow} [61]	77.96 ± 7.68	0.818 ± 0.100	0.839 ± 0.048	79.64 ± 0.75	0.867 ± 0.008	0.786 ± 0.009
DFFNN (this study)	86.21 ± 3.93	0.932 ± 0.030	0.893 ± 0.028	**82.80 ± 0.50**	**0.893 ± 0.006**	**0.825 ± 0.005**
CNN (this study)	**88.35 ± 3.29**	**0.946 ± 0.025**	**0.910 ± 0.026**	81.30 ± 0.72	0.879 ± 0.008	0.806 ± 0.009

The best results are in bold
[a]Obtained for $C = 2^3$ and $C = 2^5$ for the doctor and Amazon dataset, respectively

Table 2.2: Results of the experiments for the doctor and Amazon datasets. Table reproduced from [14]

paid to write positive comments. The purpose of these fake comments is to boost the seller's reputation, increase sales, and attract new customers. However, such practices are unethical and can lead to serious consequences for both the vendors and the customers. Fake comments can mislead consumers into buying products that may not live up to their expectations, leading to disappointment and mistrust. Moreover, fake comments can harm the credibility of e-commerce platforms, undermining the trust that customers have in online shopping. To address this problem, e-commerce platforms have implemented various measures to detect and remove fake comments, such as using automated algorithms to identify suspicious patterns, verifying the authenticity of user accounts, and manually reviewing comments. It is essential that e-commerce platforms continue to improve their efforts to combat fake comments to ensure that customers can make informed purchasing decisions based on honest and reliable feedback. [35]

2.3.2.1 Existing model

Merchants will employ professional writers to produce positive evaluations for their items or negative reviews for rivals for commercial reasons, which has a major negative influence on the ecological growth of e-commerce platforms. The feature set of product reviews is used as an entrance point in this research, and a deep-confidence network technique based on deep learning is used to assess and detect the authenticity of product evaluations for e-commerce

Methods	Aver. ranking	*p* value (vs. CNN)
SVM	7.00	0.050*
NB	7.50	0.031*
Bagging	8.25	0.014*
K-NN	10.5	0.001*
AdaBoost	9.75	0.002*
RF	9.25	0.004*
$\text{DFFNN}_{n\text{-gram}}$ [6]	2.88	0.731
SCNN [20]	3.00	0.695
$\text{DFFNN}_{\text{skipgram}}$ [6]	9.50	0.003*
CNN_{cbow} [27]	6.25	0.096
DFFNN(this study)	2.13	0.961
CNN(this study)	2.00	-
Friedman *p* value	0.0003*	

Table 2.3: Results of Friedman nonparametric test [14]

transactions. The features of regular customers' remarks are examined using model recognition. A DBN (Deep Belief Network) model shown in Fig 2.5 is built to detect bogus reviews using deep learning approaches. The model mines deep semantic features using LSTM (Long Short Term Memory) and bidirectional GRU (Gated Recurrent Unit), and CNN (Convolutional Neural Network) processes the typical discrete features recovered in this article, which are multidimensional. To build a DBN model, we link semantic and conventional properties. When the model's accuracy is verified and compared to other shallow machine learning methods, it is discovered that the deep confidence network's recognition accuracy for comment data is much greater than that of other shallow machine learning techniques. This validates the efficacy of the model suggested in this research. [35]

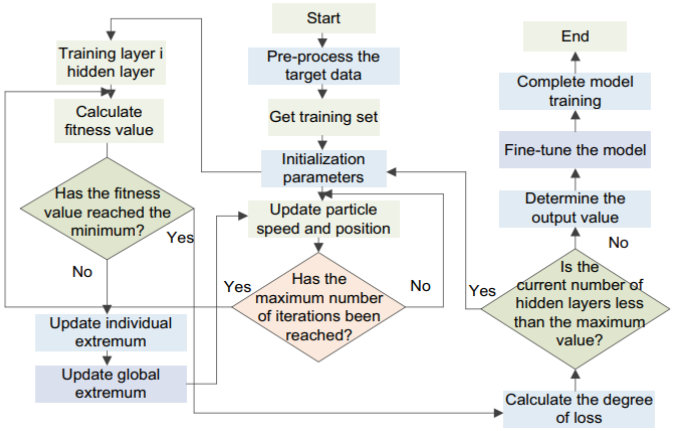
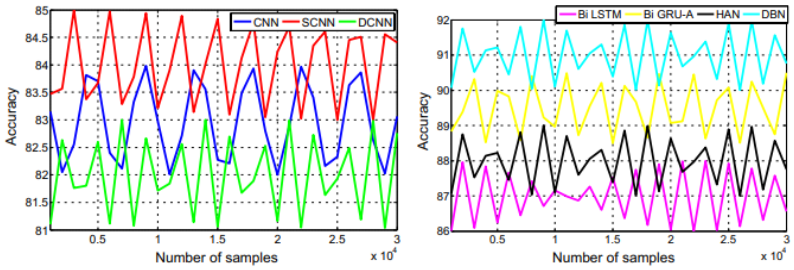


Figure 2.5: DBN hidden layer determination process. Figure reproduce from [35]

2.3.2.1.1 Experimental results

Fig. 2.6 displays the classification experiment's outcomes based on the integrated feature index. The experimental findings of the F1 value are primarily examined due to the unbalanced

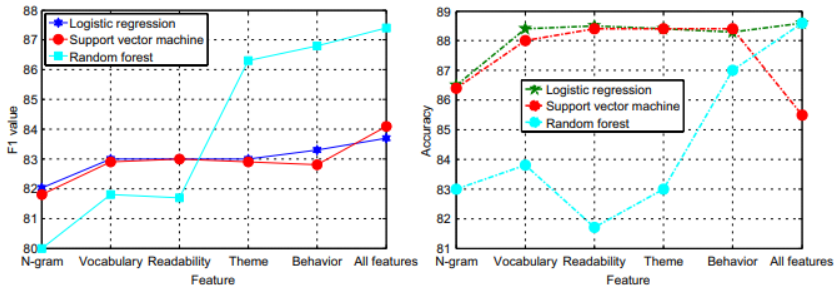
nature of the data. Comparing the trial findings, it can be shown that behavioral features have a significantly greater classification effect than readability and topical features, with a maximum performance of 87.35%. It is clear that behavioral factors have a far stronger classification impact than readability and topic criteria. The quality of the remaining three elements, however, is all over 80%. Moreover, subject characteristics perform somewhat better than readability and N-gram features. The article's suggested model framework is DBN. We employ CNN network to extract discrete feature information from reviews and LSTM and GRU models to extract content features from reviews. The outcomes of the experiment are displayed in Fig. 2.7. [35]



(a) CNN and its extended models (SCNN, DCNN)

(b) Bi LSTM, Bi GRU-A, HAN and DBN models

Figure 2.6: Classifcation experiment results based on integrated feature indicators. Figure reproduced from [35]



(a) The experimental result of F1 value

(b) The accuracy of the experiment

Figure 2.7: Experimental classifcation results based on deep learning methods under diferent sample sizes. Figure reproduced from [35]

2.3.3 Fake news

The term "Fake News" is employed to describe false information or propaganda deliberately disseminated through both traditional media outlets such as print and television, and non-traditional platforms like social media. The primary purpose behind spreading such news is to deceive readers, harm the reputation of individuals or organizations, or capitalize on sensationalism. It is widely recognized as a significant peril to democracy, open discussions, and the principles of the Western society. [31]

2.3.3.1 Counterfeit Products

Fraudsters can use fake images to promote counterfeit products. They may display images of genuine branded products, but the actual items being sold are low-quality replicas. Buyers

may be lured by the well-known brand image, only to receive a substandard and potentially unsafe counterfeit product.

2.3.3.2 Auction or sales fraud

In the context of online auctions or sales platforms, fraudsters can spread fake news about certain products or sellers to deceive buyers. They may create false narratives about high demand, limited availability, or exceptional quality to drive up prices or create a sense of urgency for buyers to make quick purchases. The products may turn out to be counterfeit or non-existent, leaving buyers defrauded.

2.3.3.3 Investment scams(ponzi fraud)

Fraudsters might create fake news articles or press releases promoting a particular investment opportunity, such as a cryptocurrency or a stock. The articles could contain fabricated information about significant returns or endorsements from influential individuals. By spreading this fake news, they aim to entice unsuspecting investors to put their money into fraudulent schemes

2.3.3.4 Charity scams

Fake news can be used to exploit people's goodwill and generosity. Fraudsters might create fabricated stories about a charitable cause, such as a natural disaster or a health crisis, and spread them through news articles or social media. They would then ask for donations to support the cause, but in reality, the funds go directly to the fraudsters rather than to any legitimate charitable organization.

2.3.3.5 Initial Coin Offering (ICO) scams

Fraudsters may create fake news articles or press releases about a new ICO, claiming that it offers a revolutionary blockchain-based product or service. They might fabricate information about partnerships, endorsements, or potential returns on investment to entice individuals to invest in the ICO. In reality, the ICO may be fraudulent or non-existent, leading to financial losses for investors.

2.3.3.6 Pump and dump schemes

In pump and dump schemes, fraudsters artificially inflate the price of a low-volume cryptocurrency through the spread of fake news, social media campaigns, or online forums. They create hype and positive sentiment around the cryptocurrency, attracting unsuspecting investors who buy the cryptocurrency at the inflated price. Once the price reaches a peak, the fraudsters sell their holdings, causing the price to plummet and leaving other investors with losses.

2.3.3.7 Fake blockchain projects

Fraudsters may create fake news stories about a blockchain project, claiming it has a groundbreaking solution or technology. They might create an elaborate website, whitepaper, and social media presence to make the project appear legitimate. By spreading false information and misleading potential investors, they can collect funds through token sales or investments, only to disappear with the funds once their fraudulent operation is revealed

2.3.3.8 Existing model

We experimented with various word vector representations and neural network architectures. Our most successful model utilizes Tf-Idf word vector representations combined with preprocessed engineered features as inputs. It employs a dense neural network architecture to predict the target stance. The input features for our model consist of Tf-Idf word vector representations of article-headline pairs, the cosine similarity between article-headline pairs represented using Tf-Idf, and the cosine similarity between article-headline pairs represented using Google's Word2Vec. We computed Tf-Idf scores for both unigrams and bigrams. To address dataset imbalance, we excluded words that appeared in more than 50% of training documents and words that appeared in less than 50 documents. Our model aims to capture the relative importance of words in article-headline pairs both locally (specific to each pair) and globally (in relation to the entire corpus). To measure the similarity between headline-article pairs, we calculated the cosine similarity using Tf-Idf representations. Due to the size of our vocabulary and the imbalanced nature of our data, there is a risk of overfitting in neural network models, which leads to inaccurate predictions on unseen test sets. To address this, we employed regularization techniques such as L2 regularization, dropout, and early stopping to overcome overfitting and improve generalization. [31]

2.3.3.8.1 Experimental result

Following an extensive hyperparameter tuning process on our most successful model, we proceeded to assess its performance using test data. Our main objective was to precisely gauge the proximity between the predicted stance and the original one, leading us to select 'classification accuracy' as our evaluation metric. The obtained accuracy results for the models can be found in Table 2.4.

Variations	Accuracy
TF-idf on unigrams and bigrams with cosine similarity fed into dense neural network	94.31%
BoW without unigrams and bigrams with cosine similarity fed into dense neural network	89.23%
Pre-trained embedding (Word2Vec) fed into dense neural network	75.67%

Table 2.4: Accuracy for different Model variation [31]

Our model achieves an accuracy of 94.21% by passing Tf-Idf word vector representations through a dense neural network. In Table 2.5, we compare the accuracy of our model with other models discussed in the literature. Our model outperforms the second best model, albeit by a small margin.

Model Description	Accuracy
TF-idf on unigrams and bigrams with cosine similarity fed into dense neural network	94.31%
BoW with multilayer perceptron [8]	89.23%
BoW with cosine similarity fed into dense neural network [30]	88.46%

Table 2.5: Comparison between best performing model and our model. Figure reproduced from [31]

Acknowledging the inherent imbalance of the FNC-1 dataset, we place significant importance on ensuring our model demonstrates satisfactory performance when handling minority stances. [31]

2.3.4 Fake NFTs

NFT frauds in the marketplace have raised concerns in recent times. The growing popularity of NFTs as a means of collecting digital art and investment has led to high-value asset sales and increased trading volumes. However, these marketplaces have received limited security scrutiny compared to decentralized finance (DeFi) protocols. Academic research has primarily focused on DeFi attacks and smart contract vulnerabilities, leaving the NFT ecosystem relatively unexplored.

2.3.4.1 Existing Model

This research focuses on the development of an audio watermarking technique to prevent the creation of fake Non-fungible Tokens (NFTs) by reusing the same audio in NFT marketplaces.

The proposed model's robustness is assessed through the calculation of the bit error rate (BER) of the embedded watermark. The quality of the watermarked audio is evaluated using Objective Difference Grade (ODG) and Signal-to-Noise Ratio (SNR) metrics. The results demonstrate that the embedded watermark can withstand low pass filtering, resampling, and MP3 compression attacks, with a maximum BER of 0.075. The proposed approach produces high-quality watermarked audio, achieving the best ODG of -0.21 and SNR of 34.36 dB. Additionally, the average execution time for similarity assessment is 1.3 seconds. [5]

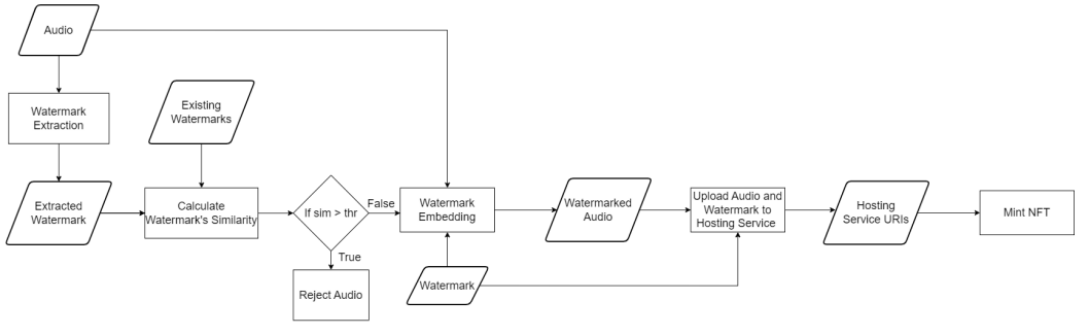


Figure 2.8: Proposed model flow chart [5]

2.4 Methods' summary

Author(s)	Paper Title	Year
Yekta Said Can	CLASSIFICATION OF ORIGINAL AND COUNTERFEIT GOLD MATTERS BY APPLYING DEEP NEURAL NETWORKS AND SUPPORT VECTOR MACHINES	2022
Lichun Zhou1 and Qian Zhang	RETRACTED ARTICLE: Recognition of false comments in E-commerce based on deep learning confidence network algorithm	2020
Petr Hajek1, Aliaksandr Barushka1 and Michal Munk2	Fake consumer review detection using deep neural networks integrating word embeddings and emotion mining	2019
Aswini Thota, Priyanka Tilak, Simrat Ahluwalia, and Nibrat Lohia	Fake News Detection: A Deep Learning Approach	2018
Ansori, Muhammad Rasyid Redha, Alief, Revin Naufal, Igboanusi, Ikechi Saviour, Lee, Jae Min, Kim, Dong-Seong, and others	Watermarking-based Fake Audio NFT Detection in NFT Marketplace	2023

Table 2.6: Methods' summary

2.5 Conclusion

In conclusion, this chapter provided an exploration of market frauds and the existing models used to detect and prevent them. We discussed various types of market frauds, including fake news, fake NFT, fake gold and fake reviews, highlighting their impact on the financial ecosystem. We also examined different models and approaches utilized to detect and mitigate market frauds. By leveraging these models and continually refining them, we can enhance our ability to identify and combat market frauds, thereby fostering a more secure and trustworthy marketplace for investors and participants.

Chapter 3
Artificial intelligence

Chapter 3

Artificial intelligence

Artificial intelligence has revolutionized cybersecurity by leveraging deep learning algorithms in areas like network security, malware detection, and intrusion prevention. It excels at analyzing data, identifying complex patterns, and adapting to evolving threats. In finance, AI's deep learning techniques aid in analyzing market data, predicting stock prices, and detecting anomalies. However, ethical concerns and security risks must be addressed. Safeguards, transparency, and accountability are necessary to mitigate biases, cyber attacks, and ensure responsible AI deployment. Overall, AI's impact on cybersecurity and decision-making is significant, but it requires careful consideration and implementation of appropriate measures to maximize benefits and minimize risks.

3.1 Machine Learning (ML)

Machine learning falls within the domain of artificial intelligence, signifying that computer algorithms possess the capability to autonomously acquire knowledge from fresh data and adjust their performance accordingly, devoid of human intervention. The process entails constructing a mathematical model utilizing sample data, commonly referred to as "training data," with the objective of making predictions or decisions without the need for explicit programming to accomplish the task [12]. Machine learning can be divided into several categories, can be supervised earning,unsupervised learning or reinforcement learning.

3.1.1 Overfitting (OF) and Underfitting (UF)

These two factors correspond to the primary challenges encountered in machine learning: underfitting and overfitting. Underfitting arises when the model fails to achieve a low enough error rate on the training set. Conversely, overfitting occurs when the discrepancy between the training error and the test error is excessively large . [17] The issue with underfitting lies in its simplistic equation, which is unable to adequately capture all the data points. Consequently, it struggles to make accurate predictions for future values, resulting in a significant loss function. On the other hand, overfitting is characterized by an excessive adjustment of the model to fit all the available data points, which can be detrimental to prediction accuracy due to the inclusion of unhelpful and redundant information.

3.1.2 Supervised learning

The process of training a model involves providing input data along with corresponding correct output data, commonly known as "labeled data." In this context, the term "labeled data" refers to the input/output pairs. The learning algorithm is supplied with labeled data, allowing the computer to establish relationships and patterns within the data in order to predict outputs when presented with new input. Supervised learning is frequently employed to construct machine learning models for two types of problems: regression and classification. In regression, the model's outputs are real numbers, while in classification, the outputs correspond to distinct classes.

3.1.2.1 Support Vector Machine (SVM)

An SVM (Support Vector Machine) is a machine learning algorithm used for classification tasks. It assigns labels to objects based on examples and learns to recognize patterns in the data. SVMs belong to the family of supervised learning algorithms, where they learn from labeled examples to classify new, unseen data. They utilize mathematical principles and optimization techniques to construct a decision boundary that separates different classes or categories in the data. By maximizing a mathematical function with respect to the dataset, SVMs are widely recognized as a popular and effective algorithm for classification tasks in various fields such as fraud detection, image recognition, and gene expression classification. [23]

3.1.3 Unsupervised learning

Unsupervised learning involves using unlabeled data to make predictions about unknown outcomes, and the algorithm must autonomously discern the underlying patterns within the dataset [28]. This process does not require human intervention, as the algorithm's primary objective is to independently identify solutions. Unsupervised learning models are commonly employed for two main tasks: clustering and dimensionality reduction. Clustering involves grouping similar data points together based on their inherent similarities, while dimensionality reduction aims to reduce the complexity of the dataset by extracting relevant features and eliminating redundant information

3.1.4 Reinforcement learning

It's a learning paradigm concerned with learning to control a system so as to maximize a numerical performance measure that expresses a long-term objective [11], Safe Reinforcement Learning can be defined as the process of learning policies that maximize the expectation of the return in problems in which it is important to ensure reasonable system performance and/or respect safety constraints during the learning and/or deployment processes [2]

3.1.5 Image classification

Image classification is a computer vision task where images are categorized into predefined classes or categories. It involves training a model to learn visual features from labeled images and then using that model to classify new, unseen images. It has various applications, and deep learning, specifically Convolutional Neural Networks (CNNs), has played a significant role in advancing image classification accuracy. [18]

3.2 Deep learning

In the 1950s, AI started to develop. It was in 1956 in the Dartmouth Conference when Dr. McCarthy claimed that "machines may be designed to reason simulating every aspect of learning or any other element of intelligence". In this area there are various topics that are strongly related: AI, ML, Artificial NNs (ANNs) and DL. Figure 3.1 shows a taxonomy showing the links between them.

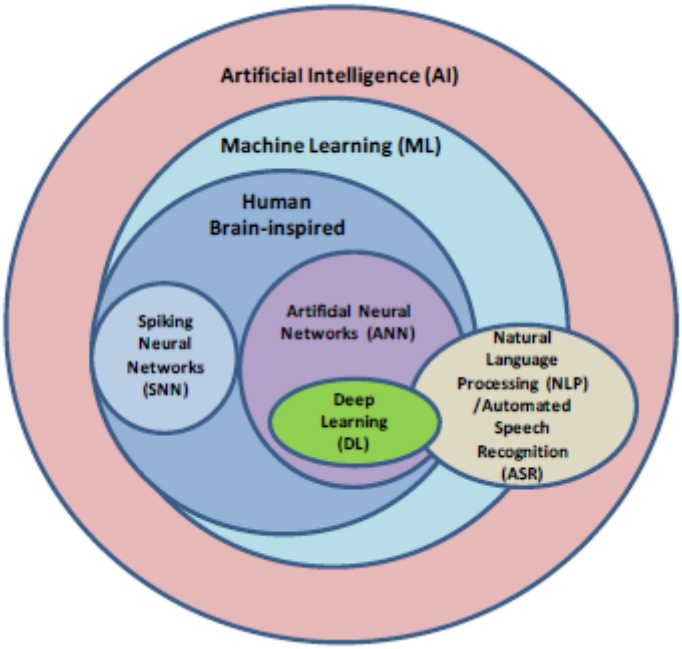


Figure 3.1: AI taxonomy: ML, SNN, NLP, ASR, ANN and DL.Figure reproduced from [26]

The purpose of AI is to automatize intellectual processes usually undertaken by people, whereas ML and DL are the specific methodologies that lead to this goal. In turn, ANNs are ML algorithms inspired by biological neural networks that model complex real-world issues. ANNs may be characterized as computing architectures created with basic processing elements, dubbed artificial neurons or nodes. Neurons are fullyconnected and imitate the way a human brain processes information, and solves issues. ANN comprised of three or more linked layers. They are able to execute massively parallel computations, enabling them to improve their conclusions as more data is fed to the model. DL is a sub-branch of ML that enables an algorithm to predict, categorize or make judgments based on data without explicitly being coded in a certain way. DL algorithms are more accurate than ML algorithms because of its multilayer nature. Hierarchically, they derive knowledge from data through many layers of non-linear processing units. DL is widely utilized by business and academics. It extends to fields of visual identification, audio processing, natural language understanding, pattern recognition, bioinformatics, mobile networking, and cybersecurity. DL gives encouraging results because to its tremendous efficiency in examining complicated data.Alongside the increase in learning methods, the main factors that contribute to DNN success are: the ever-increasing computing power, the breakthroughs in software engineering, and the vast amount of training data. [26]

Table 1: Major milestones that will be covered in this paper

Year	Contributer	Contribution
300 BC	Aristotle	introduced Associationism, started the history of human's attempt to understand brain.
1873	Alexander Bain	introduced Neural Groupings as the earliest models of neural network, inspired Hebbian Learning Rule.
1943	McCulloch & Pitts	introduced MCP Model, which is considered as the ancestor of Artificial Neural Model.
1949	Donald Hebb	considered as the father of neural networks, introduced Hebbian Learning Rule, which lays the foundation of modern neural network.
1958	Frank Rosenblatt	introduced the first perceptron, which highly resembles modern perceptron.
1974	Paul Werbos	introduced Backpropagation
1980	Teuvo Kohonen	introduced Self Organizing Map
	Kunihiko Fukushima	introduced Neocogitron, which inspired Convolutional Neural Network
1982	John Hopfield	introduced Hopfield Network
1985	Hilton & Sejnowski	introduced Boltzmann Machine
1986	Paul Smolensky	introduced Harmonium, which is later known as Restricted Boltzmann Machine
	Michael I. Jordan	defined and introduced Recurrent Neural Network
1990	Yann LeCun	introduced LeNet, showed the possibility of deep neural networks in practice
1997	Schuster & Paliwal	introduced Bidirectional Recurrent Neural Network
	Hochreiter & Schmidhuber	introduced LSTM, solved the problem of vanishing gradient in recurrent neural networks
2006	Geoffrey Hinton	introduced Deep Belief Networks, also introduced layer-wise pretraining technique, opened current deep learning era.
2009	Salakhutdinov & Hinton	introduced Deep Boltzmann Machines
2012	Geoffrey Hinton	introduced Dropout, an efficient way of training neural networks

Table 3.1: The history of DL [26]

3.3 Artificial Neural Network(ANN)

An Artificial Neural Network (ANN) is a type of machine learning model that is inspired by the structure and function of biological neural networks, such as the human brain. It consists of a large number of interconnected processing nodes or artificial neurons, which are organized into layers. Each neuron in an ANN receives input from one or more neurons in the previous layer, processes this input using a mathematical function, and passes the output to one or more neurons in the next layer. The neurons in the output layer produce the final result of the network.

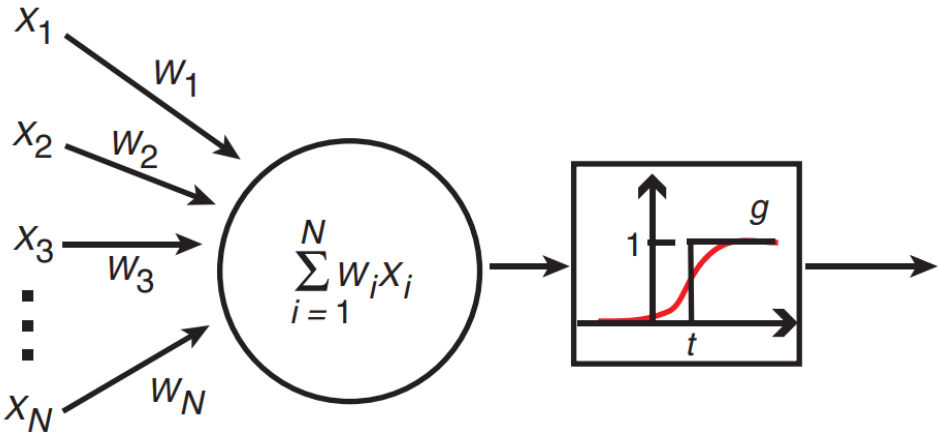


Figure 3.2: Neuron in Artificial Neural Network. [19]

During training, the weights and biases of the neurons are adjusted to minimize the error between the predicted output and the actual output. This process is typically done using a variant of gradient descent, where the network iteratively adjusts the weights and biases to improve its predictions. ANNs can be used for a wide range of tasks, such as classification, regression, and prediction, and they have been applied in many areas including computer vision, natural language processing, and robotics. Input: It is the set of features that are fed into the model for the learning process. [19]

3.3.1 Weight

Its main function is to give importance to those features that contribute more towards the learning. It does so by introducing scalar multiplication between the input value and the weight matrix. [19]

3.3.2 Transfer function

The job of the transfer function is to combine multiple inputs into one output value so that the activation function can be applied. It is done by a simple summation of all the inputs to the transfer function. [19]

3.3.3 Activation Function

It introduces non-linearity in the working of perceptrons to consider varying linearity with the inputs. [19]

3.3.4 Bias

The role of bias is to shift the value produced by the activation function. Its role is similar to the role of a constant in a linear function. [19]

3.3.5 Optimization algorithms

During the training process of a neural network, optimization algorithms are employed to modify the weights, aiming to reduce the error. Several well-known optimization algorithms utilized in deep learning are stochastic gradient descent (SGD), Adam, and Ada-grad. These algorithms vary in their weight update mechanisms and their capacity to handle diverse data types and architectures. [10]

3.3.6 Regularization and Dropout

Regularization and dropout are employed as methods to mitigate overfitting in neural networks. Regularization entails incorporating a penalty term into the loss function, which discourages the model from acquiring intricate patterns that might not generalize effectively to fresh data. Dropout, on the other hand, includes randomly omitting certain neurons in the network while training to avert excessive interdependence among them. [4]

3.4 Convolutional Neural Networks

Convolutional neural networks are deep learning algorithms that take input images and convolve it with filters or kernels to extract features. A CNN comprises three main types of neural layers, namely, convolutional layers, pooling layers and fully connected layers. Each type of layer plays a different role. Figure 3.3 shows a CNN architecture for an object detection in image task. Every layer of a CNN transforms the input volume to an output volume of neuron activation, eventually leading to the final fully connected layers, resulting in a mapping of the input data to a 1D feature vector. CNNs have been extremely successful in computer vision applications, such as face recognition, object detection, powering vision in robotics, and self-driving cars. [34]

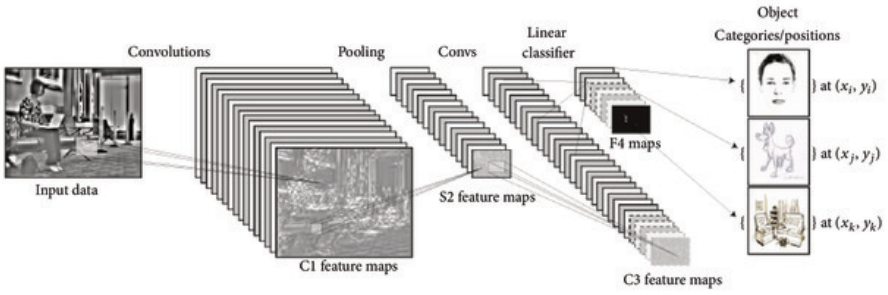


Figure 3.3: Example architecture of a CNN for a computer vision task (object detection). [34]

3.4.1 Convolutional Layers

Convolutional Layers. In the convolutional layers, a CNN utilizes various kernels to convolve the whole image as well as the intermediate feature maps, generating various feature maps. [34]

3.4.2 Pooling layers

Pooling layers reduce spatial dimensions for the next convolutional layer, without affecting depth. This operation is also known as subsampling or downsampling, causing a loss of information, but this can benefit the network by reducing computational overhead and preventing overfitting. Common strategies include average pooling and max pooling, with max pooling being shown to improve convergence, select superior features, and improve generalization. Other variations exist, such as stochastic pooling, spatial pyramid pooling, and def-pooling. [34]

3.4.3 Fully connected layers

Fully connected layers perform high-level reasoning in a CNN after convolutional and pooling layers. Neurons in this layer have full connections to the previous layer and their activation is computed with a matrix multiplication followed by a bias offset. They convert 2D feature maps into a 1D feature vector for classification or further processing. [34]

3.4.4 Stride

CNN offers more options, which gives a lot of opportunities to decrease the parameters more and more while reducing some of the side effects. Stride is one of these alternatives. We can manipulate the overlap by controlling the stride. Fig 3.5 shows a given 7x7 picture. We can only get a 5x5 result if we shift the filter one node at a time. It is worth noting that the output of the three left matrices in Fig 3.5 have an overlap (and three middle ones together and three right ones also). Yet, if we move and make every stride 2, the result will be 3x3. Simply put, not only will the overlap be reduced, but so will the size of the output. [3]

$$O = 1 + \frac{N - F}{S} \tag{3.1}$$

Equation (3.1),formalize this, given the image N×N dimension and the filter size of the F×F, the output size O as shown in 3.4. Where N is the input size, F is the filter size, and S is the stride size.

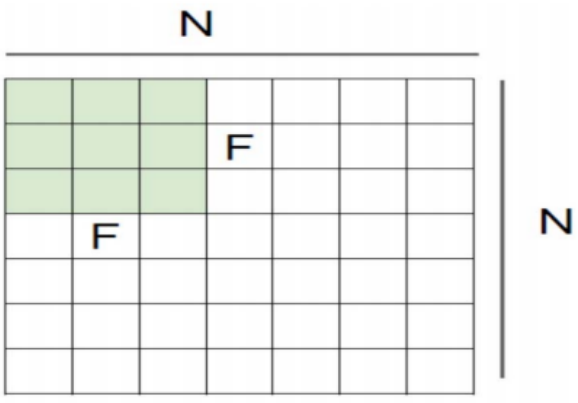


Figure 3.4: The effect of stride in the output. Figure reproduced from [3]

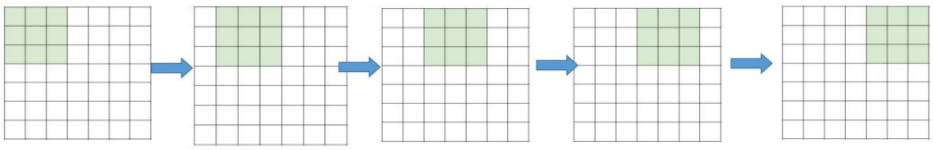


Figure 3.5: Stride 1, the filter window moves only one time for each connection. Figure reproduced from [3]

3.4.5 Padding

One drawback of the convolution process is the loss of information that might exist on the image's border. Because they are only captured when the filter slides, they are never seen. The use of zero-padding is a simple yet effective solution to the problem. Another advantage of zero padding is that it allows you to control the output size. In Fig. 3.5, for example, with N=7, F=3, and stride 1, the output is 5x5 (which reduces from a 7x7 input). [3]

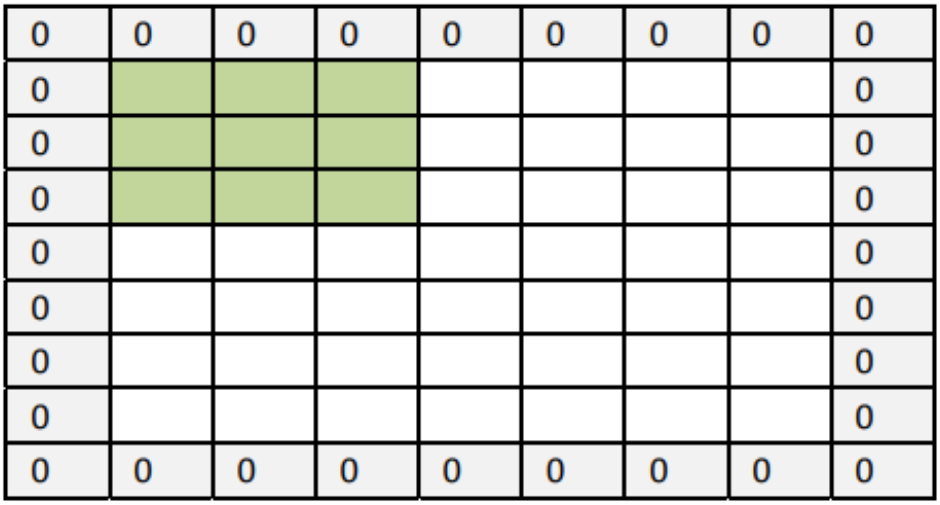


Figure 3.6: Zero-padding. Figure reproduced from [3]

However, by adding one zero-padding, the output will be 7×7, which is exactly the same as the original input (The actual N now becomes 9). The modified formula including zero-padding is formula (3.2).

$$O = 1 + \frac{N + 2P - F}{S} \tag{3.2}$$

Where P is the number of the layers of the zero-padding (e.g. P=1 in Figure(3.6), This padding idea helps us to prevent network output size from shrinking with depth. Therefore, it is possible to have any number of deep convolutional networks.

3.4.6 Pooling

The fundamental principle behind pooling is to reduce the complexity for successive layers by down-sampling. In the field of image processing, it is similar to decreasing the resolution. The number of filters is unaffected by pooling. One of the most prevalent forms of pooling procedures is max-pooling. It divides the image into sub-region rectangles and only returns the maximum value within each sub-region. The size 2x2 is one of the most commonly used in max-pooling. When pooling is performed in the top-left 2x2 blocks (pink area), it moves 2 and focuses on the top-right section, as seen in Fig. 12. This means that stride 2 is used in the pooling process. To avoid downsampling, Stride 1 can be used, which is uncommon. It should be noted that downsampling does not retain the information's location. As a result, it should only be used when the presence of information is critical (rather than spatial information). Also, pooling can be used to increase efficiency with non-equal filters and strides. For example, A 3x3 max-pooling with stride 2 maintains some overlaps between the areas. [3]

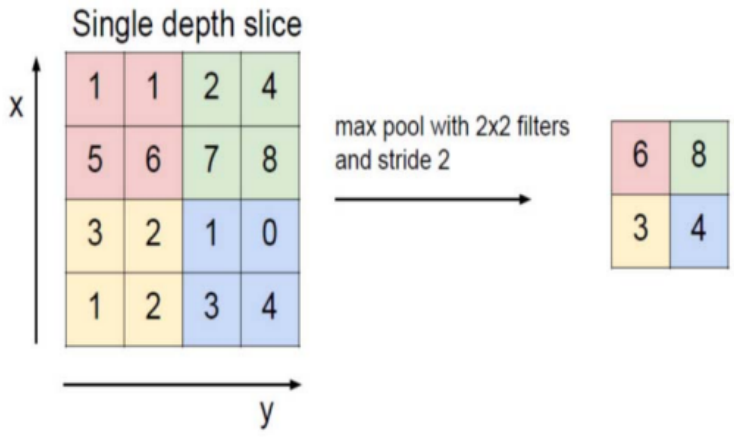


Figure 3.7: Max-pooling is demonstrated. The max-pooling with 2x2 filter and stride 2 lead to down-sampling of each 2x2 block is mapped to 1 block (pixel). Figure reproduced from [3]

3.5 Object detection

Object detection techniques facilitate the recognition, identification, and precise positioning of various visual instances of objects within images or videos. This approach goes beyond basic object classification, enabling a deeper understanding of the objects themselves. By employing this technique, we can accurately count the number of distinct objects present and annotate their specific locations. Its real-time application has significantly enhanced its speed over time. It effectively addresses the question of "Which object is located where, and how much of it is there?" The underlying concept relies on each object possessing unique characteristics, which serve as the foundation for this methodology. These characteristics aid in distinguishing one object from another, and they are leveraged in object detection processes, including face and fingerprint detection, among others. [32]

3.5.1 YOLOv8

YOLOv8 is a fast, accurate, and versatile object detection algorithm that can be used for a wide range of applications. It is based on the CSPDarknet53 architecture and has been pre-trained on the COCO dataset. YOLOv8 uses an anchor-free approach to object detection, which means that it does not need to pre-define a set of anchor boxes. This makes YOLOv8 more flexible and allows it to better handle objects of different sizes and shapes. YOLOv8 also predicts an objectness score for each bounding box, which indicates the probability that the box contains an object. This can be used to filter out false positives. YOLOv8 is trained on images at multiple scales, which helps it to better handle objects of different sizes. [29]

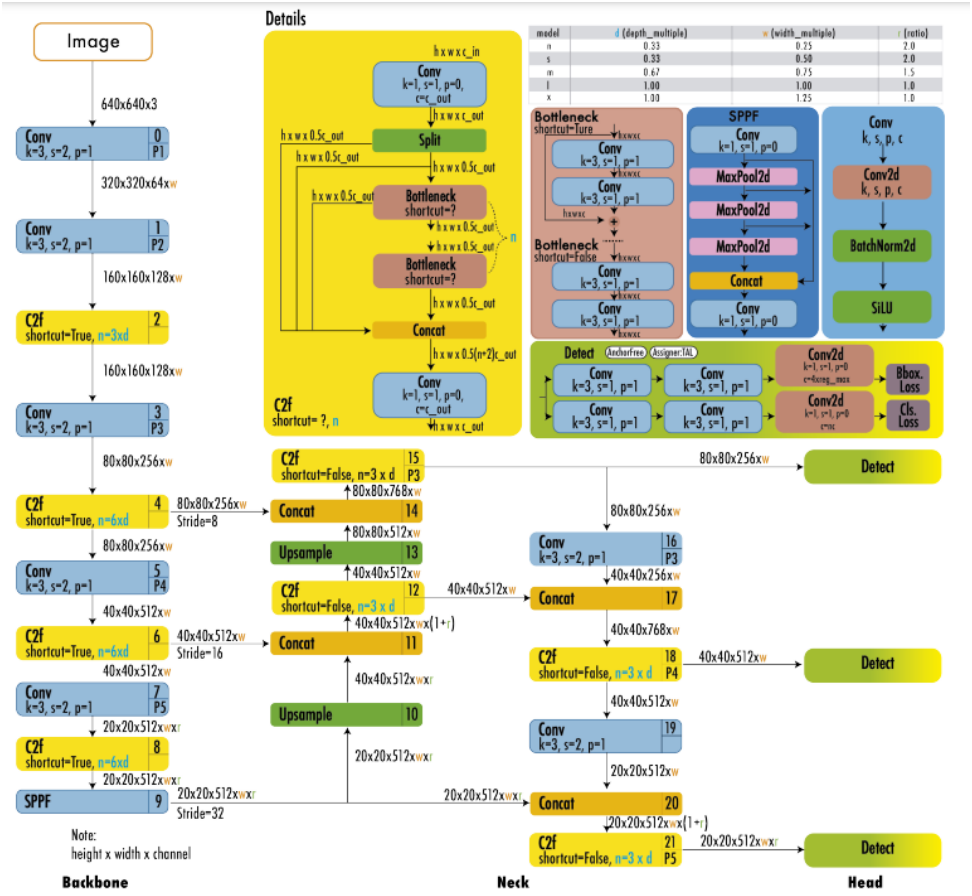


Figure 3.8: YOLOv8 Architecture. The architecture uses a modified CSPDarknet53 backbone. The C2f module replaces the CSPLayer used in YOLOv5. A spatial pyramid pooling fast (SPPF) layer accelerates computation by pooling features into a fixed-size map. Each convolution has batch normalization and SiLU activation. The head is decoupled to process objectness, classification, and regression tasks independently. [29]

3.5.2 Grounding DINO

Grounding DINO is an open-set object detector that integrates language inputs with DINO and grounded pre-training. It aims to detect arbitrary objects specified by human language. The model leverages Transformers for seamless fusion of image and language data, simplifies the model design, and achieves strong closed-set object detection performance. With its ability to generalize to unseen objects, Grounding DINO demonstrates promising applications in various domains. It combines the power of language and vision to enhance open-set object detection and supports referring expression comprehension. [21]

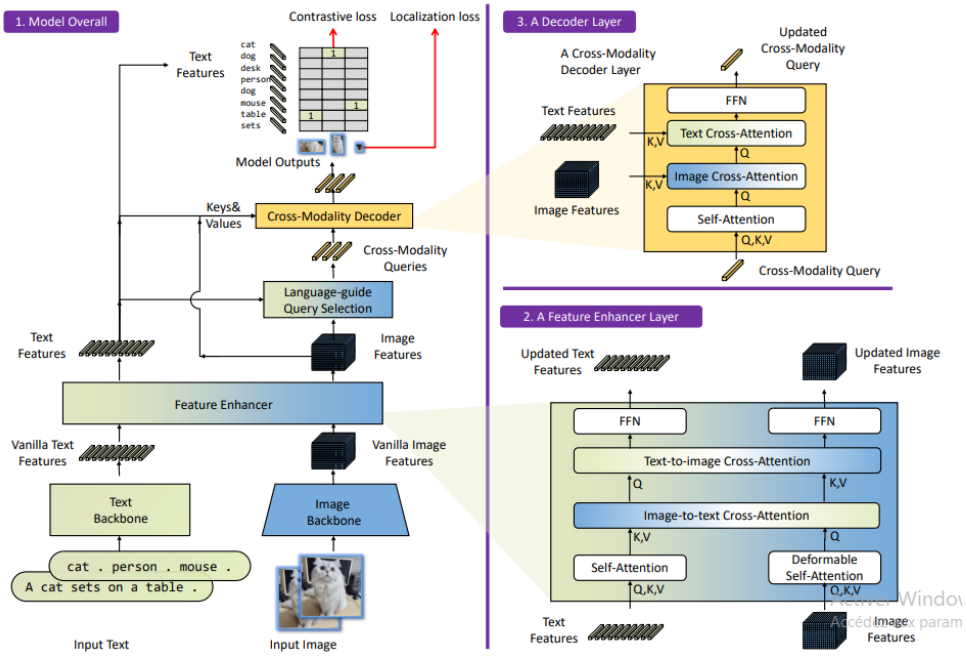


Figure 3.9: The framework of Grounding DINO.The overall framework, a feature enhancer layer, and a decoder layer in block 1, block 2, and block 3, respectively. Figure reproduced from [21]

3.6 Image segmentation

Image segmentation is a widely utilized technique in digital image processing, involving the partitioning of an image into multiple sections based on pixel properties. This method aims to separate the foreground from the background or group pixels together based on common color or shape characteristics. Image segmentation finds practical applications in various domains such as noise filtering in images, medical applications, object localization in satellite imagery, object detection and recognition tasks, automatic traffic control systems, video surveillance, and more. [15]

3.6.1 Segment Anything Model

A new model for image segmentation. By incorporating an efficient model into a data collection loop, a significantly large segmentation dataset has been created, surpassing any previous records. This dataset comprises over 1 billion masks derived from 11 million licensed images, ensuring privacy protection. The model has been specifically designed and trained to be promptable, allowing it to adapt seamlessly to new image distributions and tasks without requiring prior training. Through extensive evaluations across various tasks, we observe that its zero-shot performance is remarkably impressive, often comparable to, or even surpassing, the results achieved through fully supervised methods. [16]

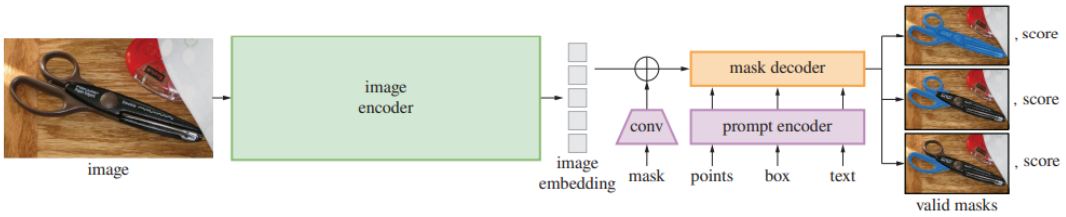


Figure 3.10: Segment Anything Model. Figure reproduced from [16]

3.7 Recurrent Neural Network

During the 1990s, recurrent neural networks (RNNs) gained significant attention in research and development. They were specifically designed to learn patterns that occur sequentially or change over time. RNNs are a type of neural network that incorporates feedback connections, forming a closed loop structure. Notable examples of RNNs include BAM, Hopfield, Boltzmann machines, and recurrent backpropagation nets. These techniques have been applied to a diverse range of problems. In the late 1980s, researchers like Rumelhart, Hinton, and Williams introduced simple partially recurrent neural networks for learning character sequences. Additionally, RNNs have been utilized in various applications involving dynamical systems with time-sequenced events. [22]

3.7.1 Recurrent Neural Net architectures

The network architectures vary from completely interconnected, as shown in Figure 3.11, to partially connected networks, as shown in Figure 3.12. This includes multilayer feedforward networks that have separate input and output layers. In fully connected networks, there are no distinct input layers, and each node receives input from all other nodes. Additionally, nodes in these networks can also receive feedback from themselves.

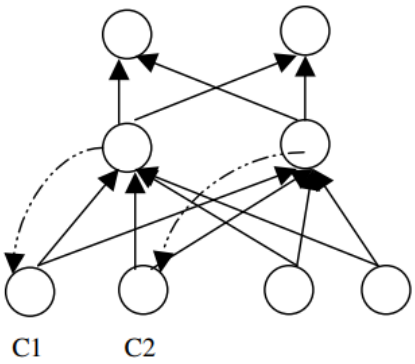


Figure 3.11: An example of a simple recurrent network. Figure reproduced from [3]

In the domain of learning strings of characters, simple partially recurrent neural networks have been employed. These networks consist of nodes that form a feedforward structure, while other nodes provide sequential context and receive feedback from one another. Context units, such as C1 and C2, undergo weight processing similar to input units, often employing backpropagation. The context units receive feedback from the second layer units, introducing a

time-delayed aspect. During training, input-output pairs are used to teach the network to predict the next character in a character string and validate the string. Two primary approaches for incorporating feedback into feedforward multilayer neural networks are described. Elman's approach focuses on feedback from the hidden layer to the context portion of the input layer, emphasizing input value sequence. In contrast, Jordan's recurrent neural networks utilize feedback from the output layer to the context nodes of the input layer, prioritizing the output value sequence. This book explores various modifications and enhancements to these core concepts, providing insights into more efficient and effective designs for recurrent neural networks, along with intriguing application examples. [22]

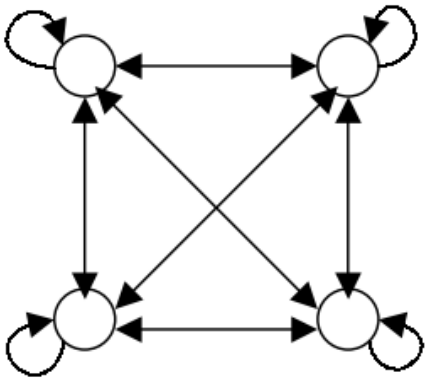


Figure 3.12: An example of a fully connected recurrent neural network. Figure reproduced from [3]

3.7.1.1 Learning in Recurrent Neural Nets

Learning is a crucial aspect of neural networks, making them highly appealing for challenging artificial intelligence applications. Extensive research has focused on learning algorithms, including Hebbian learning and gradient descent, which serve as the foundation for neural network techniques. Backpropagation, a popular form of gradient descent, was introduced to address the challenges of learning in neural networks. However, practical applications of backpropagation can encounter issues like getting trapped in local minima. Recurrent neural networks, with their time-delayed input processing, require more sophisticated algorithms for effective learning. Researchers have explored approaches like backpropagation through time, which approximates the time evolution of recurrent networks, and the use of master networks to program attractors in slave networks. Several techniques have been proposed to extend backpropagation learning to recurrent networks, as summarized in Pearlmutter's work. [22]

3.8 The transformer network

The Transformer network relies exclusively on attention mechanisms, eliminating the need for recurrent and convolutional layers. It offers higher quality results, enhanced parallelizability, and shorter training times compared to traditional models.

3.8.1 Transformer model architecture

Many state-of-the-art neural sequence transduction models adopt an encoder-decoder structure. In this structure, the encoder takes an input sequence of symbol representations (x1, ..., xn) and transforms it into a sequence of continuous representations (z1, ..., zn). Subsequently, the decoder utilizes the continuous representations (z) to generate an output sequence (y1, ..., ym) one symbol at a time. The model is auto-regressive, meaning it leverages previously generated symbols as additional input when generating the next symbol. The Transformer architecture follows this general framework, employing stacked self-attention and point-wise, fully connected layers for both the encoder and decoder. You can observe this structure in the left and right halves of Figure 3.13 respectively. [33]

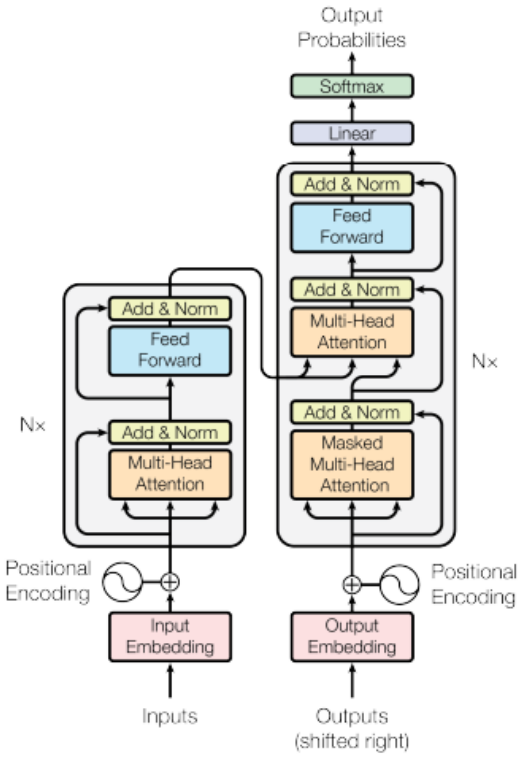


Figure 3.13: Transformer model architecture. Figure reproduced from [33]

3.8.2 BERT

BERT, an acronym for Bidirectional Encoder Representations from Transformers, is a language representation model that differs from previous models by focusing on deep bidirectional representations. Unlike previous models that relied on left-to-right or right-to-left context, BERT is designed to pretrain on unlabeled text, considering both left and right context in all layers. By doing so, BERT achieves superior performance on various tasks by fine-tuning a single additional output layer, without requiring significant modifications to the underlying architecture.

BERT's framework comprises two main steps: pre-training and fine-tuning. During pre-training, the model learns from unlabeled data by performing different pre-training tasks. In the fine-tuning stage, the pre-trained BERT model is initialized with its learned parameters and further trained using labeled data from specific downstream tasks. Each downstream task has its own fine-tuned model, even though they share the same pre-trained parameters. The example of question-answering illustrated in Figure 3.14 serves as an ongoing example in this section.[NEW] . [9]

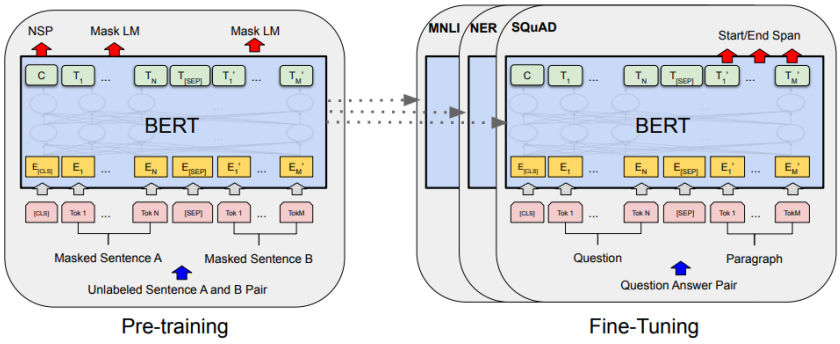


Figure 3.14: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token (e.g. separating questions/answers). [9]

3.8.2.1 Variants of BERT

3.8.2.1.1 BERT Base

The model consists of 12 transformer blocks, each with 12 attention heads, and a total of 110 million parameters. [9]

3.8.2.1.2 BERT large

The model consists of 24 transformer blocks, 16 attention heads, and a total of 340 million parameters. [9]

In figure 3.15 you can find the summary of the BERT base that we used in this study.

```
==========================================================================
Layer (type:depth-idx)                                  Param #
==========================================================================
BertModel                                               --
├─BertEmbeddings: 1-1                                   --
│    └─Embedding: 2-1                                   23,440,896
│    └─Embedding: 2-2                                   393,216
│    └─Embedding: 2-3                                   1,536
│    └─LayerNorm: 2-4                                   1,536
│    └─Dropout: 2-5                                     --
├─BertEncoder: 1-2                                      --
│    └─ModuleList: 2-6                                  --
│    │    └─BertLayer: 3-1                              7,087,872
│    │    └─BertLayer: 3-2                              7,087,872
│    │    └─BertLayer: 3-3                              7,087,872
│    │    └─BertLayer: 3-4                              7,087,872
│    │    └─BertLayer: 3-5                              7,087,872
│    │    └─BertLayer: 3-6                              7,087,872
│    │    └─BertLayer: 3-7                              7,087,872
│    │    └─BertLayer: 3-8                              7,087,872
│    │    └─BertLayer: 3-9                              7,087,872
│    │    └─BertLayer: 3-10                             7,087,872
│    │    └─BertLayer: 3-11                             7,087,872
│    │    └─BertLayer: 3-12                             7,087,872
├─BertPooler: 1-3                                       --
│    └─Linear: 2-7                                      590,592
│    └─Tanh: 2-8                                        --
==========================================================================
Total params: 109,482,240
Trainable params: 109,482,240
Non-trainable params: 0
```

Figure 3.15: Summary of BERT model

3.8.3 GPT-2

GPT-2 is a deep learning model that uses unsupervised learning to generate high-quality text in a variety of natural language processing tasks. It is based on the transformer architecture and is pre-trained on a large corpus of text using an unsupervised learning approach. The model is capable of generating text that is coherent, relevant, and semantically meaningful, and has achieved state-of-the-art performance on several benchmark NLP tasks, such as language modeling, text generation, and question answering. GPT-2 is one of the largest and most powerful language models to date, with 1.5 billion parameters, and has been made available for use by researchers and developers in the field of NLP [25], below you can find the architecture of the model. In Figure 3.16 you can find the GPT-2 architecture.

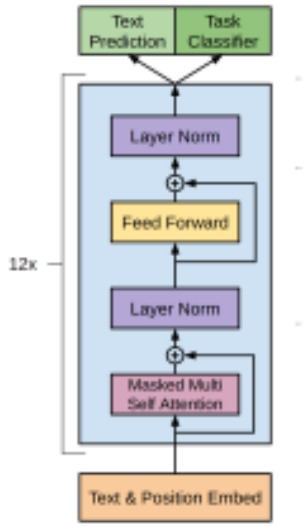


Figure 3.16: GPT-2 Architecture [25]

3.8.3.1 Variants of GPT-2

3.8.3.1.1 GPT-2 Small

The variant of GPT-2 we're referring to here is the smallest one, boasting a relatively modest parameter count of 117 million.

3.8.3.1.2 GPT-2 Medium

This particular version of GPT-2 can be classified as a medium-sized variant, featuring a substantial parameter count of 345 million.

3.8.3.1.3 GPT-2 Large

This is a large variant of GPT-2, with 762 million parameters.

3.8.3.1.4 GPT-2 Extra large

This is considered as the largest variant of GPT-2, as it encompasses an extensive parameter count of 1.5B .

Figure 3.17 provides a summary of the GPT-2 small model employed in our study.

```
==========================================================================
Layer (type:depth-idx)                                  Param #
==========================================================================
GPT2ForSequenceClassification                           --
├─GPT2Model: 1-1                                        --
│    └─Embedding: 2-1                                   38,597,376
│    └─Embedding: 2-2                                   786,432
│    └─Dropout: 2-3                                     --
│    └─ModuleList: 2-4                                  --
│    │    └─GPT2Block: 3-1                              7,087,872
│    │    └─GPT2Block: 3-2                              7,087,872
│    │    └─GPT2Block: 3-3                              7,087,872
│    │    └─GPT2Block: 3-4                              7,087,872
│    │    └─GPT2Block: 3-5                              7,087,872
│    │    └─GPT2Block: 3-6                              7,087,872
│    │    └─GPT2Block: 3-7                              7,087,872
│    │    └─GPT2Block: 3-8                              7,087,872
│    │    └─GPT2Block: 3-9                              7,087,872
│    │    └─GPT2Block: 3-10                             7,087,872
│    │    └─GPT2Block: 3-11                             7,087,872
│    │    └─GPT2Block: 3-12                             7,087,872
│    └─LayerNorm: 2-5                                   1,536
├─Linear: 1-2                                           1,536
==========================================================================
```

Figure 3.17: Summary of GPT-2 small Architecture

3.9 Generative Adversarial Network(GAN)

A Generative Adversarial Network (GAN) is a machine learning model consisting of a generator and a discriminator. The generator creates synthetic data samples, while the discriminator distinguishes between real and generated samples. Through adversarial training, GANs learn to generate realistic data that resembles the training set, enabling tasks such as image synthesis and text generation. GANs have gained prominence for their ability to capture and replicate complex data distributions. [13]

3.10 Conclusion

In conclusion, this chapter delved into the fascinating world of artificial intelligence (AI). We explored various concepts and techniques that highlight the immense potential of AI in transforming industries and shaping our future. From machine learning algorithms to deep neural networks, AI has shown remarkable capabilities in areas such as computer vision, natural language processing, and decision-making. As AI continues to advance, it promises to revolutionize numerous domains, from healthcare and transportation to finance and entertainment. The journey of AI is an ongoing one, requiring continuous research, innovation, and collaboration to unlock its full potential and ensure its positive impact on society.

Chapter 4
Contribution

Chapter 4

Contribution

4.1 AI-based services for fraud detection and market price prediction

The image depicts a prototype of our final project, after preparing the models shown in the figure, they are fed into the chatbot to reply to customers' questions related to the market, and the chatbot is also integrated with GPT3.5 to give the user better textual format, the user can also discuss with real world experts for any information regarding the market. The project is divided into two main components: fraud detection and market prediction. This thesis focuses specifically on the fraud detection aspect, while the other part is dedicated to the collaboration with our two colleagues in the MICR specialty.

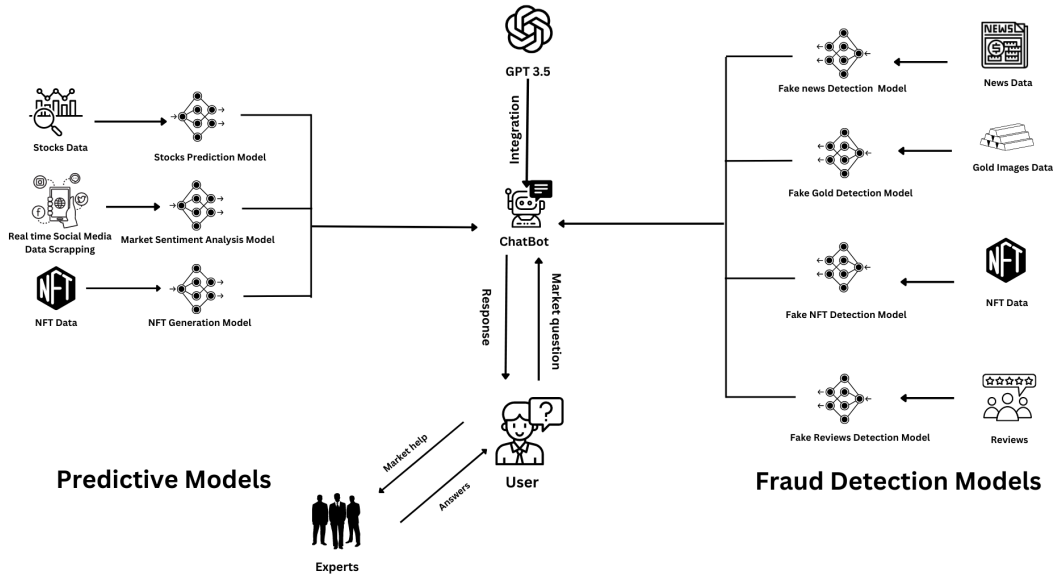


Figure 4.1: System architecture

4.2 Fake reviews

During our investigation into fake review detection, we conducted experiments using both BERT and GPT-2 models on the same dataset. Below, we provide an overview of the architecture for each of these models.

4.2.1 NadraBERT

Having explored the architecture showcased in Figure 3.15, it is now time to delve into the modified architecture we intend to implement within our model. This revised architecture has been tailored to suit our specific requirements and objectives, taking into account the unique considerations and enhancements we aim to introduce. By customizing the architecture, we can optimize our model's performance and ensure it aligns seamlessly with our desired outcomes. Through careful analysis and thoughtful adjustments, we are confident that the modified architecture will offer improved functionality and contribute to the overall success of our implementation.

```
==========================================================================
Layer (type:depth-idx)                                  Param #
==========================================================================
ReviewsClassifier                                       --
├─BertModel: 1-1                                        --
│    └─BertEmbeddings: 2-1                              --
│    │    └─Embedding: 3-1                              23,440,896
│    │    └─Embedding: 3-2                              393,216
│    │    └─Embedding: 3-3                              1,536
│    │    └─LayerNorm: 3-4                              1,536
│    │    └─Dropout: 3-5                                --
│    └─BertEncoder: 2-2                                 --
│    │    └─ModuleList: 3-6                             85,054,464
│    └─BertPooler: 2-3                                  --
│    │    └─Linear: 3-7                                 590,592
│    │    └─Tanh: 3-8                                   --
├─Dropout: 1-2                                          --
├─Linear: 1-3                                           196,864
├─Linear: 1-4                                           32,896
├─Linear: 1-5                                           258
==========================================================================
Total params: 109,712,258
Trainable params: 109,712,258
Non-trainable params: 0
==========================================================================
```

Figure 4.2: NadraBERT architecture

In order to optimize the performance of our model, we have made specific modifications. Firstly, a dropout layer with a dropout rate of 0.3 has been introduced to address overfitting concerns. This layer helps prevent the model from relying too heavily on specific features during training, leading to better generalization. Additionally, two dense layers have been incorporated into the architecture. The first dense layer has 512 units, allowing for the extraction of detailed features from the input data. The second dense layer further refines these features, condensing them into a representation with 128 dimensions. Finally, a dedicated classification layer with n_classes units has been added to enable accurate predictions and distinguish between the specific target classes. These architectural adjustments aim to enhance precision, reliability, and the overall performance of our model.

4.2.2 NadraGPT-2

After examining the architecture presented in Figure 3.17, it is now necessary to investigate the modified architecture that we plan to incorporate into our model.The modified architecture we have implemented has been carefully tailored to meet our specific requirements and objectives. We have taken into account the unique considerations and enhancements we aim to incorporate, ensuring that the architecture is optimized for performance and closely aligned with our desired outcomes. By conducting thorough analysis and making thoughtful adjustments, we have enhanced the functionality of the architecture.

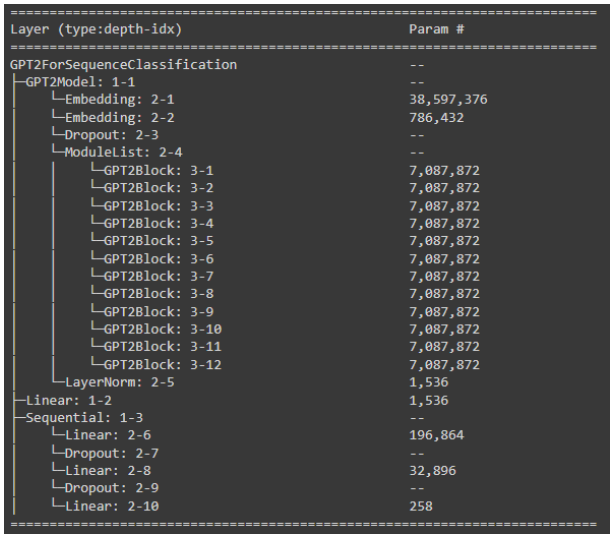


Figure 4.3: NadraGPT-2 Architecture.

By incorporating the provided change, we have modified the model's classifier. The new configuration includes three linear layers with specific dimensions. Initially, a linear layer with an input size and an output size of 256 has been added. This is followed by a dropout layer with a dropout rate of 0.4, which helps combat overfitting. Subsequently, another linear layer with an input size of 256 and an output size of 128 has been introduced. Another dropout layer with the same dropout rate of 0.4 is then applied. Finally, a linear layer with an input size of 128 and an output size has been included to enable accurate classification. With these changes, we aim to improve the model's performance, enhance feature extraction, and achieve reliable classification of the target labels.

4.3 Fake news

In our fake detection experiments, we compared the performance of NadraGPT-2 model that was fine-tuned and Nadra Embedding Convolutional model on the same dataset.

4.3.1 Nadra Embedding Convolutional Model

We experimented with various architectures in our CNN model and selected the one that demonstrated the highest performance on the dataset. The chosen architecture is presented below.

The selected layers are as follows:

- **Embedding layer:** To transform the tokens in our data into embeddings of a specific size of 100, we utilized the GloVe model as a basis.
- **The Conv1D and MaxPooling1D Layers:** We employ three convolutional layers of a specific dimension, followed by three max pooling layers.
- **Flatten layer:** We include a flatten layer to prepare the output of the max pooling layer for the fully connected layer.
- **The Dropout layer:** we applied a single Dropout layer for regularization purposes.
- **The fully connected layer:** We employed three fully connected layers with 200 hidden neurons and utilized the sigmoid function to predict the output.

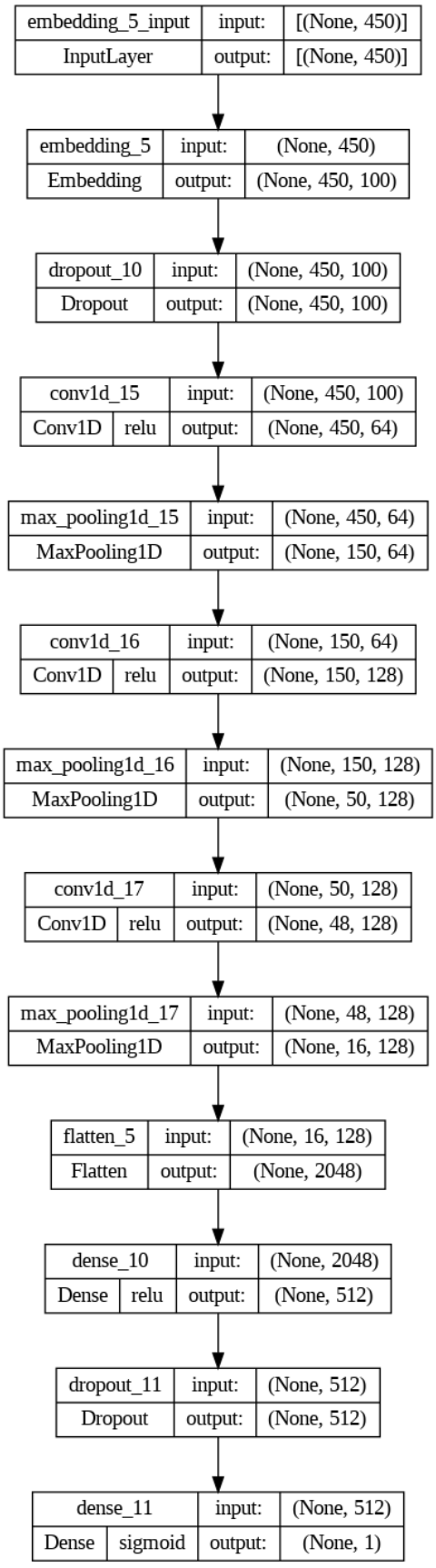


Figure 4.4: Nadra Embedding Convolutional Model

4.3.2 NadraGPT-2

Because it demonstrated slightly better performance among the evaluated architectures, we employed the identical structures as those utilized in NadraGPT-2 for fake news detection, as depicted in Figure 4.3.

4.4 Fake gold

Our gold detection methodology encompasses a sequential set of procedures. Firstly, we employ the Grounding Dino algorithm to perform object detection, thereby facilitating the identification of gold objects within the given images. Subsequently, we utilize the Segment Anything Model (SAM) for image segmentation, enabling the precise isolation of gold objects from the surrounding background. Lastly, an image classification technique employing Fully Connected Neural Networks (FCNNs) is applied to categorize the segmented regions as either gold or non-gold. By orchestrating these interconnected steps, our approach ensures heightened accuracy and robustness in detecting gold objects embedded within images.

4.4.1 Object detection with YOLOv8

After many experiments, varying with the parameters of YOLOv8 and experimenting different implementations. The best results were the following:

Figure 4.5: The original gold picture.

Figure 4.6: The picture after applying YOLOv8

4.4.2 Object detection with Grounding DINO

Using Grounding DINO for object detection to detect our gold bars from the images. For that purpose, we gave it the following prompt: [Yellow gold cast bar]

Here's an example when using it on our gold images:

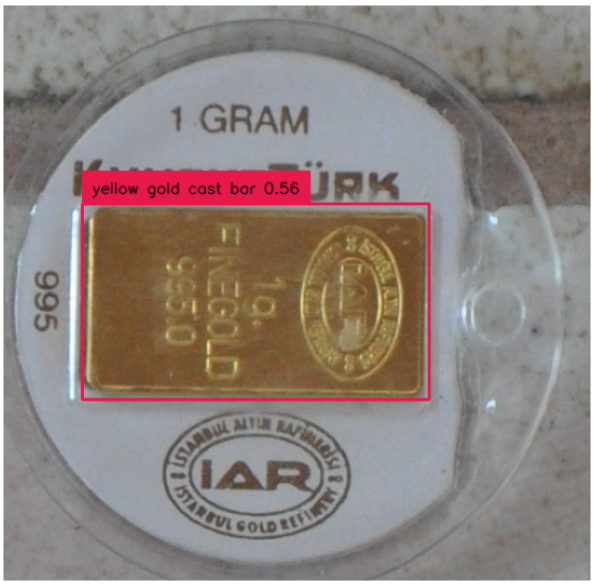


Figure 4.7: The picture after applying Grounding DINO.

4.4.3 Image segmentation with SAM

The second step is using Segment Anything Model (SAM) to segment the gold bar from the image 4.7 that contains object detected by the Grounding DINO algorithm .

Here's the result when using it on the figure 4.7 :

Figure 4.8: The picture after applying Segment Anything Model (SAM).

After that, we crop the image using the bounding box's coordinates provided by the Grounding DINO algorithm, here's an example of the result:

Figure 4.9: The picture after cropping using the bounding box's coordinates.

4.4.4 Using Fully Connected Neural Networks (FCNNs)

We extensively investigated various architectures in our CNN model and meticulously identified the one that demonstrated the best performance on our dataset. Through experimentation, we discovered that the architecture comprising solely of fully connected layers achieved the highest performance.

The architecture that showcased the best results is detailed below.

- **Flatten layer:** We include a flatten layer to prepare the input for the fully connected layers.
- **The activations:** We used the Relu activation function for the hidden layers and the sigmoid activation function for the output layer.
- **The fully connected layers:** We employed two fully connected layers with 128 hidden units and utilized the sigmoid function to predict the output.

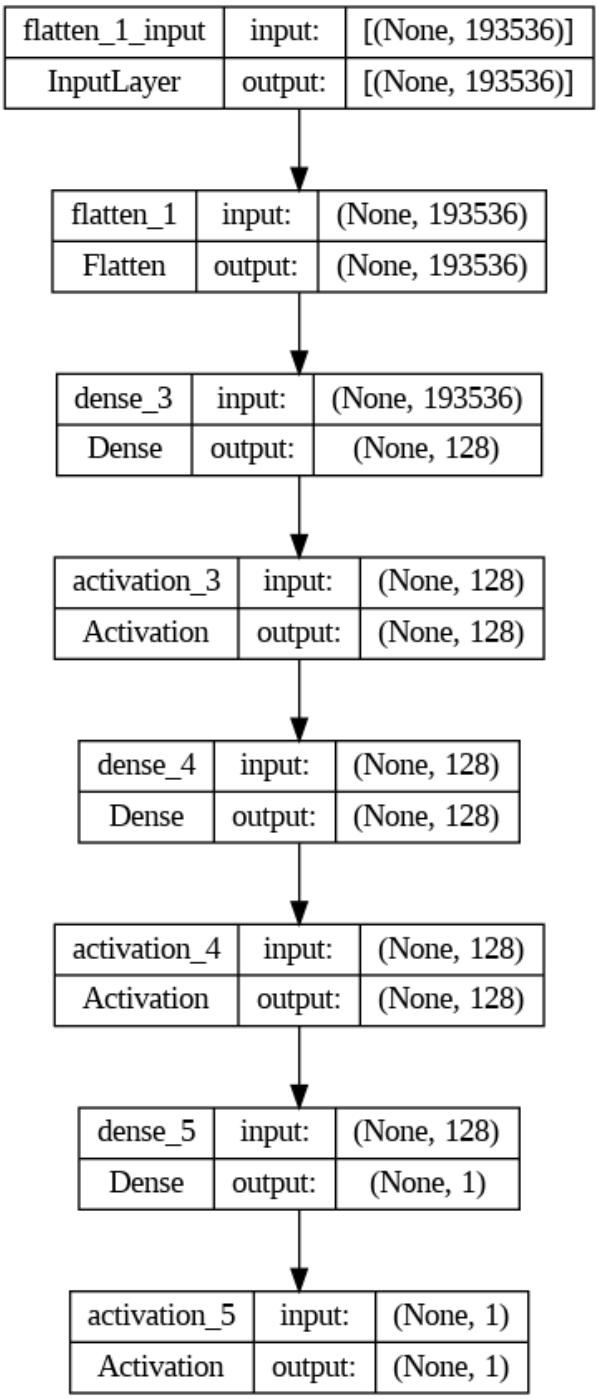


Figure 4.10: Gold Model Architecture

4.5 Fake NFT

4.5.1 Fully Connected Neural Networks (FCNNs)

We conducted a thorough exploration of different architectures within our CNN model and carefully identified the one that exhibited the most superior performance on our dataset. After conducting experiments, we discovered that utilizing zero convolutional layers yielded the best performance for our specific task. This finding suggests that the inclusion of convolutional layers did not contribute significantly to the accuracy or effectiveness of the model. The architecture that provided the highest performance solely consisted of fully connected layers.

The architecture that showcased the best results is detailed below.

- **Flatten layer:** We include a flatten layer to prepare the input for the fully connected layers.
- **The activations:** We used the Relu activation function for the hidden layers and the sigmoid activation function for the output layer.
- **The fully connected layers:** We employed two fully connected layers with 32 hidden units and utilized the sigmoid function to predict the output.

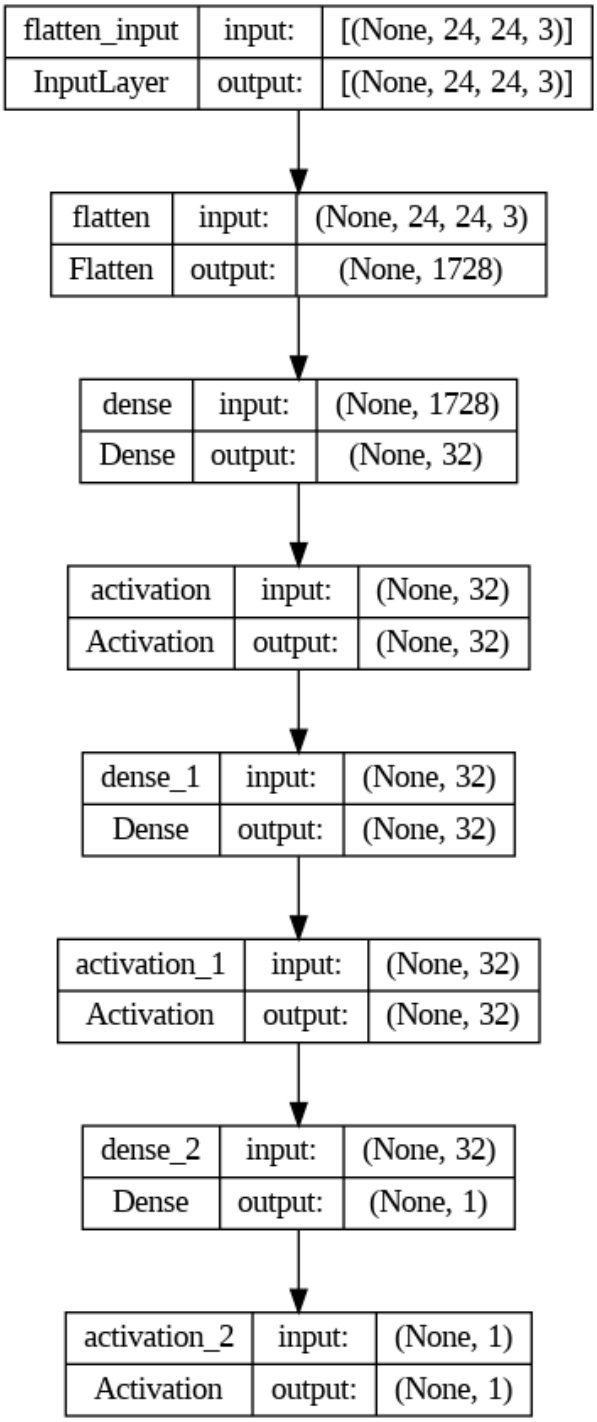


Figure 4.11: Model architecture for fake NFT detection

4.6 Chatbot

A chatbot is an AI system that exemplifies one of the most common and widely used forms of intelligent Human-Computer Interaction. It is a computer program that exhibits smart behavior when engaged in conversation, whether through text or voice, and utilizes Natural Language Processing (NLP) to comprehend one or more human languages. In simpler terms, a chatbot is defined as a "computer program created to simulate conversation with human users, particularly online". Chatbots are also referred to as smart bots, interactive agents, digital assistants, or artificial conversational entities. [1]

4.6.1 Nadra bot

In this research, we propose incorporating GPT-3.5, an advanced language model, into a chatbot to enhance the interaction with our fraud detection models. By integrating our models with this chatbot, we create a seamless communication channel between users and our fraud detection algorithms. Through the chatbot's sophisticated natural language processing capabilities, users can input queries and receive predictions regarding fake news. This interactive approach provides a user-friendly and intuitive experience, empowering users to gain insights and make well-informed decisions based on our models' predictions. The integration of the chatbot serves as a connection, enabling users to directly access our models and receive reliable predictions.

4.7 Conclusion

In summary, this chapter provided an overview of Nadra Embedding Convolutional model, NadraGPT-2, NadraBERT fine-tuning, and fully connected neural networks. CNNs are effective in text classification, GPT-2 revolutionizes natural language processing, BERT fine-tuning adapts models to specific tasks, and fully connected neural networks serve as foundational elements in deep learning. These architectures play vital roles in advancing AI and enabling intelligent systems to process and understand both visual and textual information.

Chapter 5
Results, Discussions and Experimentations

Chapter 5

RESULT, DISCUSSION AND EXPERIMENTATION

5.1 Introduction

Unlike the previous chapter, which primarily focused on providing a theoretical explanation of our proposed system to establish its significance, this chapter presents the achieved results to assess the efficacy of our solution in fake news, fake reviews, fake NFTs as well as counterfeit gold, we delve into a comprehensive analysis of our models' performance in the frauds mentioned above. Furthermore, we address the constraints of our approach, draw conclusions based on our experimental outcomes, and explore potential applications of our system.

5.2 Datasets descriptions

5.2.1 Gold dataset

The dataset contains 80 images, from which 40 represent real gold images & 40 represent copper images (consult the dataset (here)). Below you can find a subset from the dataset.

5.2.1.1 Preprocessing

As mentionned in the contribution chapter, we used Grounding DINO algorithm to detect the gold cast bar from the picture, then passed it along with its bounding box to the Segment Anything Model (SAM) to be segmented, then we cropped the gold cast bar from the image. After these steps, the images are passed to the Fully Connected Neural Network (FCNNs) layers.

5.2.2 NFT dataset

Initially, the dataset contained 10,000 images (here). A subset of 2,342 real Cryptopunks NFTs was selected from the original dataset for further analysis and comparison. In addition, 2,342 fake Cryptopunks NFTs were generated using a Generative Adversarial Network (GAN). This dataset provides a diverse collection of both real and generated Cryptopunks NFTs, enabling comprehensive exploration and analysis.

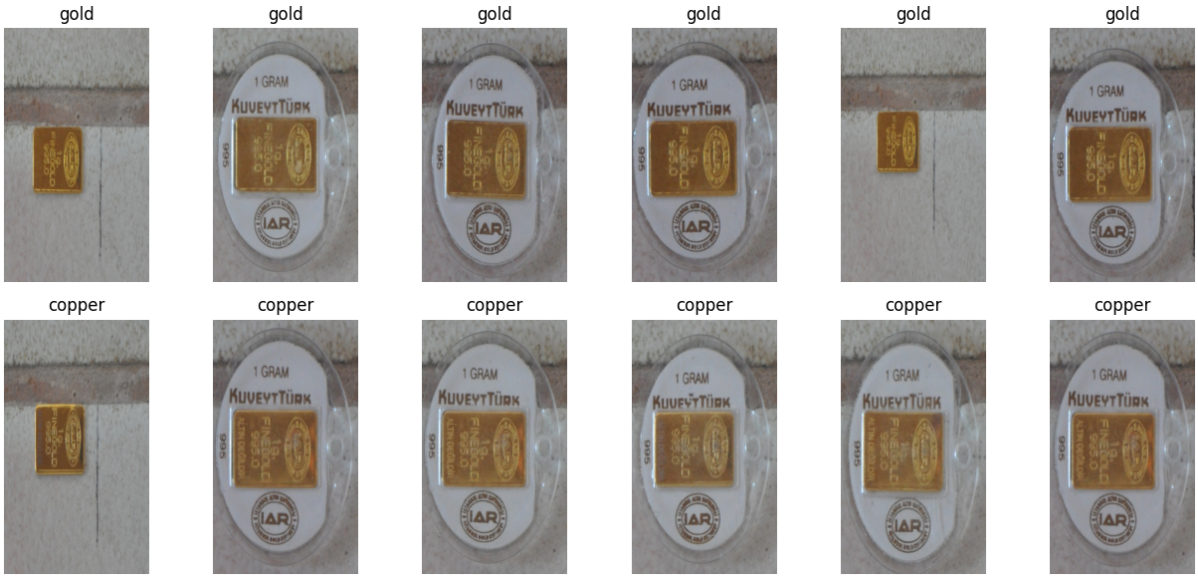


Figure 5.1: Subset from gold/copper dataset

5.2.2.1 Preprocessing

As a part of the preprocessing step, a GAN was employed to generate 2,342 fake Cryptopunks NFTs. The GAN utilized a generative model to create synthetic images resembling Cryptopunks NFTs. Subsequently, to enhance the diversity and variability of the dataset, various data augmentation techniques were applied to the training split. These techniques involved transformations such as rotation, scaling, cropping, and noise addition to the existing images. These preprocessing steps were undertaken to optimize the dataset for subsequent tasks, such as training deep learning models and analyzing the characteristics of Cryptopunks NFTs.

5.2.2.2 Dataset Samples

Here are some samples of Real NFTs from Cryptopunks dataset and Fake NFTs that we generated with Generative Adversarial Network (GAN).

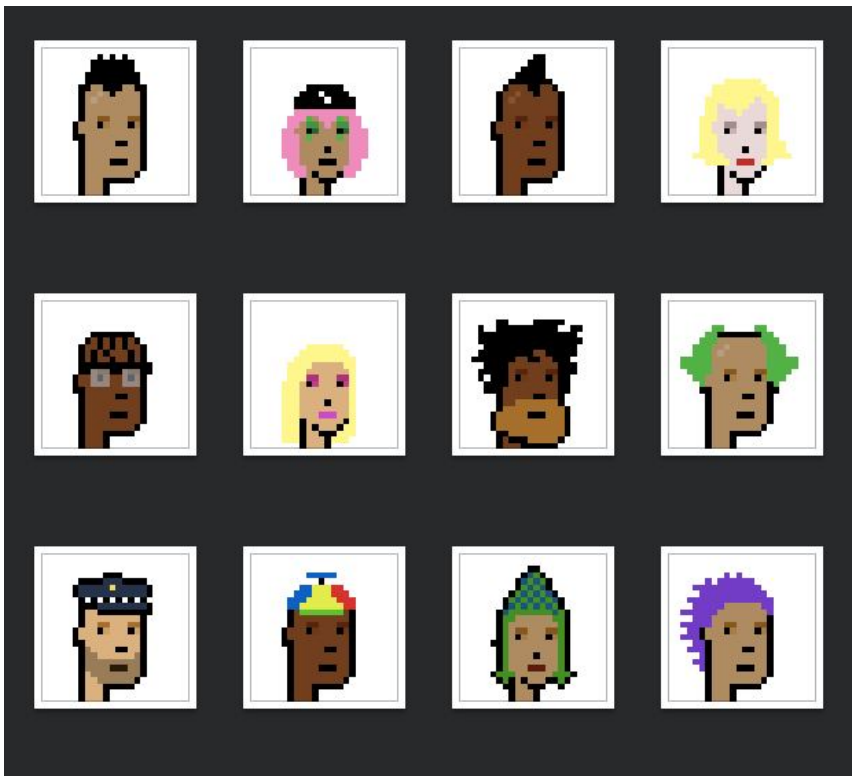


Figure 5.2: Real NFTs

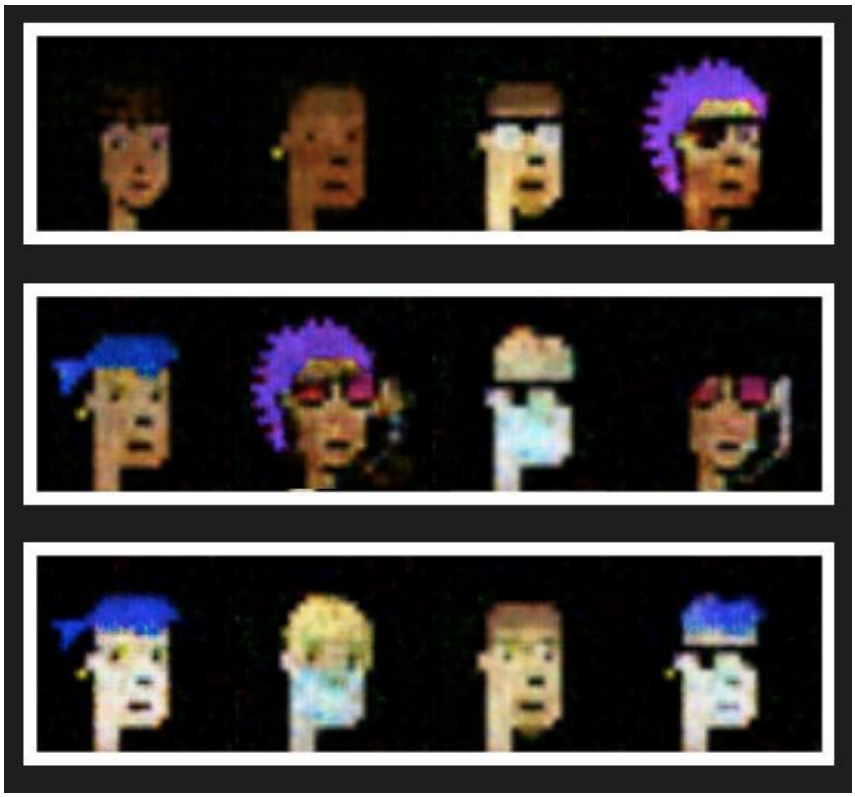


Figure 5.3: Fake NFTs

5.2.3 Fake news dataset

The dataset contains a total of 72,000 instances, divided into two categories: 35,028 instances of fake news and 37,106 instances of real news. Each instance in the dataset is accompanied by various features, including the title, text, and label. Dataset is available here. Below you can find a brief overview of the dataset.

	title	text	label
0	LAW ENFORCEMENT ON HIGH ALERT Following Threat...	No comment is expected from Barack Obama Membe...	1
1	NaN	Did they post their votes for Hillary already?	1
2	UNBELIEVABLE! OBAMA'S ATTORNEY GENERAL SAYS MO...	Now, most of the demonstrators gathered last ...	1
3	Bobby Jindal, raised Hindu, uses story of Chri...	A dozen politically active pastors came here f...	0
4	SATAN 2: Russia unveils an image of its terrif...	The RS-28 Sarmat missile, dubbed Satan 2, will...	1

Figure 5.4: Subset of fake news dataset

5.2.3.1 Preprocessing

The initial steps of data pre-processing are crucial to ensure that the data is prepared in a format suitable for analysis. We eliminated all columns except for text and label since they were the only relevant features. Additionally, we removed any missing values and conducted other essential cleaning procedures to guarantee data quality.

The pre-processing stage holds great importance as it guarantees the accuracy, completeness, and consistency of the data, which are essential factors for obtaining meaningful results.

5.2.4 Fake reviews dataset

The dataset is in csv format, consists of 40,000 instances, evenly distributed between two classes: 20,000 instances of original reviews and 20,000 instances of computer-generated fake reviews. Each instance in the dataset includes features such as category, rating, text, and label. Dataset is available here. Below you can find a brief overview of the dataset.

	category	rating	label	text_
0	Home_and_Kitchen_5	5.0	OR	I bought a protective glove for when I use thi...
1	Books_5	5.0	OR	Since the first book of this series was writte...
2	Toys_and_Games_5	5.0	CG	perfect size for my two year old granddaughter...
3	Books_5	5.0	OR	I love this series. This is the second time I...
4	Tools_and_Home_Improvement_5	4.0	CG	These bulbs are shipped from China. The shippi...

Figure 5.5: Subset of fake reviews dataset

5.2.4.1 Preprocessing

The preprocessing stage plays a crucial role in preparing the data for analysis. We specifically eliminated all columns except for "text" and "label" since these were the only features relevant to our study. Additionally, we addressed missing values and conducted other essential cleaning procedures to enhance the data quality.

Ensuring the accuracy, completeness, and consistency of the data is important during the preprocessing stage. This is vital to obtain meaningful and reliable results.

5.3 Implementation tools

With the rapid advancements in artificial intelligence, there is an increasing need for efficient and effective implementation of complex and large-scale models. To address this, there are now various open source machine learning frameworks available that aim to simplify the implementation and optimization of such models. These frameworks provide a user-friendly interface that allows researchers and practitioners to easily access and utilize cutting-edge machine learning techniques, without having to worry about the underlying complexities of the technology. By leveraging these frameworks, developers can focus more on the design and optimization of their models, rather than spending excessive time on the implementation details. This not only saves time but also facilitates faster innovation and experimentation, ultimately leading to the development of more sophisticated and accurate models.

5.3.1 Software

5.3.1.1 TensorFlow

TensorFlow is an open source software library developed by Google that provides a platform for building and training machine learning models, particularly deep neural networks. It provides a range of tools, APIs and resources that allow developers to construct, train and deploy machine learning models for a wide range of applications. TensorFlow is highly flexible and can be used with a variety of programming languages such as Python, C++, and Java. It is

widely used in various fields such as image and speech recognition, natural language processing, and robotics, and has become one of the most popular and powerful tools for developing and deploying machine learning models.

5.3.1.2 sklearn

scikit-learn (sklearn) is a comprehensive Python library for machine learning. It offers a wide range of algorithms and tools for tasks such as classification, regression, clustering, and dimensionality reduction. With its user-friendly interface, scikit-learn simplifies the process of building and evaluating machine learning models. It provides modules for data preprocessing, feature extraction, and model evaluation, making it suitable for various applications. The library emphasizes code efficiency and scalability, enabling efficient processing of small and large datasets. With extensive documentation and a vibrant community, scikit-learn has become a popular choice for machine learning practitioners and researchers alike.

5.3.1.3 Keras

Keras is an open source high-level neural network API (Application Programming Interface) written in Python that provides a user-friendly interface for building and training deep learning models. It was developed with the goal of enabling fast experimentation with deep neural networks, making it easier for researchers and developers to design, prototype, and deploy deep learning models in a quick and efficient manner. Keras supports various popular deep learning frameworks as its backends, including TensorFlow, CNTK, and Theano, and offers a wide range of built-in modules for developing different types of neural networks such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and more. With its ease of use and powerful capabilities, Keras has become a popular tool for deep learning practitioners and researchers alike.

5.3.1.4 Flask

Flask is a popular web framework for building Python web applications. It is designed to be simple and lightweight, and provides a range of tools for building and deploying web applications quickly and easily. Flask includes support for routing and request handling, template rendering, and database integration, and provides a range of extensions that can be used to add additional functionality to the framework, such as support for user authentication and authorization, caching, and API development. Flask is particularly well-suited for building microservices and other small-scale web applications.

5.3.2 Programming languages

5.3.2.1 Python

Python is a programming language that uses dynamic semantics, is object-oriented, and is interpreted. Its high-level built-in data structures and dynamic typing and binding make it an ideal choice for fast application development and for connecting existing components as a scripting or glue language. The language's syntax is easy to read and learn, which makes it easy to maintain programs in the long run. Additionally, Python supports modules and packages, which helps to organize code and encourage code reuse. Python is available for all major platforms in both source and binary form, and can be freely distributed as it comes with an extensive standard library.

5.3.2.2 HTML

HTML (Hypertext Markup Language) is a markup language used for creating and structuring web pages and other electronic documents that can be displayed on a web browser.

5.3.2.3 CSS

CSS (Cascading Style Sheets) is a stylesheet language used for describing the visual presentation of a document written in HTML or XML. It is used to define the layout, color, font, and other visual aspects of a web page.

5.3.2.4 JavaScript

JavaScript is a programming language commonly used for client-side web development that allows for interactive and dynamic effects on web pages. It can be used to manipulate the Document Object Model (DOM) of a webpage, handle user input, and communicate with servers to dynamically update the content of a webpage without the need for a page reload.

5.3.2.5 SQL (Structured Query Language)

SQL is a specialized language utilized for the management and manipulation of relational databases. It offers a collection of commands and syntax that facilitate the creation and modification of database structures, as well as the insertion and retrieval of data. Additionally, SQL enables the execution of intricate queries and analyses. It finds extensive application in enterprise and data-driven web applications and enjoys compatibility with various relational database management systems such as MySQL, PostgreSQL, and Oracle.

5.3.3 Hardware

Google Colab, an open source research tool, is designed for teaching and conducting machine learning research. It provides a Jupyter notebook environment that eliminates the need for installation or configuration since it operates entirely in the cloud. This environment is compatible with Python 2.0 and its subsequent versions. It enables users to write and execute code, save and share their work, and utilize robust computing resources directly from a web browser. For the experiments conducted in this study, a virtual machine on Google's Colab platform was used, which delivers excellent performance. The specific characteristics of this environment are outlined below:

CPU	Intel(R) Xeon(R) CPU @ 2.20GHz
GPU	Tesla T4
RAM	12.7GB

Table 5.1: Colab configuration

5.4 Evaluation metrics

5.4.1 Confusion matrix

A confusion matrix is a performance measurement tool used in machine learning and statistical classification tasks. It summarizes the results of a classification model by tabulating the predicted labels against the true labels of a dataset. The matrix provides insights into the accuracy and errors made by the model for each class. In the case of a binary classification problem, the confusion matrix exhibits four potential results:

- True Positive (TP): This refers to the cases where the model correctly predicts the positive class when the true class is indeed positive. In other words, the model correctly identifies the presence of a condition or event.

- False Negative (FN): This occurs when the model incorrectly predicts the negative class when the true class is positive. It means that the model fails to identify the positive class or misses the presence of a condition or event.

- False Positive (FP): This happens when the model incorrectly predicts the positive class when the true class is negative. It means that the model identifies a condition or event that is not present.

- True Negative (TN): This refers to the cases where the model correctly predicts the negative class when the true class is indeed negative. It means that the model accurately recognizes the absence of a condition or event.

	Positive	Negative
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

Table 5.2: Confusion Matrix

5.4.2 Accuracy

Accuracy measures the proportion of correctly classified samples to the total number of samples in the dataset. It is calculated as

$$\frac{TruePositives + TrueNegatives}{TruePositives + FalsePositives + TrueNegatives + FalseNegatives}$$

5.4.3 Precision

Precision measures the proportion of true positive predictions among the total positive predictions. It is calculated as

$$\frac{TruePositives}{TruePositives * FalsePositives}$$

.

5.4.4 Recall

Recall measures the proportion of true positive predictions among the total positive samples in the dataset. It is calculated as

$$\frac{TruePositives}{TruePositives + FalseNegatives}$$

.

5.4.5 F1 score

F1-score is the harmonic mean of precision and recall, and it provides a balanced measure of the model's performance. It is calculated as

$$2 * \frac{Precision * Recall}{Precision + Recall}$$

.

5.5 Experimentation and implementation

5.5.1 Fake news detection

To evaluate their performance, we performed experiments using two models: NadraGPT-2 and Nadra Embedding Convolutional Model. We compared their accuracy and loss function in order to identify the model with the superior performance.

5.5.1.1 Using Nadra Embedding Convolutional Model

We have done several experiments with different batch sizes, learning rates and optimizers.

5.5.1.1.1 Learning rate

In this implementation, we are conducting experiments by varying the learning rate to observe how it affects the performance of the model.

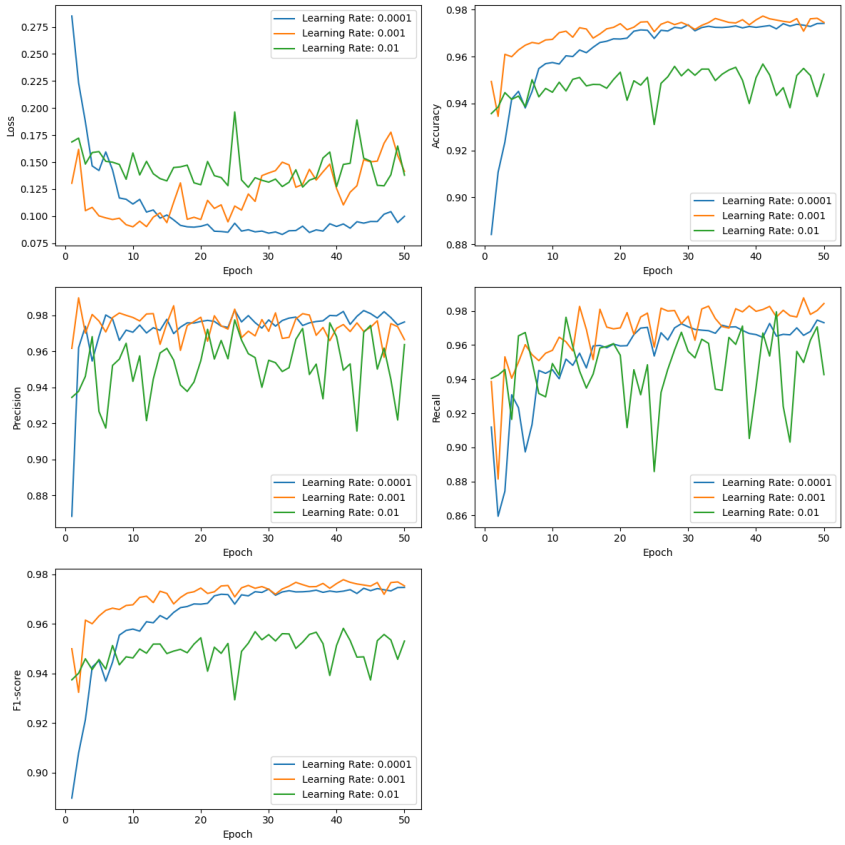


Figure 5.6: Nadra Embedding Convolutional Model with different learning rates

Among the three learning rates (0.0001, 0.001, 0.01) examined, a learning rate of 0.001 showed slightly better accuracy and stable precision. However, it was not the best in terms of loss. The learning rate of 0.0001 performed well in minimizing loss but had lower performance in other metrics. A learning rate of 0.01 performed poorly across all evaluated metrics. Therefore, the choice of learning rate significantly impacts model performance, and in this case, 0.001 struck a balance between accuracy and precision.

5.5.1.1.2 Batch size

In this particular implementation, we are conducting experiments using varying batch sizes.

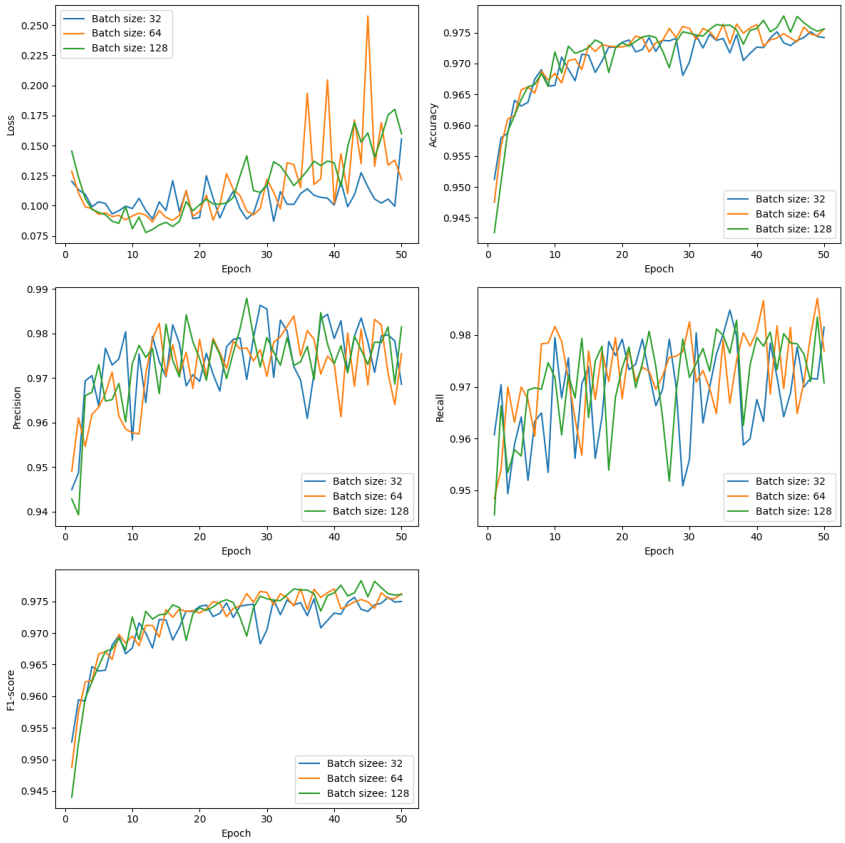


Figure 5.7: Nadra Embedding Convolutional Model with different batch sizes

The analysis of the plots for precision, recall, loss, and accuracy for different batch sizes reveals interesting insights. In terms of loss, the batch size of 128 demonstrated a slight advantage, indicating that a larger batch size may contribute to better convergence. On the other hand, when it comes to precision, the batch size of 128 emerged as the best performer, suggesting that it resulted in a higher proportion of true positive predictions. In terms of accuracy, the batch sizes of 64 and 128 were closely competitive, indicating that both batch sizes yielded similar levels of overall correctness in predictions. Finally, for recall, all the batch sizes showed comparable performance, indicating that they were all capable of capturing a similar proportion of true positives, but 64 had a bit of better recall. Overall, these results highlight the importance of considering different batch sizes when training a model, as they can have varying effects on different evaluation metrics.

5.5.1.1.3 Optimizers

In this implementation, we are exploring various optimizers to examine their effects.

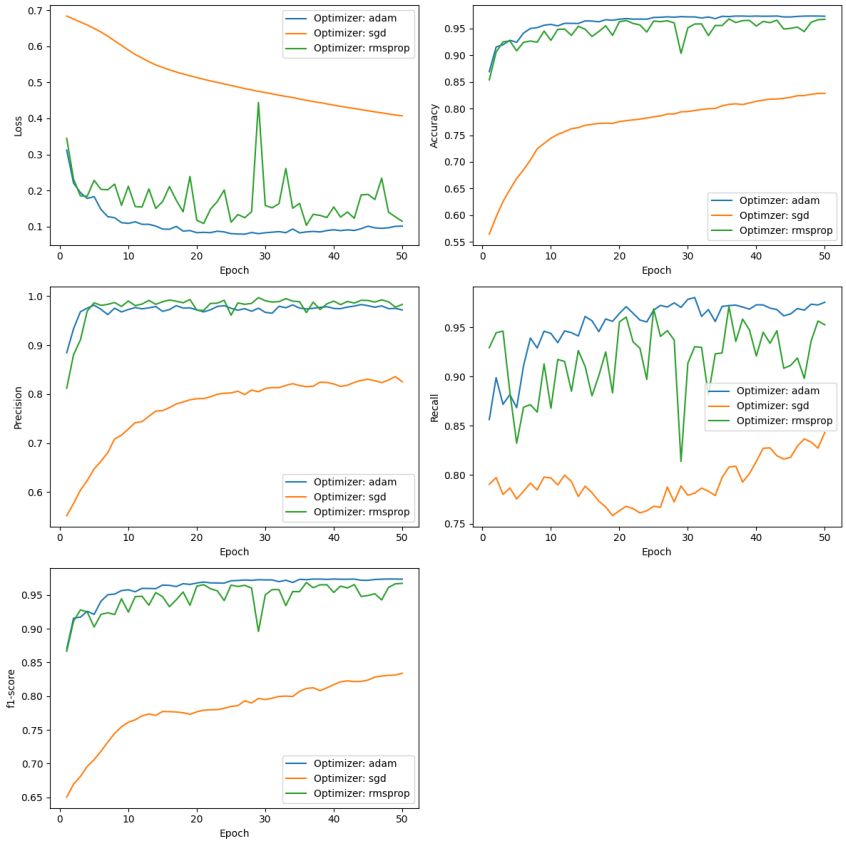


Figure 5.8: Nadra Embedding Convolutional Model results with different optimizers

Upon analyzing the plots for loss, accuracy, precision, and recall for different optimizers (SGD, Adam, and RMSprop), it is evident that the Adam and RMSprop optimizers performed the best. The results indicate that Adam slightly outperformed the other two optimizers in terms of loss, accuracy, and recall. However, when it comes to precision, RMSprop exhibited slightly better performance than Adam.

5.5.1.2 Final result

Figure 5.9 represents different evaluation metrics and the loss function for our model after the experimentation we obtain the result in table 5.3

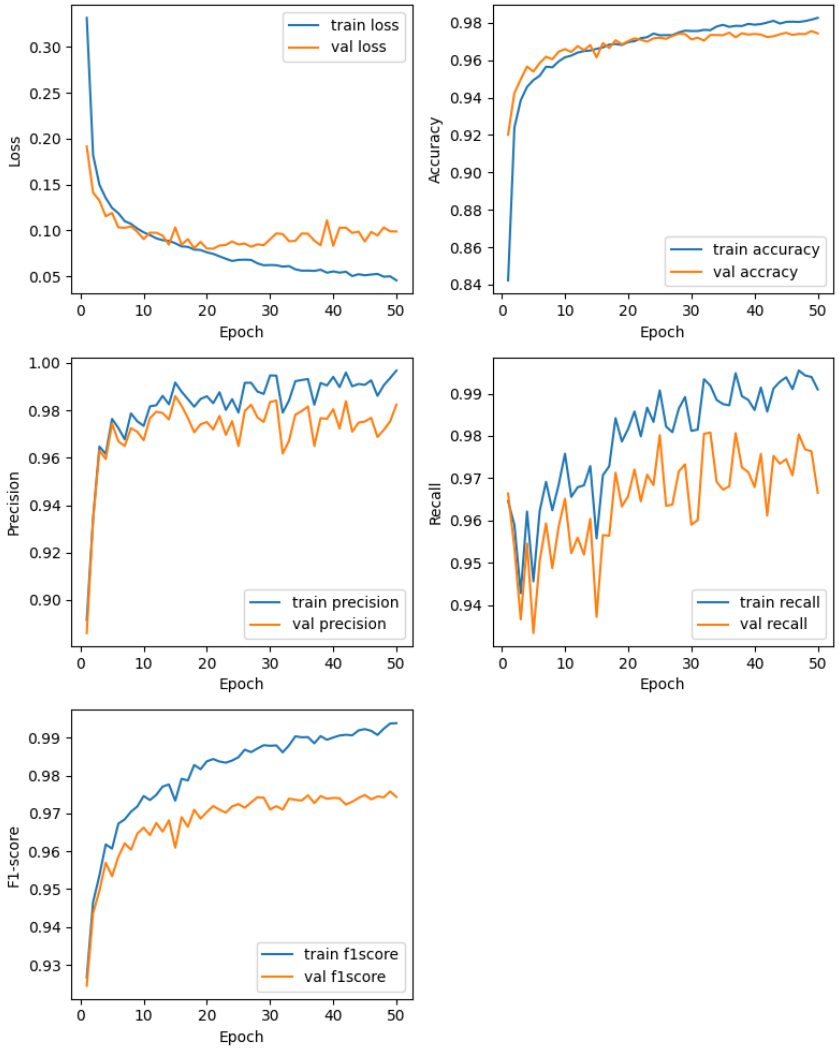


Figure 5.9: Train and validation results.

Accuracy	Precision	Recall	F1-score	Loss
0.96	0.98	0.96	0.97	0.10

Table 5.3: Final result

5.5.1.3 Using NadraGPT-2

We have performed various experiments involving different batch sizes, learning rates and optimizers.

5.5.1.3.1 Batch size

In this specific implementation, we are performing experiments by utilizing different batch sizes.

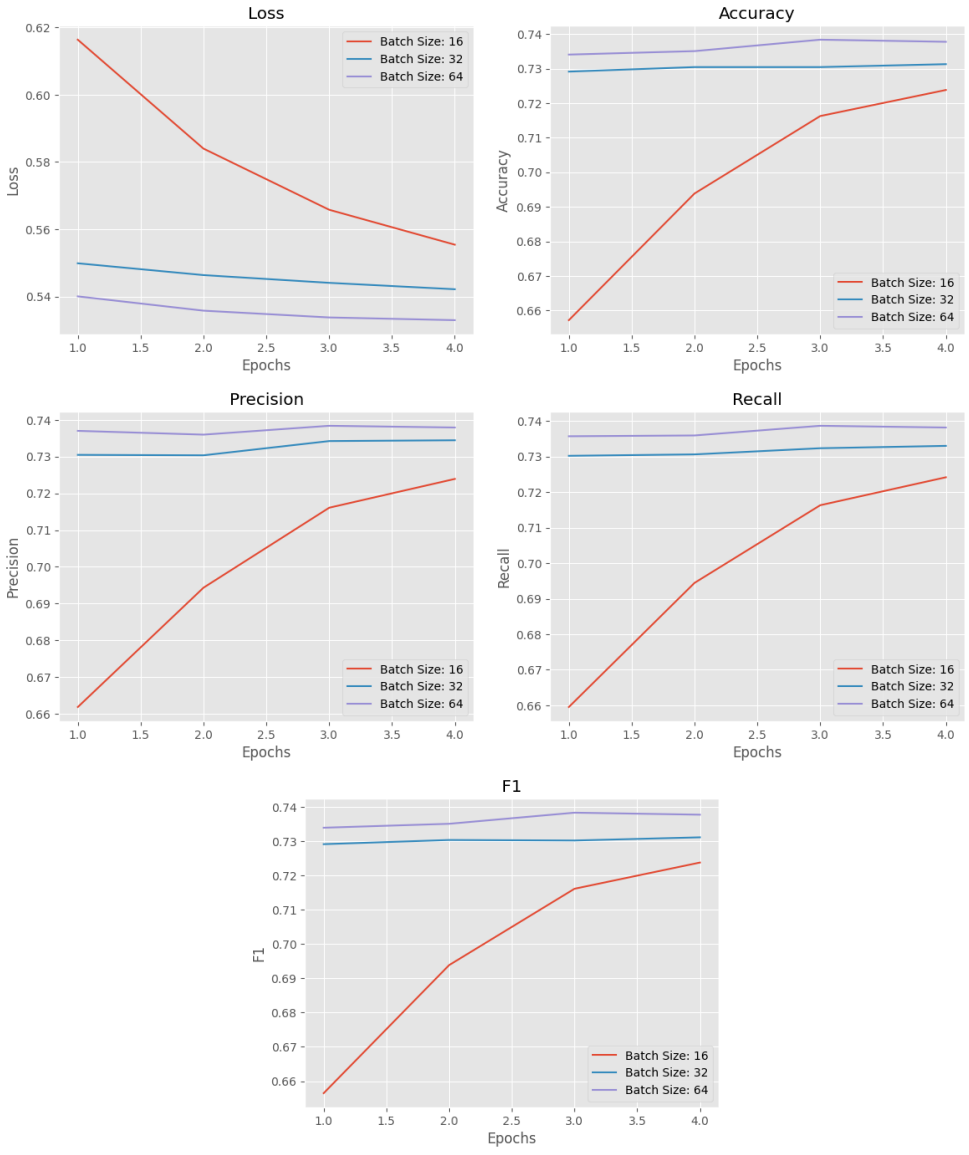


Figure 5.10: NadraGPT-2 results with different batch sizes

The results of the NadraGPT-2 fine-tuning experiment, where different batch sizes were evaluated, indicate that batch size has a significant impact on model performance. The plot of loss, accuracy, precision, recall, and F1 score for different batch sizes reveals that smaller batch sizes generally lead to poorer performance across all evaluation metrics. Conversely, a batch size of 64 consistently yielded the best results across all metrics.

5.5.1.3.2 Optimizers

In this specific implementation, we are performing experiments by utilizing different optimizers.

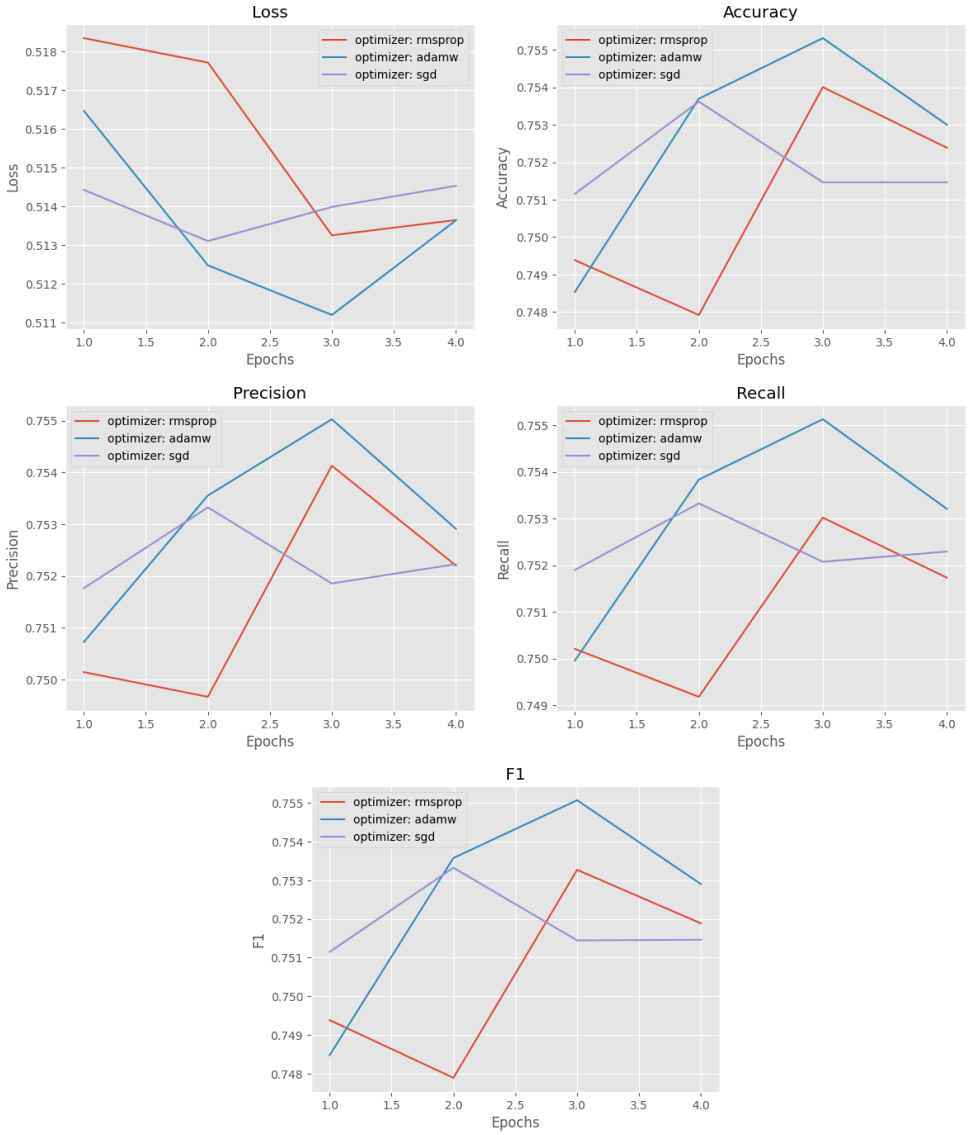


Figure 5.11: NadraGPT-2 results with different optimizers

The results of comparing three different optimizers, namely AdamW, RMSprop, and SGD, for NadraGPT-2 fine-tuning reveal that AdamW showcased a slight advantage over the other two optimizers in terms of loss and the evaluated metrics. The superior performance of AdamW can be attributed to its adaptive learning rate mechanism, which allows for efficient parameter updates during training. AdamW combines the benefits of adaptive learning rates from Adam with weight decay, which helps prevent overfitting. This combination likely contributed to its effectiveness in optimizing the NadraGPT-2 model for the specific fine-tuning task.

5.5.1.4 Final result

Figure 5.12 illustrates the accuracy and loss function of our model for both the training and validation sets.

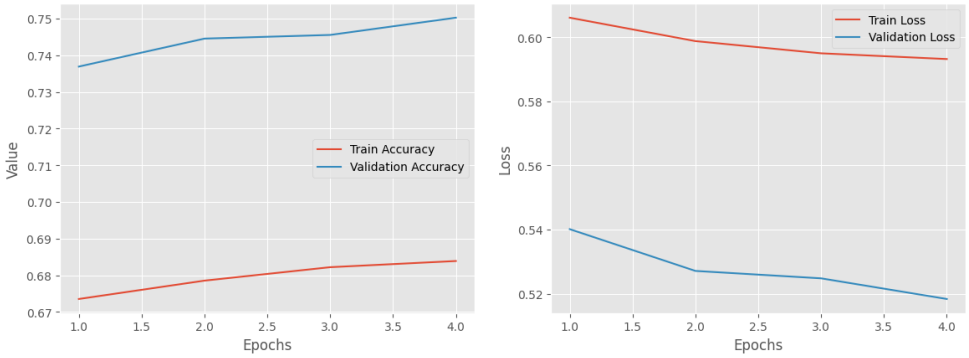


Figure 5.12: NadraGPT-2 final result

The results show that the model's predictions were generally close to the true values, as indicated by the low loss values. However, there is room for improvement to reduce the gap between the training and validation accuracies. Below is the confusion matrix for the test set.

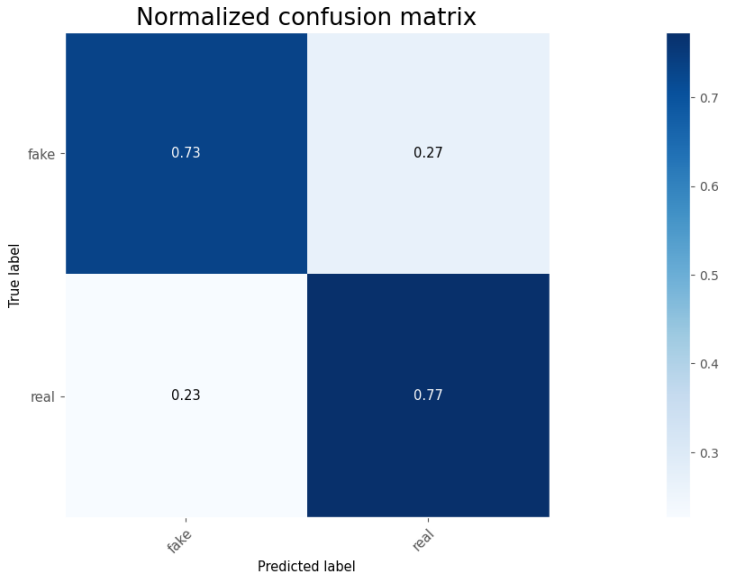


Figure 5.13: NadraGPT-2 confusion matrix

5.5.1.5 Comparison between Nadra Embedding Convolutional Model and NadraGPT-2

As we have shown in the figures above, Nadra Embedding Convolutional Model has given us better results than NadraGPT-2 in the fake news classification due the fact that GPT-2 is better at text generation and doesn't work quite well on text classification. CNN models can achieve good accuracy for fake news detection tasks, especially when trained on large and diverse labeled datasets. They tend to generalize well to new data with similar patterns, but may struggle when encountering novel or unseen types of fake news.The following table 5.4 shows the performance of each model.

	Loss	Accuracy	Precision	Recall	F1 score	Parameters
Nadra Embedding Convolutional Model	0.1	96%	98%	96%	97%	1,842,849M
NadraGPT-2	0.1	75.3%	74.8%	75.28	75.2%	124,671,362M

Table 5.4: Performance measures of Nadra Embedding Convolutional Model and NadraGPT-2 on fake news dataset

5.5.1.6 Final configuration

Following our experimental evaluations, we have established the optimal parameter configuration, which is presented in Table 5.5.

Batch size	**Optimizer**	**Learning rate**	**Model**
64	Adam	0.001	Nadra Embedding Convolutional Model

Table 5.5: Final configuration

5.5.1.7 Comparison between Nadra Embedding Convolutional Model and existing models

In the fight against fake news, our model has proven to be more accurate and effective compared to the existing model in 2.3.3.8. We have trained our model using the latest advancements in artificial intelligence and natural language processing. Additionally, our model has a better understanding of the context in which information is presented, allowing it to detect subtle signs of deception. In head-to-head comparisons, the table below demonstrates a comparison between the two.

Method	**Accuracy**
TF-idf on unigrams and bigrams with cosine similarity fed into dense neural network [31]	94.31%
Nadra Embedding Convolutional Model (ours)	96%

Table 5.6: Comparison between Nadra Embedding Convolutional Model and an existing model

5.5.2 Fake reviews detection

We conducted experiments with two models, namely NadraGPT-2 and NadraBERT, and compared their accuracy and loss function to determine the best performance.

5.5.2.1 Using NadraBERT

The model has achieved accuracy (90% on training, 92.8% on validation) and low loss (0.23 for training, 0.24 for validation). This means the model is performing well, making decent predictions and minimizing errors. It shows that the model has learned effectively and can generalize its predictions to new data.

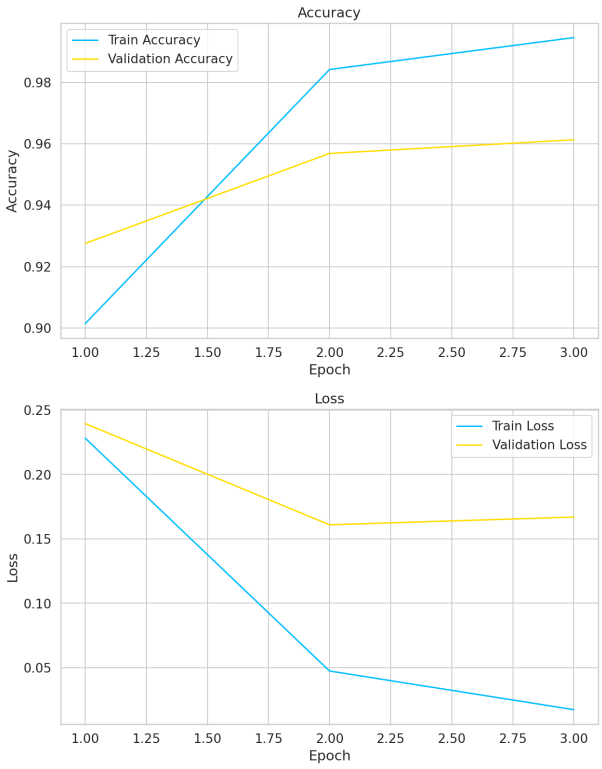


Figure 5.14: The accuracy and the loss for NadraBERT.

Optimizer	AdamW
Weight decay	0.04
Learning rate scheduler	Cosine Annealing
Maximum learning rate	0.00002
Minimum learning rate	0.000001
Number of warmup steps	100
Max length	30
Dropout	0.4
Batch size	64

Table 5.7: Hyperparameters

5.5.2.2 Using NadraGPT-2

The model has learned to perform exceptionally well on the training data but fails to generalize to unseen data. As a result, the model's performance on the training data keeps improving, leading to a decrease in the training loss and an increase in the training accuracy. However, when evaluated on new and unseen data, the model's performance drops significantly, reflected by a higher loss and lower accuracy.

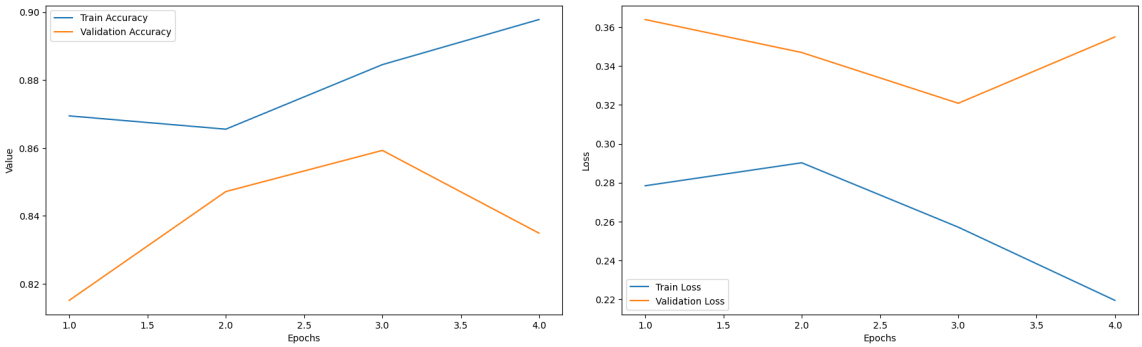


Figure 5.15: The accuracy and the loss for NadraGPT-2.

Optimizer	AdamW
Weight decay	0.05
Maximum learning rate	0.00006
Learning rate scheduler	Linear learning rate scheduler
Number of warmup steps	100
Batch size	64
Max length	30
Dropout	0.4

Table 5.8: Hyperparameters

5.5.2.3 Comparison between NadraBERT and NadraGPT-2

As we have shown in the figures above, NadraBERT has given us better results than NadraGPT-2 in the fake reviews classification due the fact that GPT-2 is good at text generation and doesn't work quite well on text classification. BERT on the other hand is designed specifically for that task. BERT can understand the meaning of words in context better than GPT-2 because it looks at the words that come before and after. It's like having a better understanding of the overall story. BERT also learns from a lot of text, so it knows a wide range of words and how they fit together.

	Loss	Accuracy	Precision	Recall	F1 score	Number of parameters
NadraBERT	0.21	94.43%	94.83%	94.43%	94.42%	109,482,240M
NadraGPT-2	0.33	85%	80%	85%	85%	124,671,362M

Table 5.9: Performance measures of NadraBERT and NadraGPT-2 on fake reviews dataset

5.5.3 Fake NFT detection

5.5.3.1 Using Incoherent Pixels Technique

We conducted a series of experiments to explore the effectiveness of (IPT) for detecting fake NFTs. The experiments involved varying of dense layers, ranging from 1 to 3. The objective is to detect the incoherent pixels between the different parts of the image in order to prove that the image is fake. Different configurations of batch sizes, learning rates, and optimizers were also tested. Additionally, data augmentation techniques were applied to enhance the dataset. After many experimentations, the best results are grouped in the next figures and tables.

5.5.3.2 Learning rate & batch size

To understand the impact of learning rate and batch size on the performance of the model, we performed experiments with varying values. Specifically, we explored learning rates of 0.0001, 0.001, and 0.01, and batch sizes of 16, 32, and 64. The goal was to observe how different combinations of these hyperparameters affected the model's detection accuracy and convergence behavior.

5.5.3.3 Optimizers

Another aspect of our investigation focused on exploring the effects of different optimizers on fake NFT detection. We experimented with several popular optimizers, namely Adam, RMSprop, and SGD, to evaluate their impact on the model's performance. By comparing the detection accuracy and training convergence achieved with each optimizer, we aimed to identify the most suitable optimizer for our task.

5.5.3.4 Best results

We ll display bellow our experiments with different optimizers, learning rates & batch sizes, as well as emphasizing on our best results after analysing.

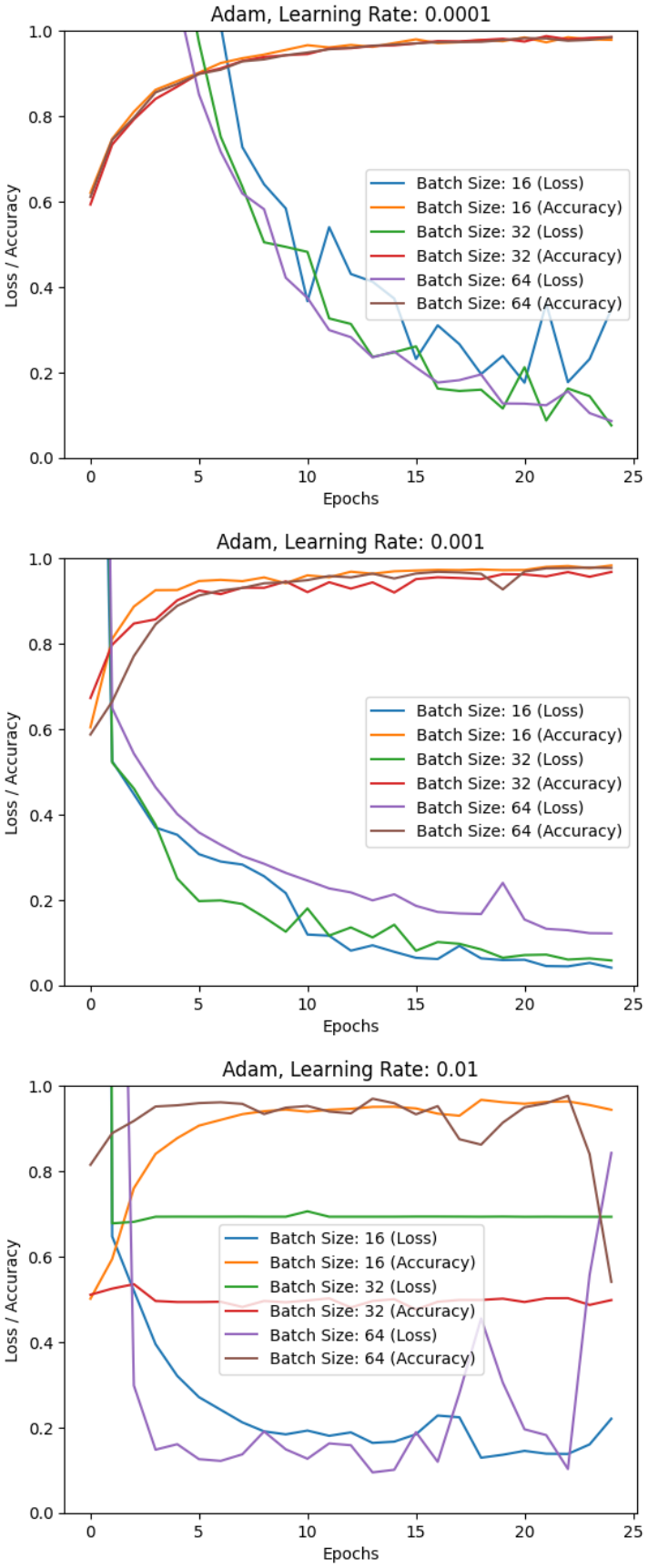


Figure 5.16: Results with Adam optimizer with different learning rates & batch sizes.

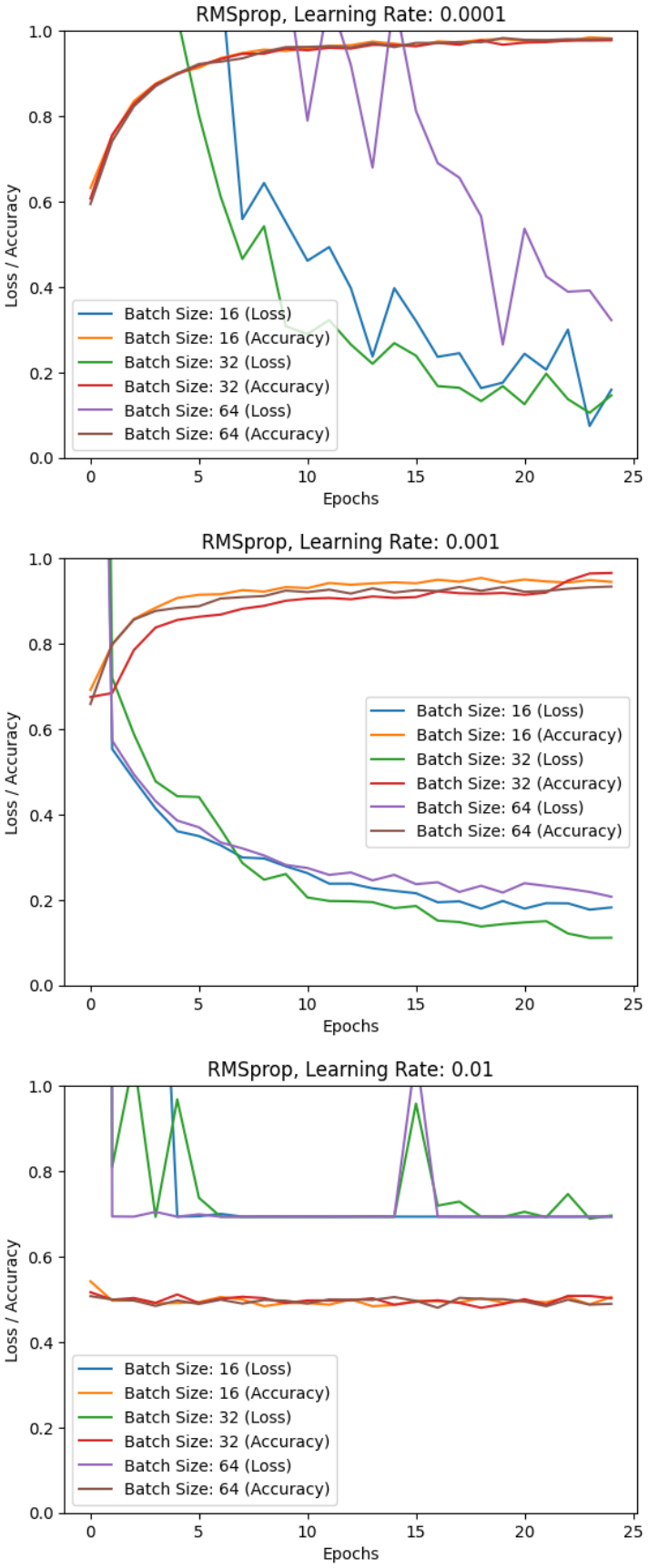


Figure 5.17: Results with RMSprop optimizer with different learning rates & batch sizes.

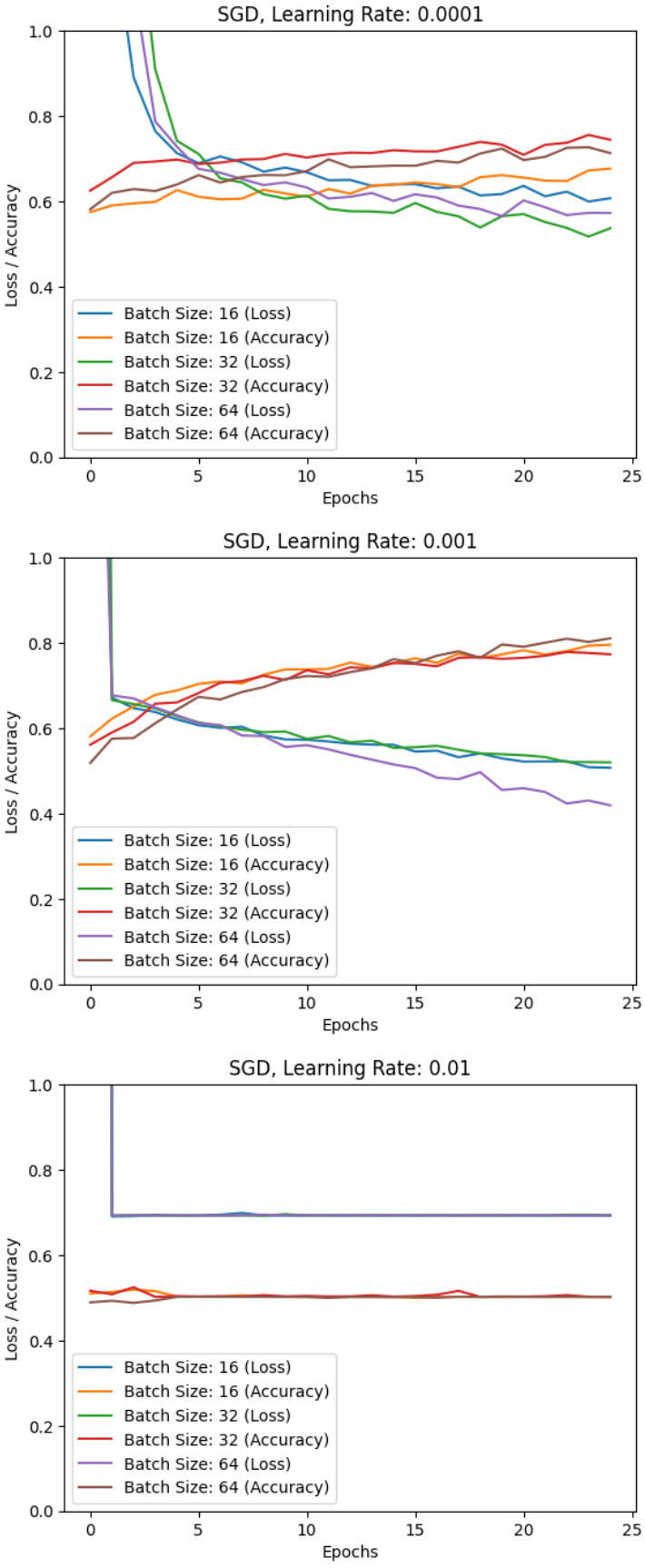


Figure 5.18: Results with SGD optimizer with different learning rates & batch sizes.

Upon analyzing the plots emphasizing Loss/Accuracy for different optimizers (ADAM, RMSprop & SGD), learning rates (0.0001, 0.001, 0.01) & batch sizes (16, 32, 64). It is evident that the Adam and RMSprop optimizers performed the best while SGD performed terribly. The results indicate that Adam slightly outperformed RMSprop optimizer in terms of loss and accuracy,

Lets dive into more details:

- **Adam optimizer:**
  - **with learning rate = 0.0001:** The best results were with a batch size of 32 with an accuracy of 98.01% and a loss of 0.08%
  - **with learning rate = 0.001:** The best results were with a batch size of 16 with an accuracy of 98.58% and a loss of 0.05%
  - **with learning rate = 0.01:** With this learning rate, Adam didn't give good results compared to the other two learning rates.
- **RMSprop optimizer:**
  - **with learning rate = 0.0001:** The best results were with a batch size of 32 with an accuracy of 98.38% and a loss of 0.17%
  - **with learning rate = 0.001:** The best results were with a batch size of 32 with an accuracy of 93.10% and a loss of 0.12%
  - **with learning rate = 0.01:** With this learning rate, RMSprop didn't give good results compared to the other two learning rates.
- **SGD optimizer:** With all the different learning rates & batch sizes, SGD optimizer didn't give satisficing results.

5.5.4 Fake Gold detection

5.5.4.1 Using Incoherent Pixels technique (IPT)

After many experiments, we conducted that the use of IPT gave the best results.

5.5.4.2 Learning rate & batch size

To understand the impact of learning rate and batch size on the performance of the model, we performed experiments with varying values. Specifically, we explored learning rates of 0.0001, 0.001, and 0.01, and batch sizes of 8, 16, 32 & 64. The goal was to observe how different combinations of these hyperparameters affected the model's detection accuracy and convergence behavior.

5.5.4.3 Optimizers

Another aspect of our investigation focused on exploring the effects of different optimizers on fake Gold detection. We experimented with several popular optimizers, namely Adam, RMSprop, and SGD, to evaluate their impact on the model's performance. By comparing the detection accuracy and training convergence achieved with each optimizer, we aimed to identify the most suitable optimizer for our task.

5.5.4.4 Best results

We ll display bellow our best results.

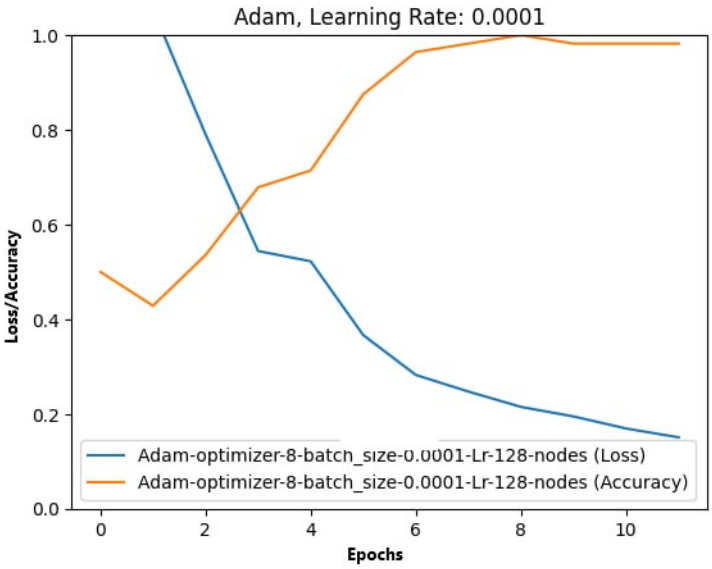


Figure 5.19: The Loss/Accuracy plot

Table 5.10: Gold Model Performance Metrics

Metric	Value
Loss	0.1343
Accuracy	0.9683
Precision	1.0000
Recall	0.9167
F1-score	0.9565

As we see, the best results were given using the Adam optimizer, a learning rate of 0.0001 & a batch size of 8 with the metrics specified in the table above.

5.6 Conclusion

This chapter offers a comprehensive overview of the evaluation measures employed in our research, accompanied by a detailed analysis of the results obtained from our experimental investigations. The implications and significance of these findings are thoroughly examined and explained. Moreover, we delve into the implementation stages of our work, providing an in-depth insights into the key methodologies and techniques utilized. Furthermore, we conduct a comparative study to assess the performance of various fraud detection models in the domains of fake news, fake reviews, fake gold, and fake NFTs, aiming to identify the most effective approach. Lastly, we compare NadraGPT-2 and Nadra Embedding Convolutional models, focusing on their respective strengths and weaknesses in the realm of fake news detection.

Chapter 6
Nadra's Platform

Chapter 6

Nadra's Platform

6.1 Web Application

6.1.1 Home Page

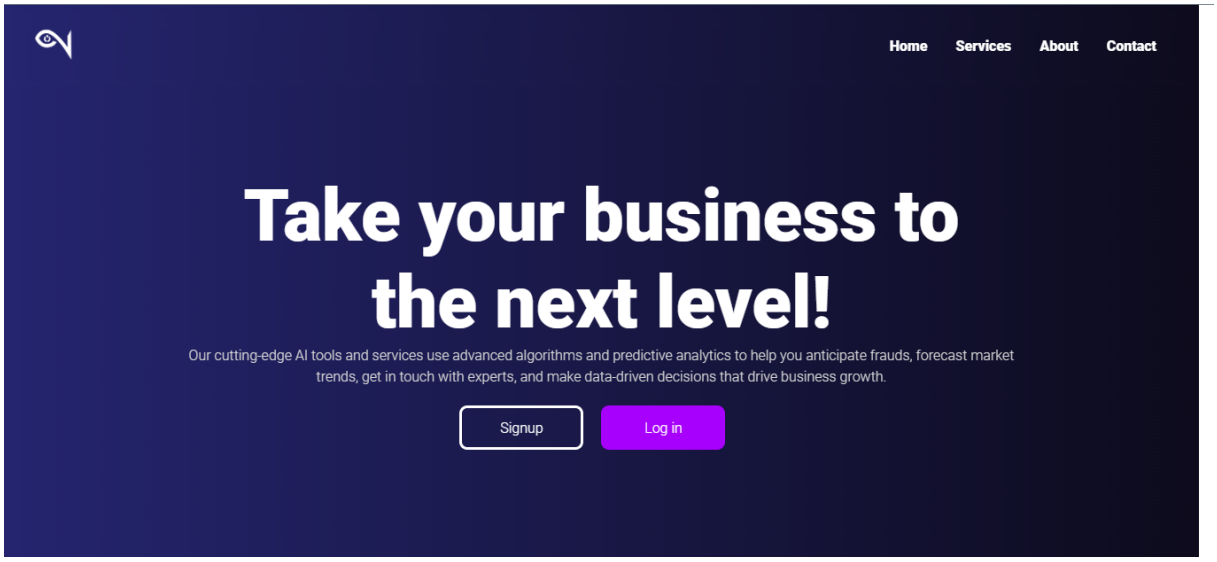


Figure 6.1: Home Page

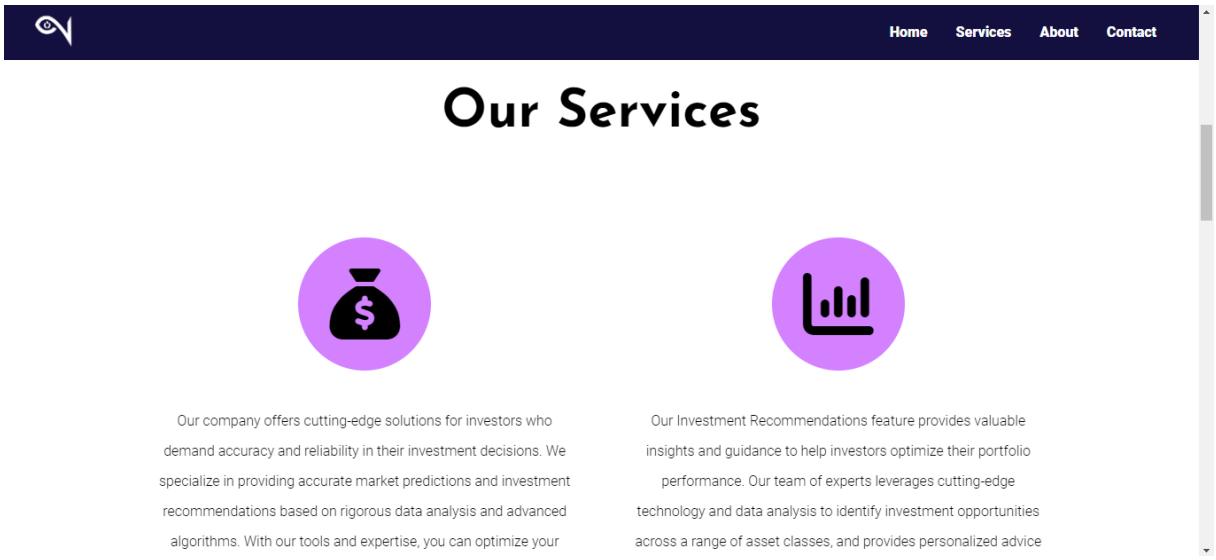


Figure 6.2: Our Services Page

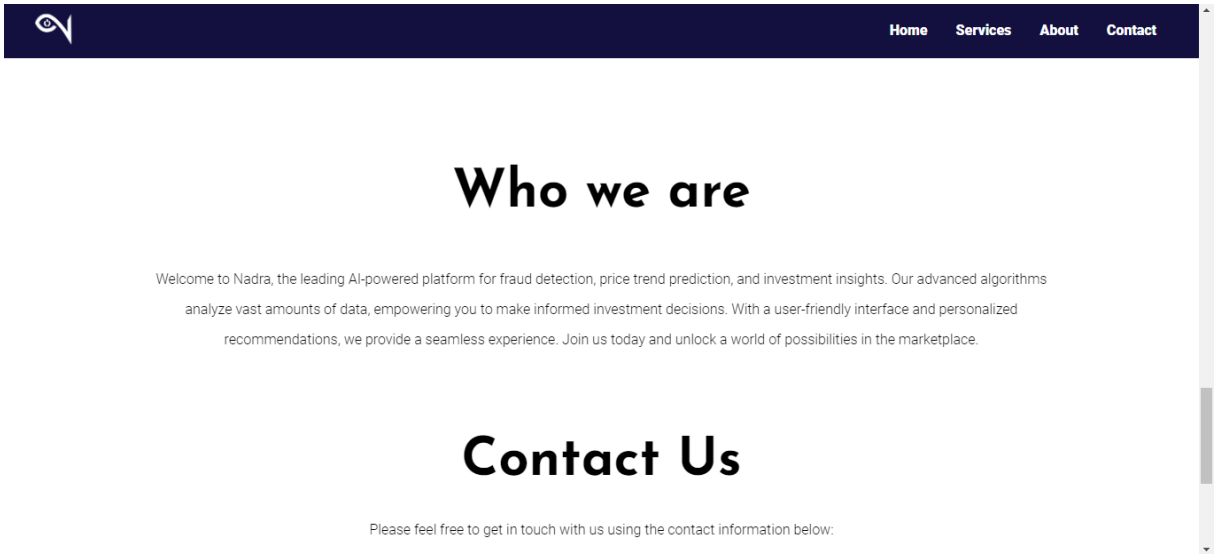


Figure 6.3: About Us Page

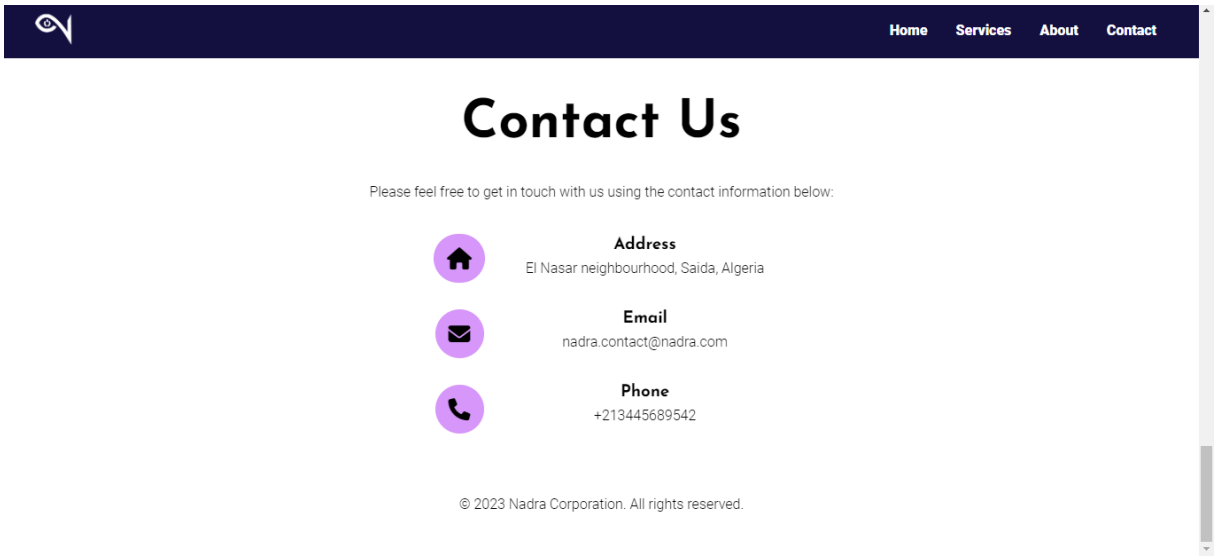


Figure 6.4: Contact us Page

6.1.2 Sign Up Page

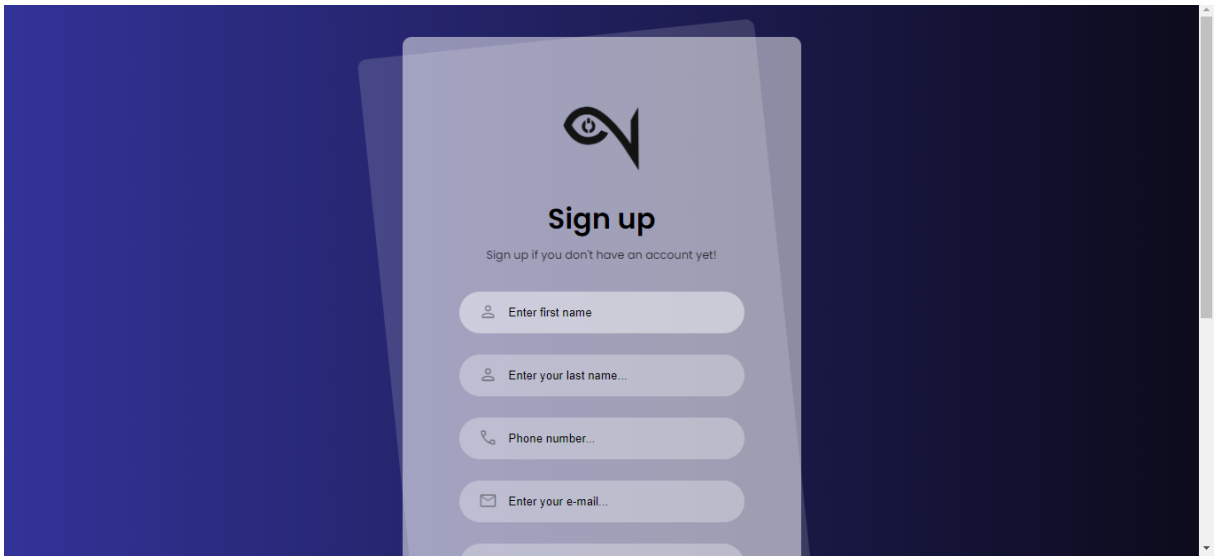


Figure 6.5: Sign Up Page

6.1.3 Sign In Page

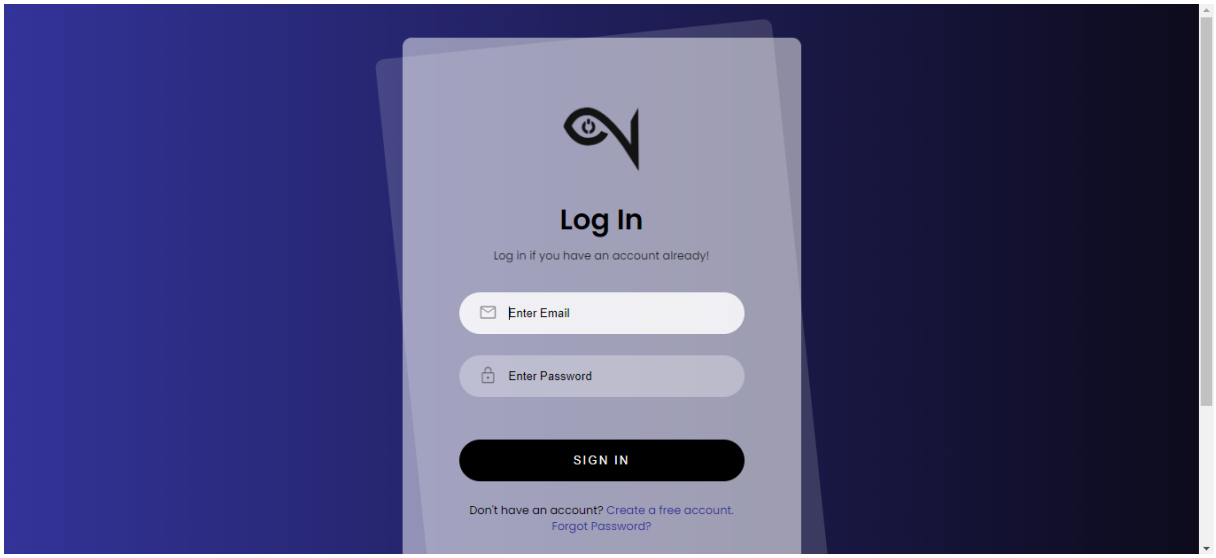


Figure 6.6: Sign In Page

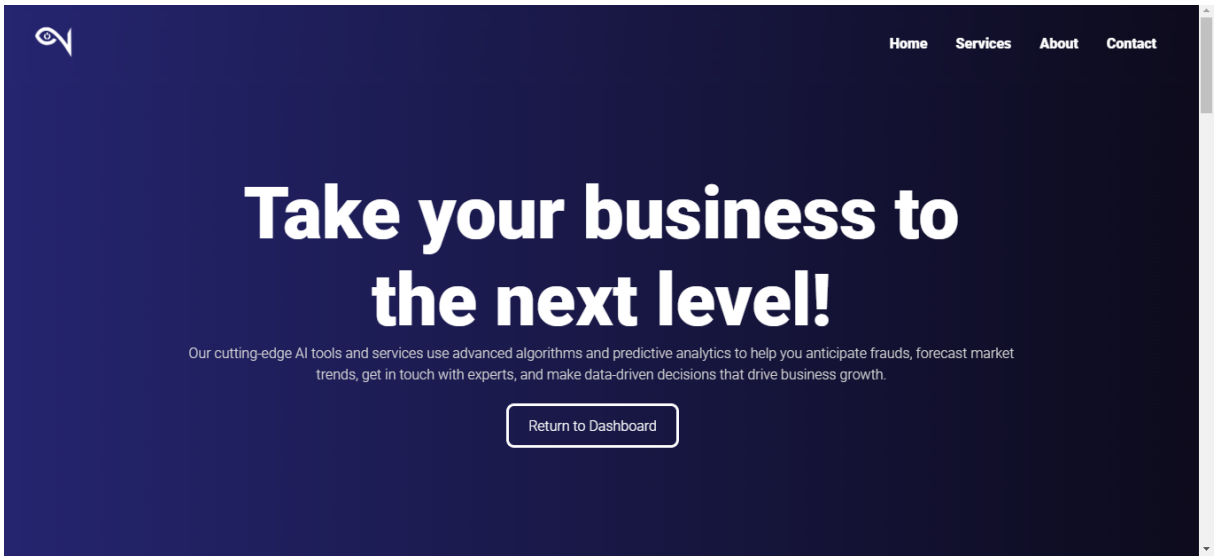


Figure 6.7: Home after Signing In

6.1.4 Services Page

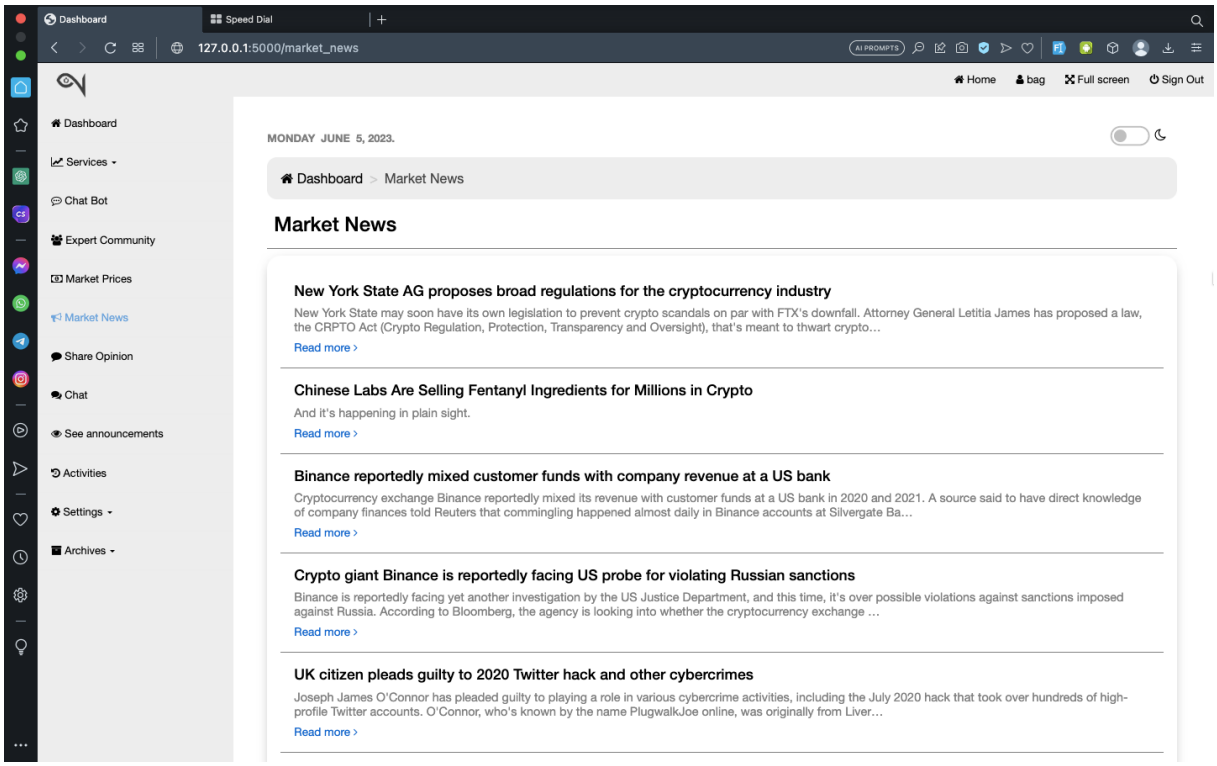


Figure 6.8: Market News Page

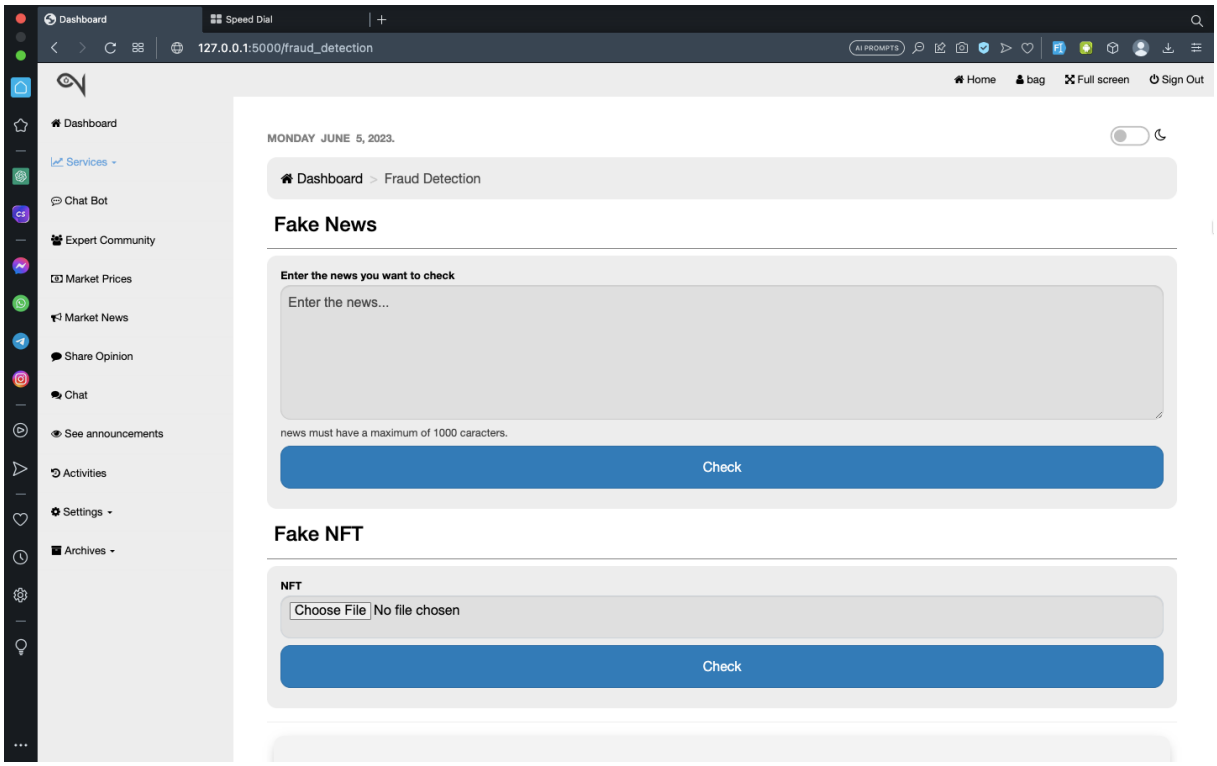


Figure 6.9: Fraud Detection Services Page

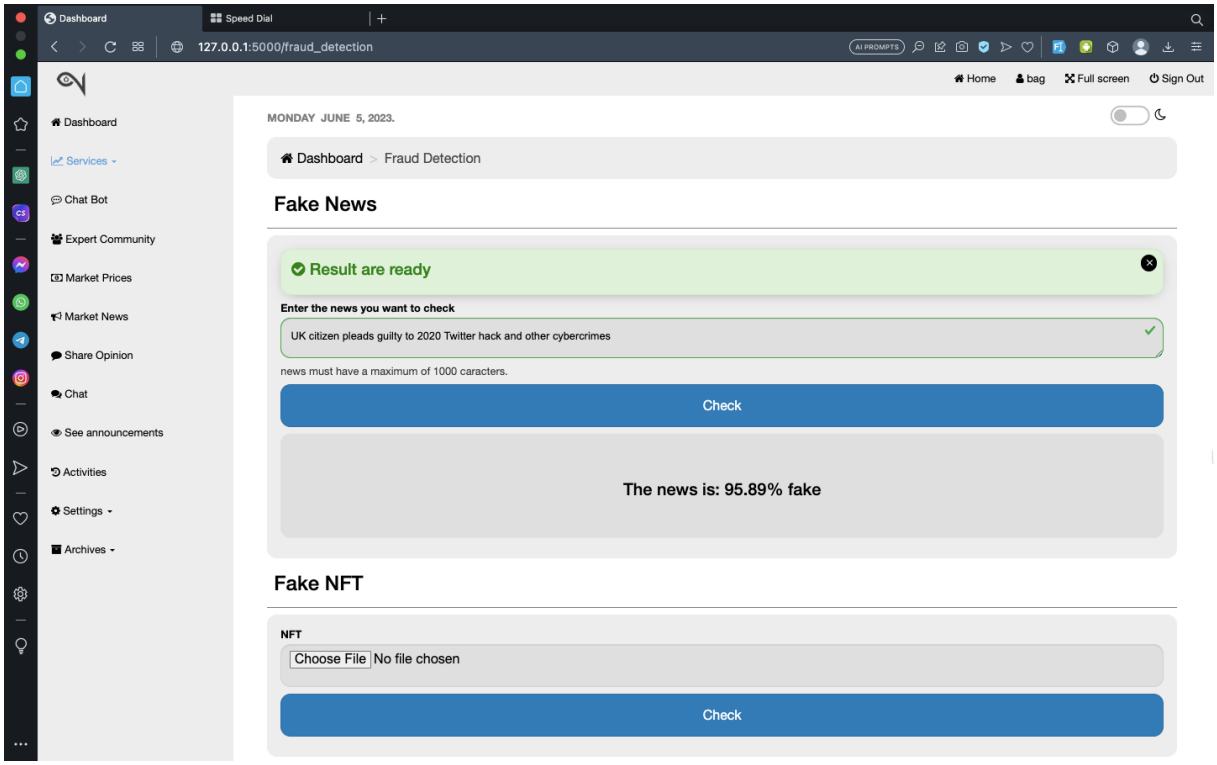


Figure 6.10: Nadra Fake News Detection Service

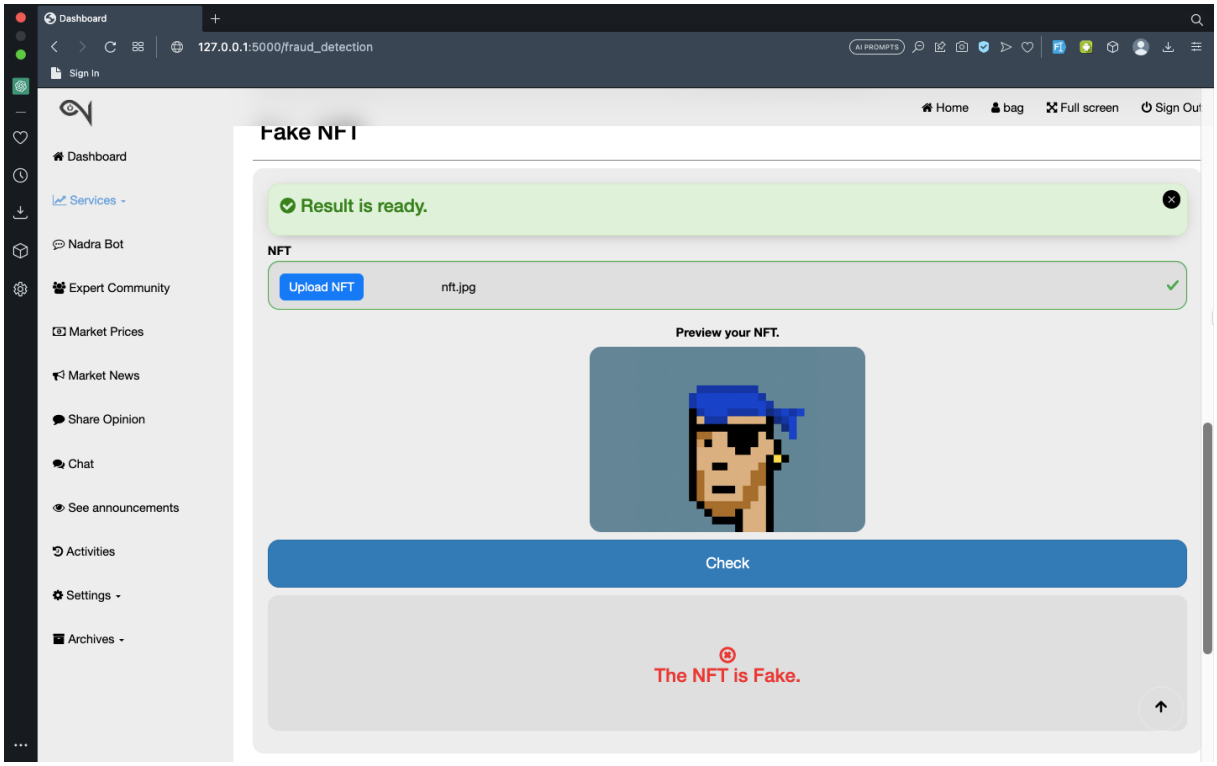


Figure 6.11: Nadra Fake NFT Detection Service

6.1.5 Community of Experts

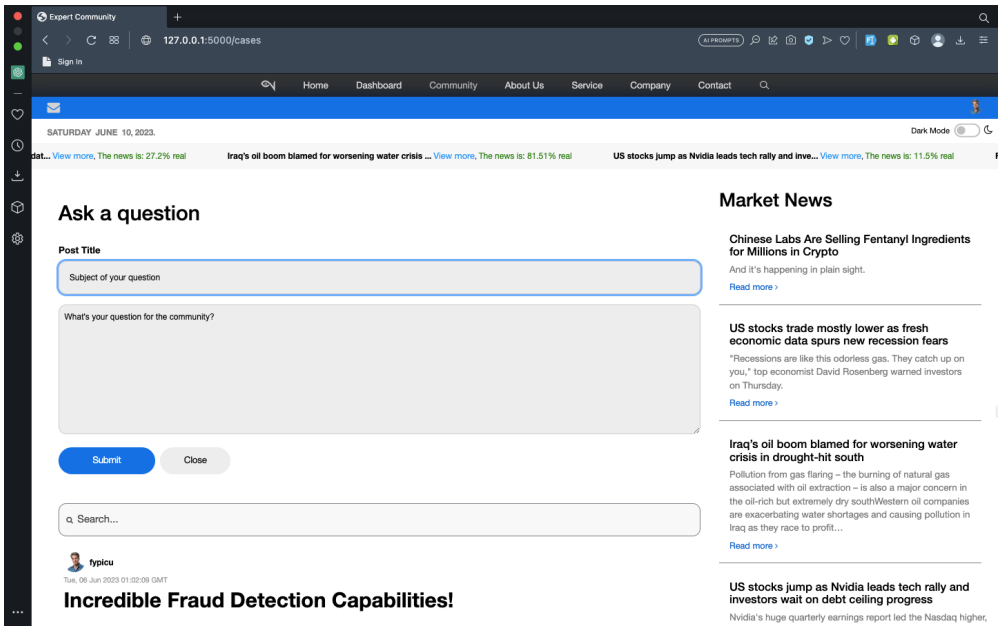


Figure 6.12: Community of Nadra

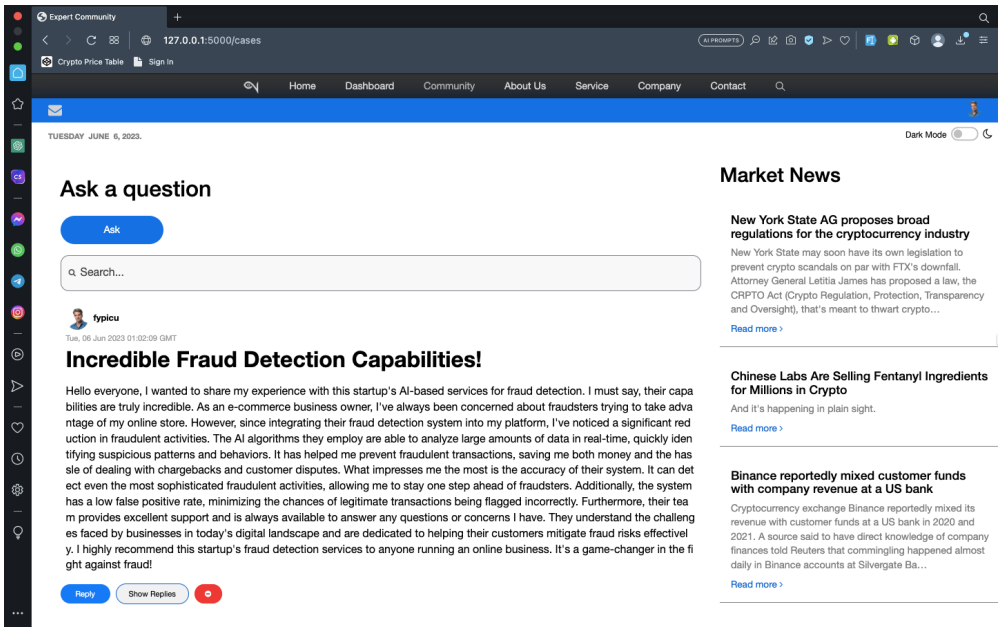


Figure 6.13: Community of Nadra

6.1.6 Nadra Bot

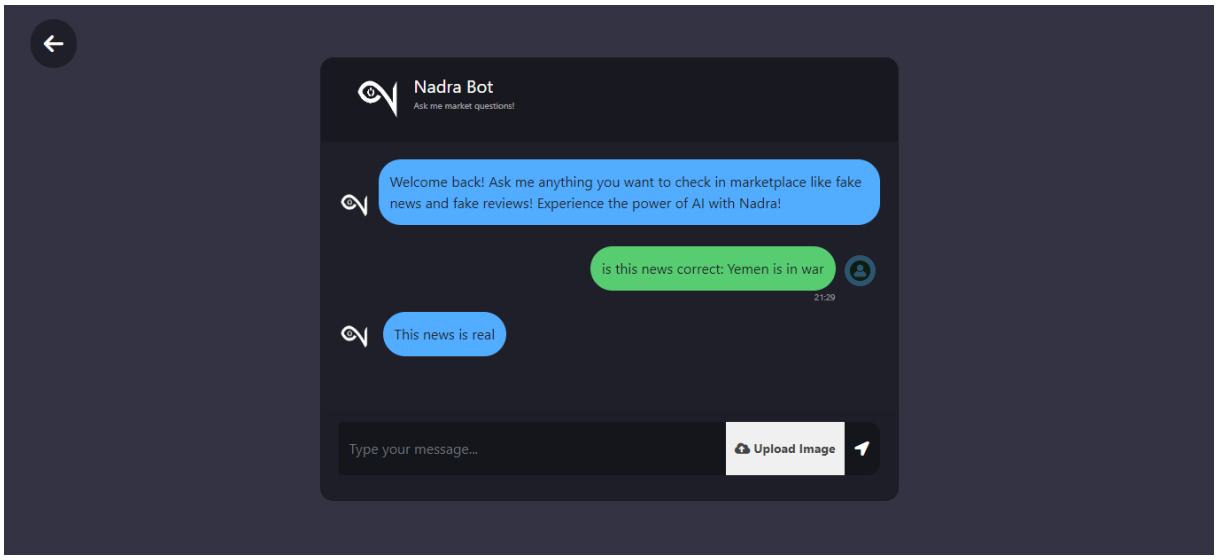


Figure 6.14: Nadra Bot

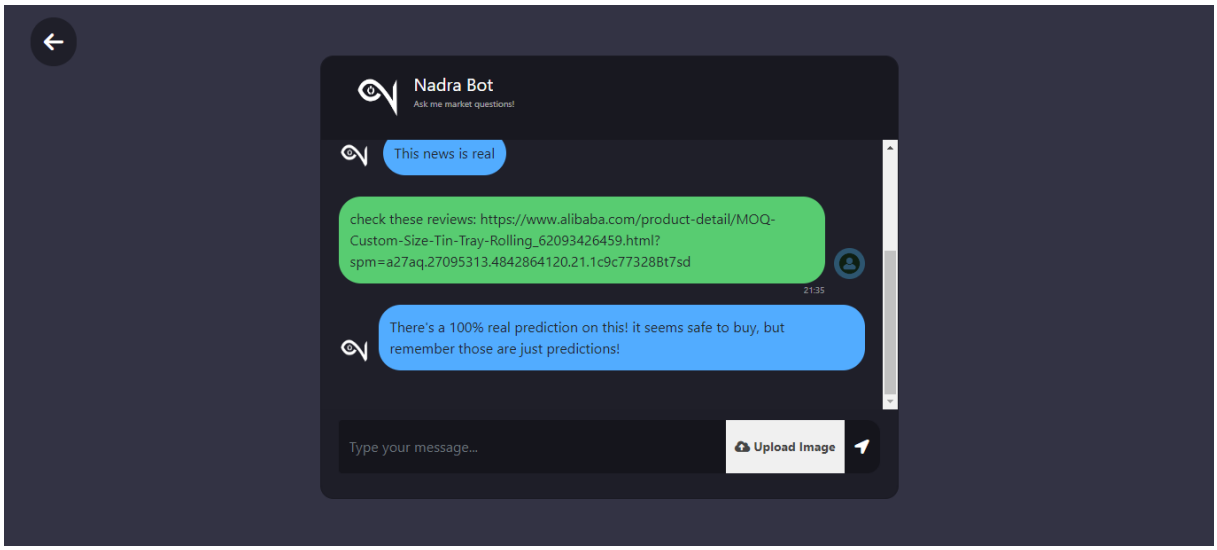


Figure 6.15: Nadra Bot

6.2 Mobile Application

6.2.1 Home Page

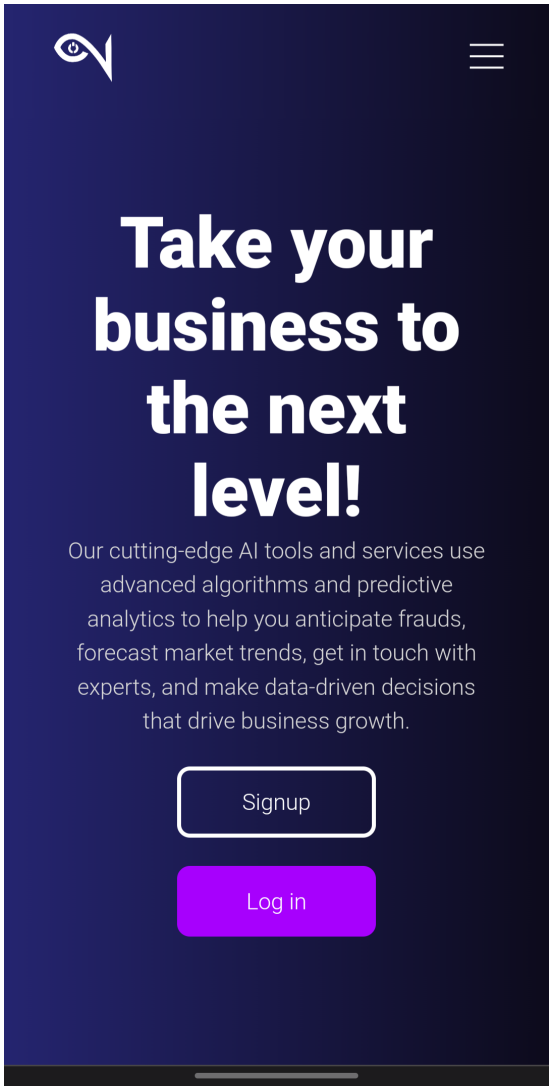


Figure 6.16: Home Page

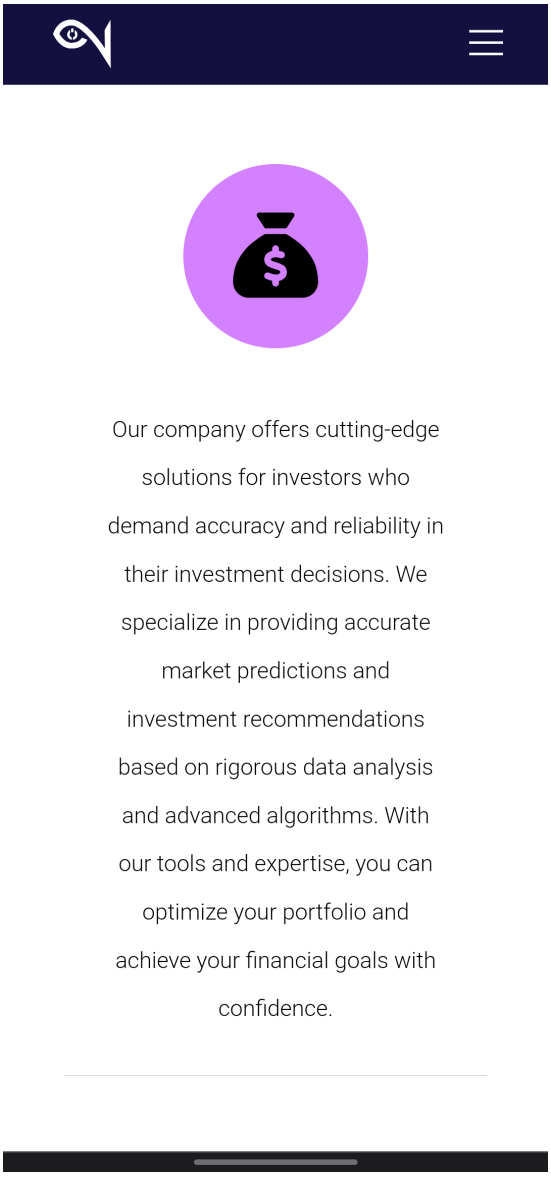


Figure 6.17: Features Page

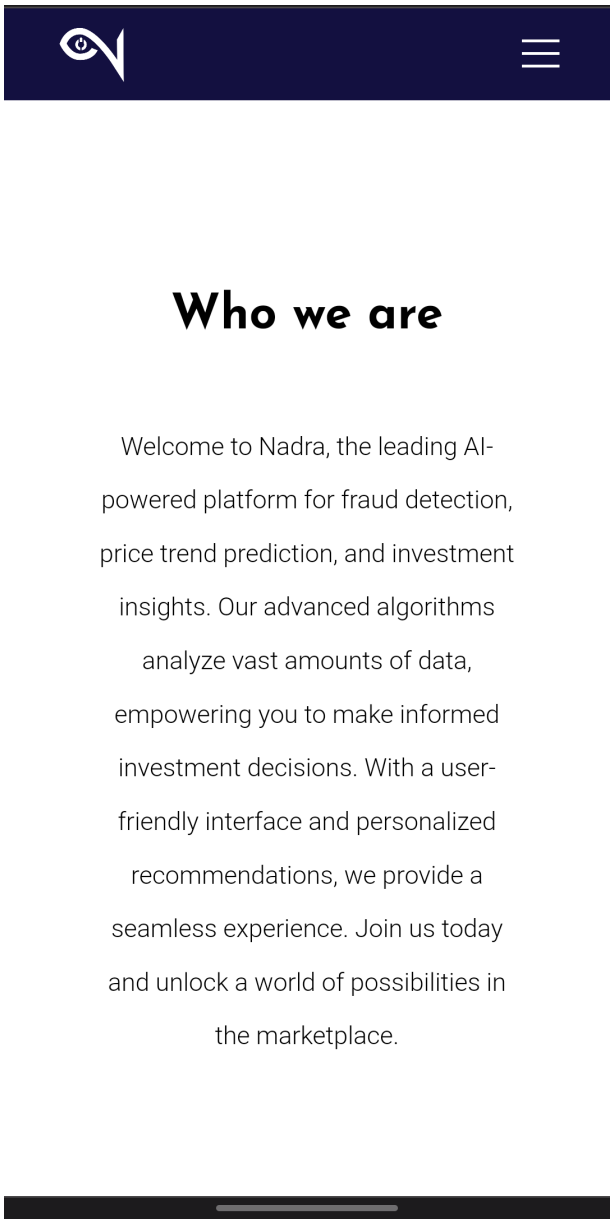


Figure 6.18: About us

Who we are

Welcome to Nadra, the leading AI-powered platform for fraud detection, price trend prediction, and investment insights. Our advanced algorithms analyze vast amounts of data, empowering you to make informed investment decisions. With a user-friendly interface and personalized recommendations, we provide a seamless experience. Join us today and unlock a world of possibilities in the marketplace.

Figure 6.19: About us Page

recommendations, we provide a seamless experience. Join us today and unlock a world of possibilities in the marketplace.

Contact Us

Please feel free to get in touch with us using the contact information below:

Address
El Nasar neighbourhood, Saida, Algeria

Email
nadra.contact@nadra.com

Phone
+213445689542

Figure 6.20: Contact us phone

6.2.2 Sign Up Page

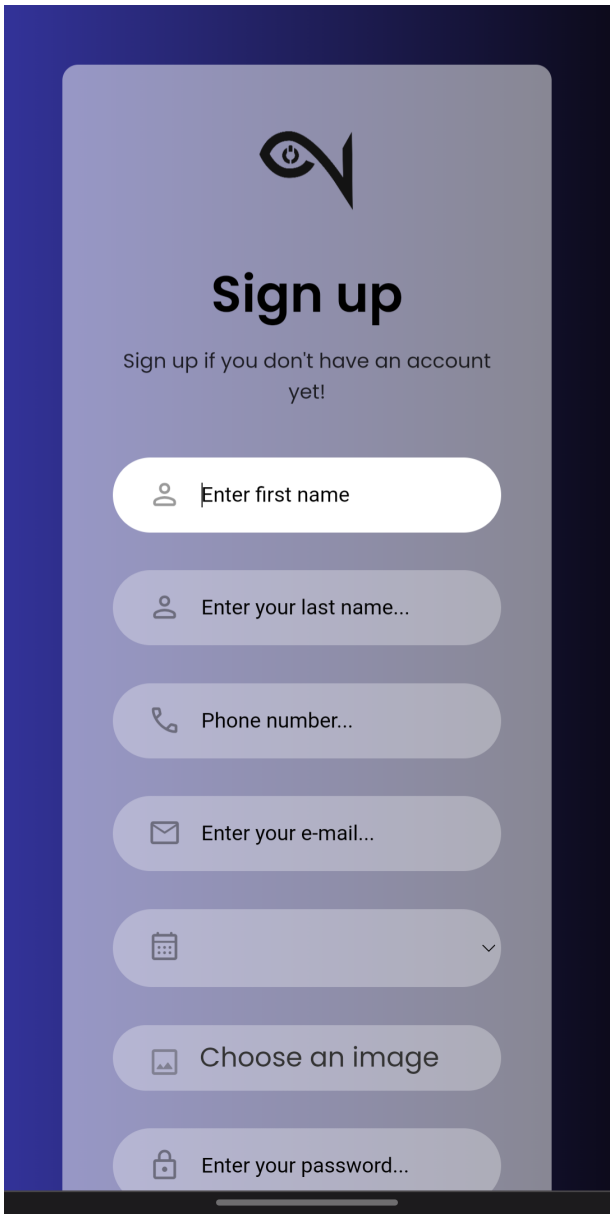


Figure 6.21: Signup page

6.2.3 Sign In Page

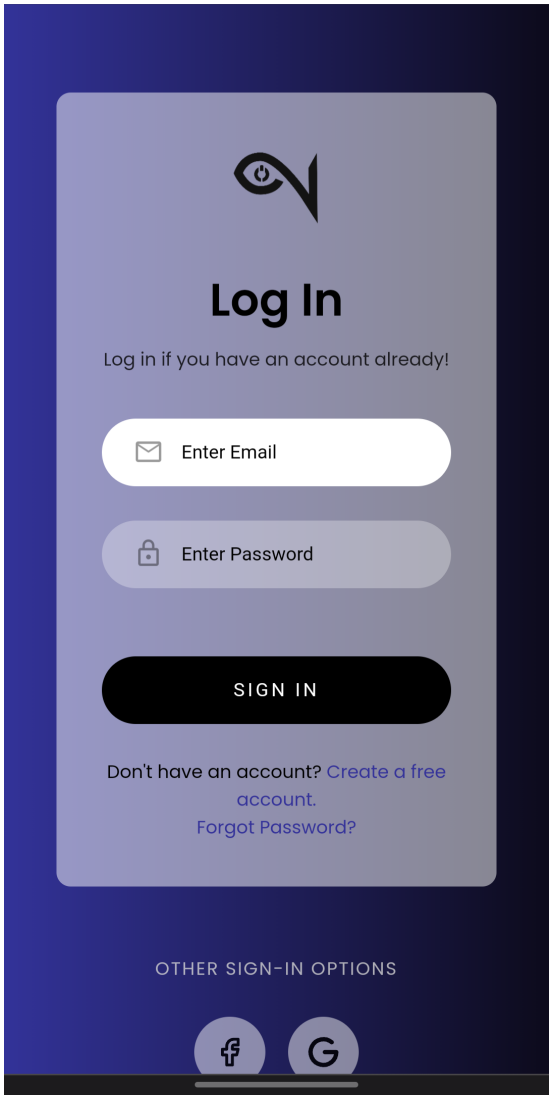


Figure 6.22: Sign In

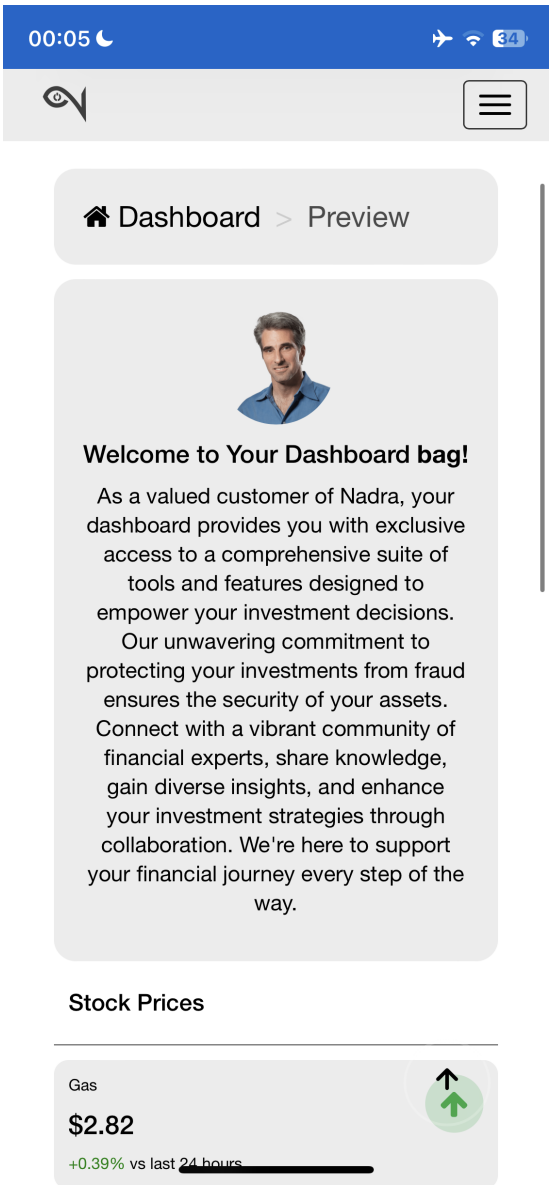


Figure 6.23: Dashboard

6.2.4 Services Page

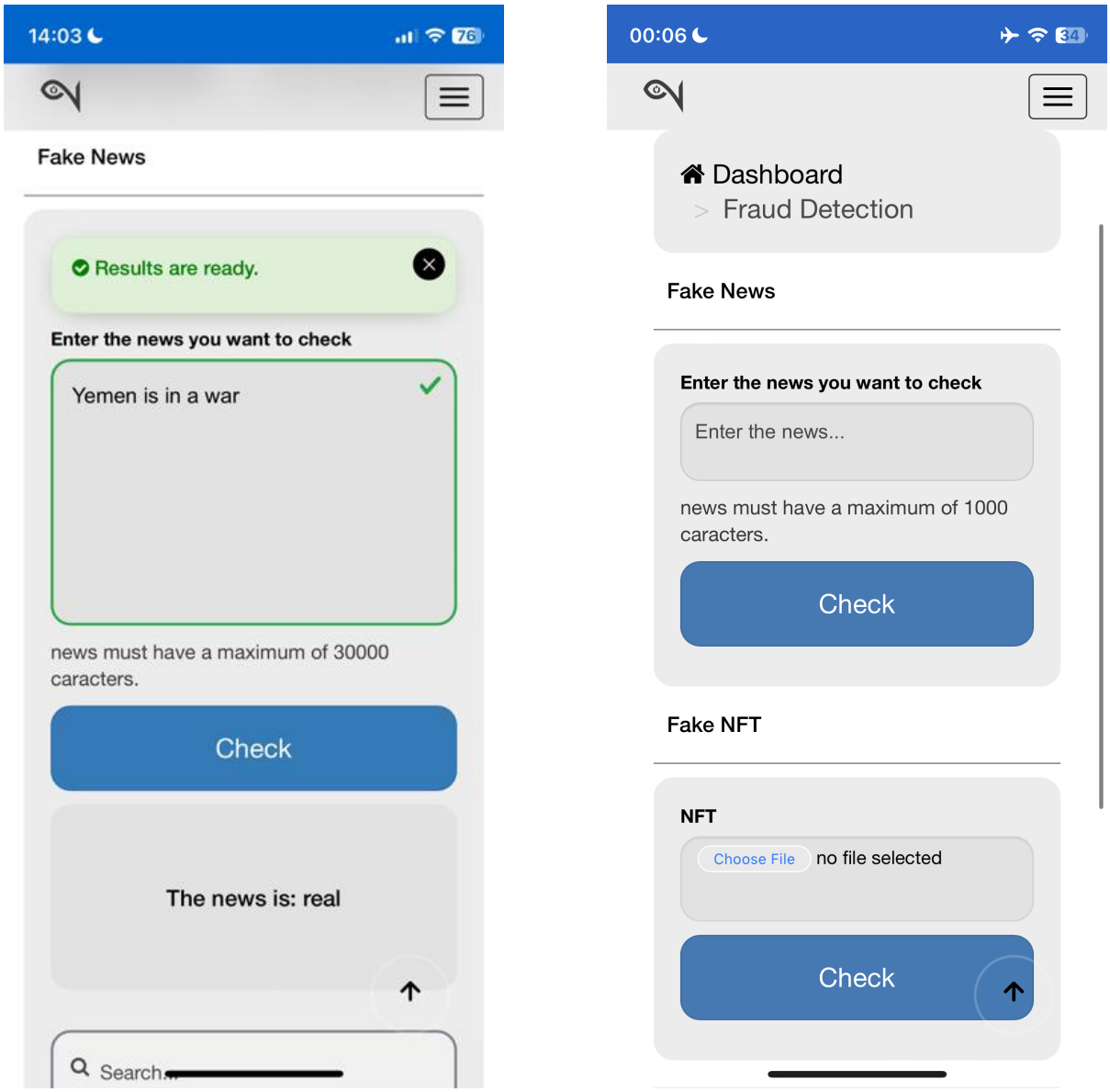


Figure 6.24: Fraud Detection Page

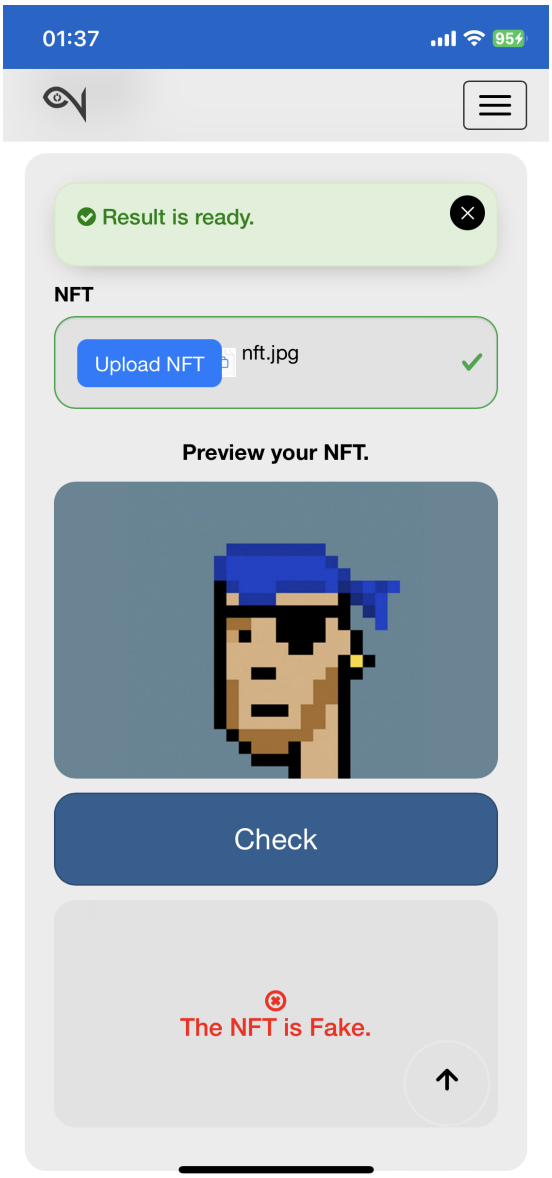


Figure 6.25: Nadra Fake NFT Detection

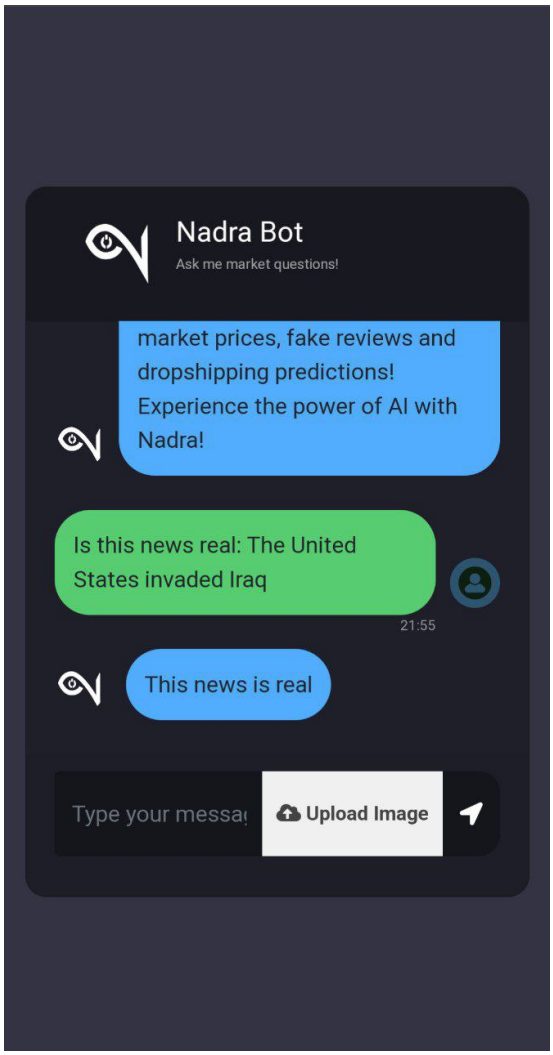


Figure 6.26: Nadra Bot

Chapter 7

General Conclusion

This thesis focuses on the issue of market frauds, with a specific focus on investigating the extent of fake news, fabricated reviews, counterfeit gold, and fraudulent non-fungible tokens (NFTs). However, despite our efforts, we were unable to fulfill our supervisor's objectives, which included detecting various other forms of fraud such as malicious smart contracts testing algorithms within a blockchain setting, identifying dropshipping fraud, uncovering e-commerce scammers, and more. The primary focus of the research revolves around investigating the effectiveness of Nadra Embedding Convolutional network and NadraGPT-2 model in detecting fake news.

Apart from tackling fake news, the study also incorporates the detection of fake reviews by utilizing cutting-edge models such as NadraGPT-2 and NadraBERT. These models are employed to analyze and assess the reviews found on e-commerce websites, with a comparative analysis conducted to evaluate the strengths and weaknesses of each model in terms of sentiment classification.

Additionally, the thesis addresses the problem of counterfeit gold detection using FCNN, which has yielded promising results thus far.

Furthermore, by creating Nadra recommendation system that is capable of self-learning, investors can be assisted by various recommendations in investing sector.

To combat the issue of fake non-fungible tokens (NFTs), a collaboration with MICR students has been established. In this collaboration, a DCGAN model is employed to generate synthetic NFT data, which is then used to train our detection models. The ultimate goal of this partnership is to enhance the detection capabilities and effectively combat fraudulent activities within the NFT space.

The experiments and analyses conducted have substantiated the efficacy of the proposed method in detecting market frauds. Nevertheless, there is room for enhancements and future prospects in this research.

7.1 Challenges and limitations

There were various obstacles and constraints faced during this research. The main difficulty revolved around time limitations, as creating and training detection models, as well as evaluating their performance, demanded substantial computational resources and time. Additionally, the availability and quality of data presented challenges, as obtaining accurate and diverse datasets proved to be crucial but quite challenging.

7.2 Future work

This thesis not only contributes to the understanding of market frauds but also paves the way for further research and improvements in this field. Building upon the findings and achievements of this study, there are several promising directions that can be explored in future work:

1. Broadening the scope of counterfeit gold detection to various used gold in Algeria such as necklaces, earrings and rings.

2. Detecting Arabic fake news will be our next move.

3. Recognizing the crucial role of data in enhancing the performance of detection models, it is imperative to focus on acquiring larger, more comprehensive, and reliable datasets. The availability of such data can greatly contribute to improving the accuracy and effectiveness of fraud detection algorithms.

4. The next step involves refining and experimenting with different models. While the current study has achieved notable results, exploring alternative algorithms and techniques can lead to further advancements in fraud detection. Continual refinement and optimization of the models can help enhance their efficiency, precision, and ability to adapt to evolving fraudulent strategies.

By pursuing these directions, future research in the field of market frauds can continue to make significant contributions, deepen our understanding of fraudulent activities, and ultimately foster the development of more robust and effective detection methods.

Bibliography

[1] Eleni Adamopoulou and Lefteris Moussiades. An overview of chatbot technology. In *Artificial Intelligence Applications and Innovations: 16th IFIP WG 12.5 International Conference, AIAI 2020, Neos Marmaras, Greece, June 5–7, 2020, Proceedings, Part II 16*, pages 373–383. Springer, 2020.

[2] Ahmed Ali Mohammed Al-Saffar, Hai Tao, and Mohammed Ahmed Talab. Review of deep convolution neural network in image classification. In *2017 International conference on radar, antenna, microwave, electronics, and telecommunications (ICRAMET)*, pages 26–31. IEEE, 2017.

[3] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pages 1–6. Ieee, 2017.

[4] Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, 8:1–74, 2021.

[5] Muhammad Rasyid Redha Ansori, Revin Naufal Alief, Ikechi Saviour Igboanusi, Jae Min Lee, Dong-Seong Kim, et al. Watermarking-based fake audio nft detection in nft marketplace. , pages 487–488, 2023.

[6] Aliaksandr Barushka and Petr Hajek. Review spam detection using word embeddings and deep neural networks. In *Artificial Intelligence Applications and Innovations: 15th IFIP WG 12.5 International Conference, AIAI 2019, Hersonissos, Crete, Greece, May 24–26, 2019, Proceedings 15*, pages 340–350. Springer, 2019.

[7] Yekta Said Can, Fatih Alagöz, Emre Özer, and Mücahit Gündebahar. Counterfeit gold identification using sound and image processing. In *2015 23nd Signal Processing and Communications Applications Conference (SIU)*, pages 1074–1077. IEEE, 2015.

[8] Richard Davis and Chris Proctor. Fake news, real consequences: Recruiting neural networks for the fight against fake news, 2017.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[10] Vijaypal Singh Dhaka, Sangeeta Vaibhav Meena, Geeta Rani, Deepak Sinwar, Muhammad Fazal Ijaz, and Marcin Woźniak. A survey of deep convolutional neural networks applied for prediction of plant leaf diseases. *Sensors*, 21(14):4749, 2021.

[11] Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

[12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[14] Petr Hajek, Aliaksandr Barushka, and Michal Munk. Fake consumer review detection using deep neural networks integrating word embeddings and emotion mining. *Neural Computing and Applications*, 32:17259–17274, 2020.

[15] K Jeevitha, A Iyswariya, V RamKumar, S Mahaboob Basha, and V Praveen Kumar. A review on various segmentation techniques in image processsing. *European Journal of Molecular & Clinical Medicine*, 7(4):1342–1348, 2020.

[16] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[17] Chayakrit Krittanawong, HongJu Zhang, Zhen Wang, Mehmet Aydar, and Takeshi Kitai. Artificial intelligence in precision cardiovascular medicine. *Journal of the American College of Cardiology*, 69(21):2657–2664, 2017.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[19] Anders Krogh. What are artificial neural networks? *Nature Biotechnology*, 26:195–197, 2008.

[20] Luyang Li, Bing Qin, Wenjing Ren, and Ting Liu. Document representation and feature combination for deceptive spam review detection. *Neurocomputing*, 254:33–41, 2017.

[21] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[22] Larry R Medsker and LC Jain. Recurrent neural networks. *Design and Applications*, 5:64–67, 2001.

[23] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

[24] Dominika Podstawka-Piechnik, Bartosz J Spisak, and Paweł Głuszcz. Identification of gold-plated items by x-ray fluorescence spectrometry. *Microchemical Journal*, 156:104827, 2020.

[25] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[26] Eva Rodriguez, Beatriz Otero, Norma Gutiérrez, and Ramon Canal. A survey of deep learning techniques for cybersecurity in mobile networks. *IEEE Communications Surveys Tutorials*, PP:2, 06 2021.

[27] Jitendra Kumar Rout, Anmol Dalmia, Kim-Kwang Raymond Choo, Sambit Bakshi, and Sanjay Kumar Jena. Revisiting semi-supervised learning for online deceptive review detection. *IEEE access*, 5:1319–1327, 2017.

[28] Csaba Szepesvári. Algorithms for reinforcement learning. *Synthesis lectures on artificial intelligence and machine learning*, 4(1):1–103, 2010.

[29] Juan Terven and Diana Cordova-Esparza. A comprehensive review of yolo: From yolov1 to yolov8 and beyond. *arXiv preprint arXiv:2304.00501*, 2023.

[30] James Thorne, Mingjie Chen, Giorgos Myrianthous, Jiashu Pu, Xiaoxuan Wang, and Andreas Vlachos. Fake news detection using stacked ensemble of classifiers. Association for Computational Linguistics, 2017.

[31] Aswini Thota, Priyanka Tilak, Simrat Ahluwalia, and Nibrat Lohia. Fake news detection: a deep learning approach. *SMU Data Science Review*, 1(3):10, 2018.

[32] Pavan Vadapalli. Ultimate Guide to Object Detection Using Deep Learning [2023]. *upGrad blog*, 11 2022.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[34] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, Eftychios Protopapadakis, et al. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.

[35] Lichun Zhou and Qian Zhang. Recognition of false comments in e-commerce based on deep learning confidence network algorithm. *Information Systems and e-Business Management*, pages 1–18, 2021.