SACAIR 2023

Book of Abstracts

Presentations at the Unconference of the Fourth Southern African Conference for Artificial Intelligence Research

Editors:

Tevin Moodley
Sibonelo Dlamini
Asad Jeewa
Yuvika Singh

December 2023

SACAIR
Southern African Conference for Artificial Intelligence Research

Preface

Message from the Unconference Chairs

Dear authors and readers,

It is with great pleasure that we write this foreword to the Book of Abstracts of the Unconference of the Fourth Southern African Conference for Artificial Intelligence Research (SACAIR2023). The unconference, held in-person on 4th December 2023. It preceded the main conference held from the 5th to the 8th of December. This is a student-driven event that provides a platform for developing researchers and young academics to present their research and interact with peers, sponsors and potential employers. It also includes a hackathon, hosted by Entelect this year, as well as a shared task, released to the public in the months preceding the conference.

This document contains the extended abstracts that were accepted after a light review process. Each paper was reviewed by two reviewers and a meta-reviewer decided the final outcome, resulting in 10 accepted extended abstracts. The purpose of the process is to introduce young researchers to the formal review process and provide them with valuable insight that will improve their work. The authors formally presented their work at the unconference. The accepted abstracts cover a wide scope within artificial intelligence, covering both applied and theoretical machine learning in subfields of deep learning, reinforcement learning and symbolic AI.

We would like to express our sincere gratitude to all those individuals who have helped us in organising this event. In particular, the conference committee, for their unwavering support as well as Entelect and the Goethe Institut who we have partnered with this year. We are pleased with the level of interaction from students and are look forward to witnessing their development into established researchers within our country.

December 2023

Tevin Moodley
Sibonelo Dlamini
Asad Jeewa
Yuvika SIngh

Message from the General Chairs

The Southern African Conference for Artificial Intelligence Research was established four years ago with multiple objectives. One of the most important of these was for the conference to serve a developmental agenda. We intend that the conference will be instrumental in developing the research careers of postgraduate students and emerging academics. For many students, this conference will be their first exposure to an international conference.

The unconference, a fairly informal gathering of students, academics and industry, is intended to provide a friendly environment with an emphasis on interaction. It provides a forum for students to share work in progress and to elicit feedback from seasoned researchers and peers. We are pleased with the interest shown by emerging academics and students and congratulate the authors whose work appears in this Book of Abstracts. It is heartening to note the excellent quality of the work presented here.

We also take this opportunity to congratulate the young academics who chaired the unconference and thank them for their hard work in organising the this event.

December 2023

Terence van Zyl
Dustin van der Haar

Our Sponsors

NITheCS
National Institute for
Theoretical and Computational Sciences

Artificial Intelligence
www.elsevier.com/locate/artint

cair
CENTRE FOR ARTIFICIAL
INTELLIGENCE RESEARCH

NRF National Research Foundation

entelect
everything is possible

UNIVERSITY OF JOHANNESBURG

GOETHE INSTITUT

MaSS
DSI-NRF CENTRE OF EXCELLENCE IN MATHEMATICAL & STATISTICAL SCIENCES

Table of Contents

Explanation for Defeasible Entailment

Victoria Chama[1][0000−0003−3641−4849]

University of Cape Town Private Bag X3 Rondebosch 7700, Cape Town
chmvic006@myuct.ac.za

Abstract. Explainable AI is a highly relevant topic of current research. It aims to come up with intelligent systems that are able to provide reasons for decisions made and actions taken. Explanations for monotonic systems are provided through justifications. However, little research has gone into providing explanations for non-monotonic systems where additional information may lead to the withdrawal of earlier consequences. We demonstrate that in order to derive justifications for defeasible knowledge bases accurately, we need to establish the point at which conflicts arise by using the rational closure algorithm to come up with a ranking of defeasible statements. This is of great importance because it has been shown that the semantics is appropriate in the sense that the representation and derived algorithm to compute rational entailment checking links to the set of preferential and rational properties of the KLM approach. This means that the algorithm to derive justifications from defeasible knowledge bases is also appropriate since it conforms to the intuitive semantics for defeasible subsumption in description logics.

Keywords: Defeasible entailment · Rational closure · Justification-based explanations.

1 Explanation for Entailments

The argument for extending classical reasoning services with explanations has been made clear through uses such as: Explanations support users in detecting and repairing errors in potentially large and highly complex knowledge bases [1, 2], knowledge base comprehension can be made easier and explanations allow the user to pinpoint the statements which are responsible for inconsistencies.

Additionally, explanation has been shown to be crucial for user acceptance and satisfaction in many studies [3]. There has been a significant amount of research devoted to the area of explanation. In particular, a specific type of explanation called justifications [2]. However, users still have trouble understanding justifications. Thus, another research area has emerged that uses justifications as a foundation for explanations but expresses them in natural languages such as English for easy readability. It is important to mention that justifications are not the only way to express explanations but rather they are the most commonly used because they appear to be closer to the way people do reasoning, more concise and readable than reasoning-based explanations [4]

2 Defeasible Explanation Components

The algorithm to compute justifications for defeasible knowledge bases is therefore based on two main bodies of work, namely that of classical justification generation, and that of non-monotonic reasoning in description logics in particular the computation of rational closure as shown in Figure 1.

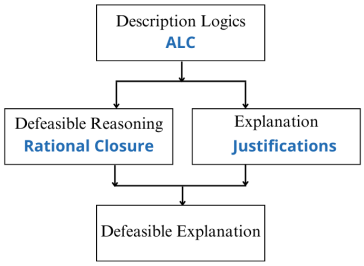


Fig. 1. Defeasible Explanation components

3 Computation of Defeasible Justifications Algorithm

By combining the rational closure algorithm to handle defeasible knowledge bases and the algorithms to compute justifications we provide a framework to compute explanations for defeasible reasoning

Algorithm : ComputeDefeasibleJustifications($\mathcal{K}, \eta$)

Input: $\mathcal{K} = \mathcal{T} \cup \mathcal{D}$, the corresponding $\mathcal{K}^* = \mathcal{T}^* \cup \mathcal{D}^*$, the sequence $\mathcal{E}_0, \ldots, \mathcal{E}_n$, the ranking $R = \mathcal{D}_0, \ldots, \mathcal{D}_n$, and a query $\eta = C \sqsubseteq D$.

Output: Justification $\mathcal{J}$

```
i ← 0
J_result ← ∅
rank ← RationalClosureForJustifications(K, R, η)
if rank = 0 then
    J_result ← ComputeAllJustifications(K, η)
    return J_result
while i < rank do
    K_new ← K_new \ D_i
    i ← i + 1
J_result ← ComputeAllJustifications(K_new, η)
return J_result
```

Fig. 2. Algorithm to Compute Defeasible Justifications

The algorithm to compute justifications for defeasible knowledge bases for an entailment η is summarised as follows: it uses RationalClosure and ComputeAllJustifications

as sub-procedures. The first step is to compute whether or not the query is rationally entailed from the knowledge base and return the rank at which the rational closure was determined. The rank is central in determining justifications for defeasible knowledge bases because it indicates which layers have been discarded.

Suppose we have a knowledge base that has been ranked using the rational closure algorithm.

2	$\mathsf{Bird} \mathrel{\tilde{\sqsubset}} \mathsf{Fly}$, $\mathsf{Bird} \mathrel{\tilde{\sqsubset}} \mathsf{Wings}$
1	$\mathsf{Penguin} \mathrel{\tilde{\sqsubset}} \neg\mathsf{Fly}$
0	$\mathsf{SpecialPenguin} \mathrel{\tilde{\sqsubset}} \mathsf{Fly}$
∞	$\mathsf{Penguin} \sqsubset \mathsf{Bird}, \mathsf{Robin} \sqsubset \mathsf{Bird}, \mathsf{SpecialPenguin} \sqsubset \mathsf{Penguin}$

Fig. 3. The Ranked Interpretation of K

When we query the ranked knowledge base and ask $\mathsf{SpecialPenguin} \sqsubset \mathsf{Fly}$, "do special penguins typically fly". The system responds "YES". There are 2 minimal subsets that support the statement special penguins typically fly

$\mathcal{J}_1 = \mathsf{SpecialPenguin} \mathrel{\tilde{\sqsubset}} \mathsf{Fly}$

$\mathcal{J}_2 = \mathsf{SpecialPenguin} \sqsubset \mathsf{Penguin}, \mathsf{Penguin} \sqsubset \mathsf{Bird}, \mathsf{Bird} \mathrel{\tilde{\sqsubset}} \mathsf{Fly}$ Additionally, you could give the argument that another subset $\mathcal{J}_3 = \mathsf{SpecialPenguin} \sqsubset \mathsf{Penguin}, \mathsf{Penguin} \mathrel{\tilde{\sqsubset}} \neg\mathsf{Fly}$ exists which will return "NO" to the query. However only $\mathcal{J}_1$ is the only valid answer.

4 Conclusion

In order to accurately derive justifications for situations where a knowledge base may contain conflicting information, we need to establish the point at which conflicts arise i.e., determine the rank.

This is of great importance because compared to other frameworks such as the one by Casini and Straccia [6] which lacks an appropriate semantics, the syntax presented for defeasible subsumption has an appropriate semantics that is intuitive. Furthermore, the derived algorithm to compute rational entailment can be reduced to classical entailment checking meaning its computational complexity is no worse than that of entailment checking in the classical underlying description logic. There are four distinct paths for future work available for justification-based explanations for defeasible reasoning: An implementation and testing of the ComputeDefeasibleJustifications algorithm, possibility of extending justifications to the other forms of defeasible reasoning within the KLM approach, a theoretical understanding of explanations for other non-monotonic reasoning frameworks and extend justification-based explanations to natural language explanations to assist users in understanding entailments.

References

1. Bail, S. The justificatory structure of OWL ontologies. (University of Manchester,2013)
2. Horridge, M. Justification based explanation in ontologies. (University of Manchester,2011)
3. Biran, O. & Cotton, C. Explanation and justification in machine learning: A survey. *IJCAI-17 Workshop On Explainable AI (XAI)*. pp. 8 (2017)
4. Nguyen, T. Generating natural language explanations for entailments in ontologies. (The Open University,2013)
5. Britz, K., Meyer, T., Casini, G., Moodley, K., Sattler, U. & Varzinczak, I. Theortical Foundation of Defeasible Description Logics. (2018)
6. Casini, G. & Straccia, U. Defeasible Inheritance-Based Description Logics. *JAIR*. 48 pp. 415-473 (2013)

Towards General Learning Agents for Markov Decision Processes

Eben du Plessis[1], Pieter de Villiers[2], Hans Grobler[3], Johan Viljoen[4]

[1] Hensoldt SA, Irene, South Africa
[2] University of Pretoria, Pretoria, South Africa
[3] University of Pretoria, Pretoria, South Africa
[4] Hensoldt SA, Irene, South Africa
eben.duplessis@hensoldt.net

Abstract. The goal of this work is to contribute towards the design and understanding of agents in the context of general learning. We aim to show that progress can be made, and that positive steps can be taken in the direction of general learning, irrespective of whether it can be achieved in absolute terms. The agents considered in this work for general intelligence will be referred to as Generic Problem-Solving Agents. A description of such an agent, the environment for such an agent, how to evaluate these agents, and a limited instantiation of such an agent, called *GPSA_basic*, is described in this paper. Four of the key problems regarding general learning are discussed. The novel approach in which the *GPSA_basic* algorithm solves these problems is demonstrated by evaluating the algorithm on some familiar problem domains represented as Markov Decision Processes. *GPSA_basic* paves the way for a single artificial problem solver to operate in many diverse environments, doing so through self-learning.

Keywords: Markov Decision Processes, General Game Playing, General Learning, Deep Reinforcement Learning, Generic Problem-Solving Agents

1 Introduction

While most existing artificial intelligence (AI) projects aim at a certain aspect or application of intelligence, a general intelligence (GI) project aims at intelligence as a whole, which has many aspects, and can be used in various situations. A major obstacle to creating agents that can function in such a generalized learning context is that the problem domain is unknown, thus requiring the evolutionary development of the agent rather than pre-constructing it. Two contemporary methods having a similar objective is neuroevolution [1-4], and, arguably the most state-of-the-art method instantiating AI today, namely deep reinforcement learning (DRL) [5]. The *GPSA_basic* algorithm is presented as a potential alternative to current implementations of DRL. Four key problems for GI are introduced and it is shown that *GPSA_basic* provides a possible solution to them. In the experiments *GPSA_basic* was compared with a general implementation of DRL. The results are discussed.

2 What a GPSA Is, and Four Key Problems of GI

In this section four key problems of general problem solving are discussed as well as the implications they hold for DRL, and most other methods depending on artificial neural networks (ANNs). This is followed by a description of a typical environment for a generic problem-solving agent (GPSA), concluded by a description of what a GPSA is.

2.1 Four Key Problems of GI and Their Impact on DRL

The first problem pertains to the ability of an agent to function within an unknown problem space. This problem is about the construction and efficiency of an agent without any prior knowledge of the problem domain. All this knowledge must be acquired as the discovery process unfolds [6]. Such a problem space will be a multi-task environment [6]. This in turn gives rise to fractured problems because of the discontinuous relation between problem states and possible actions [7]. Related to the sequence in which problems are encountered is interleaving between problems [8, 9].

The second problem is about the integration of newly attained information into an existing knowledge base without losing existing information captured, thus avoiding "forgetting" or "unlearning" [10].

The third problem is the speed of acquiring new knowledge from single observations containing limited information. Modern artificial methods that evolve or develop their solutions such as deep learning and neuroevolution [11-13] can take a considerable number of time steps to recognise and integrate new information.

The last problem is the processing resources required for DRL. Current implementations of DRL such as AlphaZero requires significant computational resources and expensive dedicated hardware [14].

2.2 Typical Problem Domain for, and Description of a GPSA

From the four key problems a nine-point foundational description of the problem domain for a GPSA was constructed for evaluation. Briefly, these points are: (1) the number of problem domains is unknown, (2) the nature of the problem domains is unknown, (3) problem domains may be isolated, interleaved or blended, (4) problem domains are governed by underlying rules which may drastically change dynamically, (5) problem states may be highly repeatable or not, (6) there can be deterministic as well as random transitions between states, (7) problem states may differ subtlety yet have drastically different outcomes, (8) reward or punishment may be hidden at any depth, and (9) event chains where actions have no effect on the immediate outcome. The criteria proposed for deciding between the performances of different GPSAs can be stated for the purposes of this work as: *The GPSA that can function best inside this 9-point domain, while doing so using least processing resources and power.*

An agent qualifying as a GPSA will at least require the following characteristics: generic input, the ability to internalise inputs, the ability to discover the rules that govern these entities, the ability to find complex rules and a set of generic effector functions.

Lastly, the *GPSA_basic* algorithm is an instantiation of a GPSA that is not based on an artificial neural network, but rather consists out of the following real-time sub-algorithms: I-net, TSMF, Optimisation, Change Detect, Stagnation Detect and Stitching.

3 Findings

GPSA_basic and a general implementation of DRL were compared on four well-known, but specially adapted problems namely CartPole, Tic-Tac-Toe, Collision Avoidance for Multiple Mobile Robot Environment, and finally a problem space consisting of a combination of all the above. The combination problem represents an instantiation of the discussed GPSA problem domain. In this section the results of the experiments are discussed to show that *GPSA_basic* can indeed solve the four key problems better than DRL and can function in a GPSA problem domain.

Key Problem 1 - Functioning Within an Unknown Problem Space: From the experiments it was found that significant tuning of parameters such as learning rate, replay memory size and topology was required for DRL to produce acceptable results. Clearly, a form of solution evolution is required. *GPSA_basic* starts off *tabula rasa*, then internalises the problem in such a way that it eliminates the problem of upfront design and secondly reduces learning time. The algorithm evolves its output interface to suit the environment. *GPSA_basic* does not require upfront design of topology or replay memory.

Key Problem 2 - Integration of Newly Attained Information: From the experiments it was seen that *GPSA_basic* exhibits a level of insensitivity to the scale of new problems, irrespective of the order in which they were learned. *GPSA_basic* disposes of the cumbersome need to first insert the new input into experience replay memory and then slowly starting to integrate this new information into the existing weights of the trained ANN. It was seen that *GPSA_basic* could learn a new ruleset in just a few minutes, while this was not the case for DRL. *GPSA_basic* outperformed DRL in the experiments.

Key Problem 3 - The Acquisition Speed of New Knowledge: *GPSA_basic* accomplishes one-shot learning. As soon as a change in the environment is detected, or a new problem is introduced, the algorithm will quickly detect the change and respond to it immediately by learning it. Since DRL relies on a potentially very large replay memory, from which batch training is done on the ANN, the algorithm took longer and longer as this memory grew to integrate new information, or to discover a change in the environment.

Key Problem 4 - Required Processing Resources: In the conducted experiments *GPSA_basic* used less than 25% of the time, and less than 1% of the memory required by DRL for the same problems. Both methods executed on standard personal computers (PCs).

References

1. Floreano, D., Dürr, P., Mattiussi, C.: Neuroevolution: from architectures to learning. Evolutionary Intelligence, 47-62 (2008)
2. Lehman, J., Miikkulainen, R.: Neuroevolution. Scholarpedia, 30977 (2013)
3. Levine, J., Bates Congdon, C., Ebner, M., Kendall, G., Lucas, S.M., Miikkulainen, R., Schaul,T., Thompson, T.: General Video Game Playing. Artificial and computational intelligence in games, 77-84 (2013)
4. Schrum, J., Miikkulainen, R.: Evolving multimodal networks for multitask games. IEEE Transactions on Computational Intelligence and AI in Games, 94-111 (2012)
5. Mousavi, S.S., Schukat, M., Howley, E.: Deep reinforcement learning: An overview. In: Proceedings of SAI Intelligent Systems Conference, pp. 426 – 440 (2016)
6. Lange, S., Riedmiller, M., Voigtlander, A.: Autonomous reinforcement learning on raw visual input data in a real world application. Neural Networks (IJCNN), In: Conf. The 2012 International Joint Conference on, pp. 1-8. (2012)
7. Kohl, N., Miikkulainen, R.: Evolving neural networks for strategic decision-making problems. Neural Networks, 326-337 (2009)
8. J. Schrum, private communication, Jun 2015
9. Schrum, J., Miikkulainen, R.: Solving multiple isolated, interleaved, and blended tasks through modular neuroevolution. Evolutionary Computation, 459-490. (2016)
10. D'Silva, Janik, T., R., Chrien, M., Stanley, K.O., Miikkulainen, R.: Retaining learnt behaviour during real-time neuroevolution. In: Conf. Artificial Intelligence and Interactive Digital Entertainment, pp. 39-44. (2005)
11. Risi, S., Togelius, J.: Neuroevolution in games: State of the art and open challenges. IEEE Transactions on Computational Intelligence and AI in Games, 25-41 (2017)
12. Eiben, A.E., Smith, J.E.: Introduction to evolutionary computing. Springer, Switzerland (2007)
13. Bartunov, S.A., Botvinick, S., Wierstra, M., Lillicrap, D.: Meta-learning with memory-augmented neural networks. In: Conf. International Conference on Machine Learning, pp.1842-1850. (2016)
14. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A.: Mastering the game of go without human knowledge. Nature, 354-359 (2017)

A comparative analysis of segmentation techniques for malignant tumour detection

Celumusa Duma and Yüvika Singh

School of Mathematics, Statistics and Computer Science,
University of KwaZulu Natal, Westville, Durban, 3629, SA

Abstract. This study aimed to compare three segmentation techniques—U-Net, Random Forest, and Fuzzy C-Means—for malignant tumour detection in medical images. We used a publicly available 2D brain tumour segmentation dataset comprising 3064 images from 233 patients. The experimental setup consisted of an encoder-decoder U-Net architecture with a Random Forest classifier and Gabor features for feature extraction. We also proposed a modified Fuzzy C-Means algorithm for the purpose of brain tissue density segmentation. The results indicate that the U-Net model produced the most accurate results with an accuracy of 99.7% and dice coefficient of 92.9% on training data, followed by Random Forest with an accuracy of 98.9%. The study highlights the strengths, weaknesses, and potential areas for improvement of each segmentation technique.

Keywords: Medical image segmentation, Tumour detection, Fuzzy-c means, Convolutional Neural Network, Random Forest, U-Net.

1 Introduction

A tumour is a mass of dysfunctional mutated cells that spread throughout the body, causing pain, and invading organs [1]. Unlike benign tumours, malignant tumours pose a high risk due to the presence of cancerous cells and their metastasizing nature. In 2020, an estimated 9.96 million lives were claimed worldwide by these tumours, which stand as a prominent cause of death [2].

Segmentation techniques are vital for the spatial location of tumours and for converting related images into readable formats. Medical imaging techniques and processing, such as magnetic resonance imaging (MRI) and computed tomography (CT), have revolutionized cancer diagnosis and treatment planning. However, accurate segmentation of tumour regions from medical images is a challenging task that requires sophisticated algorithms and techniques.

When a patient is sick and medical professionals suspect an underlying health issue, various diagnostic scans may be required to assess the patient's condition and determine the appropriate treatment. The choice of scan depends on the patient's symptoms, medical history, and the suspected or potential diagnoses. After obtaining an image, a medical professional will meticulously examine it in an effort to identify any underlying diseases and potential contributing factors. This process can be quite labour-intensive and time-consuming, varying from several hours to a few days, contingent upon factors like the gravity of the pathology and the level of expertise of the clinician involved [3].
Several segmentation techniques have been proposed in the literature, but their effectiveness and suitability for automated systems are still unclear. Some of the techniques proposed are thresholding, region-based, edge-detection, clustering, and deep learning among others. Thresholding is a common method for segmenting images into regions of interest. In thresholding, an image is divided into two groups of pixels based on their intensity values. Pixels with intensity values below a specified threshold are assigned to one group, while pixels with intensity values above or equal to the threshold are assigned to another group [4]. Pandey et al [5] proposed a method for the diagnosis of COVID-19 using deep learning and image segmentation. They conducted a performance comparison of different image segmentation techniques, which encompassed simple thresholding at 0.3 and 0.6, multiple thresholding within the range of 26 to 230, and the application of Otsu's method. They found that Otsu's method produced the best results with precision of 95.04% and specificity of 69.65% with an accuracy of 95.52% for COVID-19 CT scan images. Ng et al. [6] proposed a novel methodology for medical image segmentation that combines k-means clustering and an improved watershed segmentation algorithm. The methodology was evaluated on a dataset of 50 2-D MR images of the head, and it was shown to be effective in generating general segmentation maps. The authors concluded that their proposed approach of using k-means clustering to obtain a primary segmentation of MR images prior to applying the improved watershed segmentation algorithm is a promising one. More recently, Guo et al. [7] designed and

implemented an image segmentation system based on deep Convolutional Neural Networks (CNN). CNN is a popular deep learning-based segmentation technique that has achieved state-of-the-art results in several medical imaging applications, including tumour detection. Given all of this, a need for a comparative analysis of segmentation techniques for malignant tumour detection arises. In this study, we conducted a comparative analysis of segmentation techniques for malignant tumour detection for automated systems.

2 Related Work

Segmentation of malignant tumours from medical images is a crucial task in computer-aided diagnosis. It is used to extract useful anatomical structures, locate tumours, abnormalities, lesions, quantify tumour growth, and conduct radiation dose calculations for treatments. Various segmentation techniques have been proposed and applied to different medical imaging modalities. Among them, convolutional neural network (CNN) segmentation, fuzzy c-means (FCM) segmentation, and random forest (RF) segmentation have shown promising results.

CNN is a popular deep learning-based segmentation technique that has achieved state-of-the-art results in several medical imaging applications, including tumour detection [7]. It was first designed in 1980 to classify manuscript numerals. This was later translated into recognition techniques, patterns and shapes from unprocessed images. CNN-based segmentation techniques have been developed for different medical imaging modalities, such as computed tomography (CT) and magnetic resonance imaging (MRI) [7]. The tech7nique's ability to learn discriminative features from images has led to improved segmentation accuracy and reduced the need for manual intervention.

Multiple efforts have been placed in creating CNN-based segmentation techniques that generate a segmentation map in one forward pass. One such study proposed a computational method called deconvolution to visualise images [8]. This method enhances the image intensity by finding the inverse of the mathematical expression designed to distinguish the image from the artefacts. Long et al [9], further proposed using deconvolution to restore the dimensions of segmentation maps. This method initiates neural networks to cypher bilinear interpolation. Badrinarayanan et al [10] proposed an alternative approach that requires keeping activation indices, storing them in max-pooling layers and later upsampling them. A similar method was proposed by Noh et al [11]. Other studies join conditional random fields (CRN) with traditional CNN techniques. One such work uses pixels in image segmentation via CNN and then post-processes these results using CRF [12]. Ciseran et al [13] used convolution of neural networks to segment electron microscopy images. This method uses classifiers to extract pixels in each layer of a moving window and consequently repeats the process for each stack. Ronneberger et al [14], used the same method to acquire better results. The size of the latter images was smaller, but the features were bigger and clearer when compared to Ciseran et.al.

In a study to evaluate CNN-based segmentation of MRI images of the hand and the brain, it was concluded that combining segmentation maps at different scales produces better results when compared to a summation of forward features in sequential stages [15]. This was due to challenges of class imbalances and high memory demand. Another CNN-based technique is superpixel. This technique separates images into areas of similar characteristics such as colour and texture [16].

A study comparing a combination of the superpixels and SegNet in carcass images showed that SegNet produces 97%-pixel accuracy due to convolution and upsampling layers [16]. Although this technique exhibits good background information, it shows poor precision in the edges. This study was done in a pool of 226 carcass images. A study to design a segmentation technique for MRI brain images proposed a 14-staged, 9-layer convolutional neural network approach [17]. In this approach, input images are converted into similar dimensions, convoluted, batch normalised then max-pooled for accurate segmentation. When tested, the technique produced 20.83% accuracy using the convolution of SVM on CNN 98.10% when Adamax was used as an optimizer, and SOFTMAX used the initial parameter [17]. The final results obtained were 99.745% accurate when RMSProp was used as an optimizer [17].

FCM segmentation is a clustering-based segmentation technique that groups image pixels based on their intensity values and spatial proximity. FCM has been applied to various medical imaging modalities and has shown promising results in tumour segmentation [18]. However, FCM's performance may be sensitive to the initialisation of its parameters, and it may require extensive manual intervention to achieve accurate segmentation results. FCM-based techniques with spatial constraints are useful for half-volume segmentation of MRI images and inhomogeneity correction of intensity. To compensate for the shortcomings of these techniques, Ahmed et al [19] proposed a corrected FCM called BCFCM. BCFCM computes each localised term in a single iteration step and hence the process becomes

time-consuming [19]. This can be eliminated by spatial constrained kernelised FCM where localised terms are grouped into functions [19].

RF segmentation is a machine learning-based segmentation technique that uses an ensemble of decision trees to classify image pixels [20]. RF has been applied to different medical imaging modalities, including MRI and CT. RF has shown promising results in tumour segmentation and can handle high-dimensional data with a large number of features. However, RF's performance may be sensitive to the selection of its hyperparameters, and it may require extensive computational resources.

In a study to compare CNN segmentation and RF-type techniques, a deep convolutional neural network (DCCN), Random Forest, feature extraction architecture and whole image architecture were used [21]. Using 80% training and 20% testing data, DCCN showed the most accurate results and sensitivity while requiring less memory. This was linked to the ability of neural networks to recognize accurate features even in constantly changing data.

Edge-based segmentation techniques use edges as distinct image boundaries. These boundaries are detected, measured as amplitudes, compared to the threshold and accepted or declined based on the crack-edge position [7]. This technique works best when combined with region-based algorithms to avoid discrepancies related to noise or weak or fake edge detection. Region-based segmentation clusters pixels with similar properties together.

In conclusion, CNN segmentation, FCM segmentation, and RF segmentation are promising techniques for malignant tumour detection in medical images. However, their suitability for automated systems may vary, depending on their sensitivity to hyperparameter selection and their need for manual intervention. A comparative analysis of these techniques can provide insights into their strengths and weaknesses and identify areas for improvement.

3 Materials and Methods

Brain tumour segmentation is a challenging task due to the high variability in the appearance of tumours, as well as the presence of noise and artefacts in medical images. In recent years, deep learning methods, such as convolutional neural networks (CNNs), have shown promising results for brain tumour segmentation. However, traditional machine learning methods, such as random forest (RF) and fuzzy c-means (FCM), are still widely used in clinical practice. In this study, we embark on a comprehensive comparative analysis of these three methods, each of which offers unique advantages and trade-offs in the context of brain tumour segmentation.

3.1 Data acquisition and preprocessing

Medical image analysis, particularly in the context of brain tumours segmentation, is a crucial area of research with significant clinical applications. Accurate and efficient segmentation of brain tumours from MRI can aid in the diagnosis, treatment planning, and patient monitoring. To facilitate advancements in this field, we use a publicly available, comprehensive 2D brain tumour segmentation dataset [22]. The dataset comprises 3064 T1-weighted contrast-enhanced images from 233 patients, encompassing three distinct types of brain tumors: glioma (1426 images), meningioma (708 images), and pituitary tumors (930 images), along with their corresponding ground truth masks.

4 References

[1] Tandon, S., George, S.M., McIntyre, R. and Kandasubramanian, B., 2020. Polymeric immunosensors for tumor detection. Biomedical Physics & Engineering Express, 6(3), p.032001.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Ferlay J, Ervik M, Lam F, Colombet M, Mery L, Piñeros M, et al. 2020. Global Cancer Observatory: Cancer Today. Lyon: International Agency for Research on Cancer. (https://gco.iarc.fr/today, accessed May 2023).

[3] Ummadi, V., 2022. U-Net and its variants for Medical Image Segmentation: A short review. arXiv preprint arXiv:2204.08470.

[4] Jardim, S., António, J. and Mora, C., 2023. Image thresholding approaches for medical image segmentation-short literature review. Procedia Computer Science, 219, pp.1485-1492.

[5] Pandey, S.K., Bhandari, A.K. and Singh, H., 2022. A transfer learning based deep learning model to diagnose covid-19 CT scan images. Health and Technology, 12(4), pp.845-866.

[6] Ng, H.P., Ong, S.H., Foong, K.W.C., Goh, P.S. and Nowinski, W.L., 2006, March. Medical image segmentation using k-means clustering and improved watershed algorithm. In 2006 IEEE southwest symposium on image analysis and interpretation (pp. 61-65). IEEE.

[7] Guo, Z., Li, X., Huang, H., Guo, N. and Li, Q., 2019. Deep learning-based image segmentation on multimodal medical imaging. IEEE Transactions on Radiation and Plasma Medical Sciences, 3(2), pp.162-169.

[8] Zeiler, M.D. and Fergus, R., 2014. Visualizing and understanding convolutional networks. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13 (pp. 818-833). Springer International Publishing.

[9] Long, J., Shelhamer, E. and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).

[10] Badrinarayanan, V., Handa, A. and Cipolla, R., 2015. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. arXiv preprint arXiv:1505.07293.

[11] Noh, H., Hong, S. and Han, B., 2015. Learning deconvolution network for semantic segmentation. In Proceedings of the IEEE international conference on computer vision (pp. 1520-1528).

[12] Kamnitsas, K., Ledig, C., Newcombe, V.F., Simpson, J.P., Kane, A.D., Menon, D.K., Rueckert, D. and Glocker, B., 2017. Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. Medical image analysis, 36, pp.61-78.

[13] Ciresan, D., Giusti, A., Gambardella, L. and Schmidhuber, J., 2012. Deep neural networks segment neuronal membranes in electron microscopy images. Advances in neural information processing systems, 25.

[14] Ronneberger, O., Fischer, P. and Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18 (pp. 234-241). Springer International Publishing.

[15] Kayalibay, B., Jensen, G. and van der Smagt, P., 2017. CNN-based segmentation of medical imaging data. arXiv preprint arXiv:1701.03056.

[16] Gonçalves, D.N., de Moares Weber, V.A., Pistori, J.G.B., da Costa Gomes, R., de Araujo, A.V., Pereira, M.F., Gonçalves, W.N. and Pistori, H., 2021. Carcass image segmentation using CNN-based methods. Information Processing in Agriculture, 8(4), pp.560-572.

[17] Chattopadhyay, A. and Maitra, M., 2022. MRI-based brain tumour image detection using CNN based deep learning method. Neuroscience informatics, 2(4), p.100060.

[18] Shen, S., Sandham, W., Granat, M. and Sterr, A., 2005. MRI fuzzy segmentation of brain tissue using neighborhood attraction with neural-network optimization. IEEE transactions on information technology in biomedicine, 9(3), pp.459-467.

[19] MEVIS, F., 2014. Improved bias-corrected Fuzzy C-Means segmentation of brain MRI data. International Journal of Hybrid Information Technology, 7(3), pp.65-84.

[20] Breiman, L., 2001. Random forests. Machine learning, 45, pp.5-32.

[21] Hartmann, D., Müller, D., Soto-Rey, I. and Kramer, F., 2021. Assessing the role of random forests in medical image segmentation. arXiv preprint arXiv:2103.16492.

Exploring Explanations for KLM-Style Defeasible Reasoning

Chipo Hamayobe[0009−0008−0987−814X] ⋆ ⋆⋆

University of Cape Town
chipo@cs.uct.ac.za

1 Introduction

Knowledge representation and reasoning constitute a domain within the realm of Artificial Intelligence. It involves the modelling of information through formal logical frameworks, enabling the application of rule-based manipulations that pertain to specific modes of reasoning [1]. Within the diverse array of logical systems, varying levels of expressive power emerge. Classical logic, for instance, lacks the capacity to encapsulate the concept of *typicality*, posing a challenge when endeavouring to succinctly represent exceptional knowledge. This typicality encompasses a notion of ambiguity, enabling us to revise our inferences should we acquire information that contradicts them. Such a style of reasoning is known as *defeasible reasoning*. We deal with a specific method for establishing defeasible reasoning which was proposed by Kraus et al. recognised as the *KLM approach* [9], which employs *propositional logic* as its foundation. We focus on *rational closure*, the most conservative.

A vital element of the process of reasoning involves the ability to deduce fresh insights from preexisting information. Beyond merely deriving new knowledge, it proves highly advantageous to understand the rationale behind the derivation of specific information. Within formal logic, this understanding is facilitated through *explanations* [4]. Explanations hold significant importance as they are intrinsic to reasoning services that enhance the comprehension of knowledge bases. As a result, they assume a pivotal role in the practical utilisation of reasoning systems. In logical reasoning services, the primary means of offering explanations is via *justifications*, and this paper explores this aspect.

2 Entailment

A collection of finite propositional formulas is termed a *knowledge base*, $\mathcal{K} \subseteq \mathcal{L}$. An interpretation becomes a model of a knowledge base $\mathcal{K}$ if it functions as a model for every formula within $\mathcal{K}$, that is, for every model of $\mathcal{K}$, a formula $\alpha \in \mathcal{L}$ holds `true`. We assert that $\mathcal{K}$ *entails* a statement α, denoted as $\mathcal{K} \models \alpha$, when every model of $\mathcal{K}$ also serves as a model for α, that is $Mod(\mathcal{K}) \subseteq Mod(\alpha)$ [8].

⋆ Supervised by Professor Tommie Meyer (tmeyer@airu.org.za)
⋆⋆ Supported by the Artificial Intelligence Research Unit (www.airu.org.za)

Definition 1. *Let $\mathcal{F} \subseteq \mathcal{L}$ be a set of formulas and α a formula. $\mathcal{F}$ entails α, denoted $\mathcal{F} \models \alpha$, if and only if every model of $\mathcal{F}$ is a model of α [12].*

The *KLM Framework* expands propositional logic by incorporating the notion of *defeasible implication*, $\mid\sim$, the counterpart to the classical implication $\rightarrow$. Defeasible implications are represented as $\alpha \mid\sim \beta$, where $\alpha, \beta \in \mathcal{L}$, interpreted as *"$\alpha$ typically implies β"* [10]. *Defeasible entailment*, $\mid\!\approx$, is then established as a binary relationship that spans defeasible knowledge bases and defeasible implications. For instance, $\mathcal{K} \mid\!\approx \alpha \mid\sim \beta$ signifies that within the scope of $\mathcal{K}$, *"$\mathcal{K}$ defeasibly entails that α typically implies β"* [5].

The `BaseRank` procedure arranges all statements based on their level of generality. All classical statements are positioned at the lowest rank, $\mathcal{R}_\infty$, and materialised defeasible statements are placed progressively in generality as one ascends the ranking order towards the topmost, $\mathcal{R}_0$ [3] [6]. Rational Closure represents the most cautious approach to deriving conclusions from a defeasible knowledge base $\mathcal{K}$. This form of entailment draws limited inferences from the given $\mathcal{K}$ [3].

3 Explanations

Within reasoning systems, *explanations* serve the purpose of clarifying the relevant statements within the knowledge base $\mathcal{K}$ that contribute to the logical implication between the said $\mathcal{K}$ and an entailed statement, the *query* [4]. Among the various methods for explanations in classical logics, *justifications* have been extensively studied. A justification, $\mathcal{J} \subseteq \mathcal{K}$, for the logical implication of a formula β consists of a minimal subset of $\mathcal{K}$ that results in the implication of β. The set of all justifications when $\mathcal{K} \models \beta$ is symbolised as $\mathcal{J}\ (\mathcal{K}, \beta)$.

Definition 2. *Let $\mathcal{K}$ be a knowledge base and β be a query such that $\mathcal{K} \models \beta$. The set of formulas $\mathcal{J}$ is a justification for $\mathcal{K} \models \beta$ if $\mathcal{J} \subseteq \mathcal{K}$, $\mathcal{J} \models \beta$ and for all $\mathcal{J}^{'} \subseteq \mathcal{J}$, it holds that $\mathcal{J}^{'} \not\models \beta$ [12].*

Horridge introduced the concept of justifications and also introduced an algorithm to identify justifications for classical entailments [7]. However, when dealing with defeasible entailments, the concept of justification becomes more intricate. Based on work by Brewka and Ulbricht [2], Chama [4] provided a distinct definition for defeasible justification, which Everett et al. [5] expanded on. As emphasised by Wang [12], Chama's definition of justification is applicable to the explanations we are exploring in this paper. Chama [4] modified the `RationalClosure` algorithm to include an extra integer indicating the count of formula ranks that are ignored and Wang [12] made further changes to produce the `BaseRankForJustification` and `RationalClosureForJustification` algorithms.

Example 1. Given the defeasible knowledge base:

$$\mathcal{K} = \{animal \mid\sim wild, animal \mid\sim hunt, pet \rightarrow animal, simba \rightarrow pet\}$$

The `BaseRankForJustification` algorithm produces the formula ordering $\mathcal{R}_0 = \{animal \mid\sim wild, animal \mid\sim hunt\}$ and $\mathcal{R}_\infty = \{pet \rightarrow animal, simba \rightarrow pet\}$ respectively. From this ranking, the `RationalClosureForJustification` algorithm determines that $\mathcal{K} \mid\approx simba \mid\sim wild$ and does not discard any ranks. We might initially consider both $\mathcal{J}_1 = \{animal \mid\sim wild, pet \mid\sim animal, simba \mid\sim pet\}$ and $\mathcal{J}_2 = \{animal \mid\sim wild, animal \mid\sim hunt, pet \mid\sim animal, simba \mid\sim pet\}$ as explanations supporting the claim. However, $\mathcal{J}_2$ contains $animal \mid\sim hunt$, which was not considered by the `KLMDefeasibleJustification` algorithm. Consequently, only $\mathcal{J}_1$ remains the valid defeasible explanation since it only includes the suitable and relevant minimal formulas for this entailment.

4 The KLMDEETool

We transformed the defeasible entailment and explanations algorithms into a working software system tool. The `KLMDEETool` is developed in Java and is structured around a well-defined architecture that adheres to the *Multi-tier Architecture* pattern of software engineering [11]. It consists of several interconnected components that work together to facilitate the processing and analysis of defeasible reasoning scenarios. The architecture is designed to ensure efficiency, modularity, and ease of use. The source code is publicly available on *GitHub*[1].

At its core, the system encompasses the *UI Manager* responsible for providing a unified user interface. This *user interface* tier offers both a *command line interface (CLI)* and a *graphical user interface (GUI)* for user interaction. Both of these interfaces consume and provide the same functionality but only differ in presentation and the verbosity of the debug and information output. The *UI Manager* interacts with various services such as the *entailment service*, *justification service*, *explanation service* and the *knowledgebase service*, responsible for storing and managing the defeasible knowledge base related data.

5 Conclusions and Future Work

The challenges in comprehending conclusions drawn from defeasible reasoning are highlighted, and the role of justifications in explaining defeasible entailments and aiding user understanding is discussed. The paper addresses this need by presenting algorithms developed by Chama [4] and extended by Wang [12] that compute justifications for defeasible entailments.

The primary contribution of this paper is a software system tool implementing the proposed defeasible entailment and justification algorithms which can be used as a debugging service for knowledge bases. The `KLMDEETool` can further be extended to other defeasible reasoning formalisms such as Lexicographic Closure and Relevant Closure. The logical operations can also be modified and tested with bigger knowledge bases and complex queries. *Natural Language Processing (NLP)* techniques and tools could also be employed to provide more robust explanations, perhaps in a user's chosen language.

[1] The source code is available at https://github.com/chiefmonk/dee

References

1. Brachman, R.J., Levesque, H.J.: Actions. In: Knowledge Representation and Reasoning. pp. 285–303. Elsevier (2004)
2. Brewka, G., Ulbricht, M.: Strong explanations for nonmonotonic reasoning. Description Logic, Theory Combination, and All That: Essays Dedicated to Franz Baader on the Occasion of His 60th Birthday pp. 135–146 (2019)
3. Casini, G., Meyer, T., Varzinczak, I.: Taking defeasible entailment beyond rational closure. In: Logics in Artificial Intelligence, pp. 182–197. Lecture Notes in Computer Science, Springer International Publishing, Cham (2019)
4. Chama, V.: Explanation for defeasible entailment. Master's thesis, Faculty of Science, University of Cape Town, Rondebosch, Cape Town, 7700 (2020)
5. Everett, L., Morris, E., Meyer, T.: Explanation for klm-style defeasible reasoning. In: Southern African Conference for Artificial Intelligence Research. pp. 192–207. Springer, South Africa (2021)
6. Freund, M.: Preferential reasoning in the perspective of poole default logic. Artificial Intelligence **98**(1), 209–235 (1998). https://doi.org/https://doi.org/10.1016/S0004-3702(97)00053-2
7. Horridge, M.: Justification based explanation in ontologies. Ph.D. thesis, University of Manchester, UK (2011)
8. Kaliski, A.: An Overview of KLM-Style Defeasible Entailment. Master's thesis, Faculty of Science, University of Cape Town, Rondebosch, Cape Town, 7700 (2020)
9. Kraus, S., Lehmann, D., Magidor, M.: Nonmonotonic reasoning, preferential models and cumulative logics. Artificial Intelligence **44**(1), 167–207 (1990). https://doi.org/https://doi.org/10.1016/0004-3702(90)90101-5
10. Lehmann, D., Magidor, M.: What does a conditional knowledge base entail? Artificial Intelligence **55**(1), 1–60 (1992). https://doi.org/https://doi.org/10.1016/0004-3702(92)90041-U
11. Walter, S., Kenrick, M.: Absolute Java. Pearson Education, Boston :, sixth edition edn. (2016)
12. Wang, S.: Defeasible Justification for the KLM Framework. Master's thesis, Faculty of Science, University of Cape Town, Rondebosch, Cape Town, 7700 (2022)

Simulated Annealing Using Quantum Inspired Algorithms *

Eleanor Kedem[1] **

University of Cape Town, Cape Town, South Africa
eleanor.kedem@gmail.com

Abstract. In recent years quantum annealers were developed. These are based on quantum annealing algorithms, which evolve a multi quantum bit quantum state using an operator which evolves slow enough over time, so as to allow the state to continuously remain in its lowest energy state, with its final lowest energy state representing the optima. The quantum annealers had great promise and also proved to achieve considerable speed-up in certain cases. However, despite their potential, quantum annealers are still restricted by today's technology. Quantum Inspired Annealing is an algorithm which runs on a classical computer and simulates the dynamical evolution of a physical system, using inspiration from quantum mechanics. This work aims at comparing the classical simulated annealing with three different quantum inspired simulated annealing algorithms to solve the traveling salesman problem.

Keywords: Quantum Inspired Algorithm · Quantum Annealing · TSP.

1 Introduction

Combinatorial optimisation is a class of optimisation problems that has applications in nearly all areas of society. Such problems involve searching for an optimal solution in a finite range of potential candidates, and optimisation is standardly achieved by approximating the optimal solution. These problems are considered difficult to solve. One combinatorial optimisation problem is the Travelling Salesman Problem (TSP), which aims at finding the shortest cyclic path given a set of cities and their respective distance. In its optimisation form, TSP is an NP-hard problem [1].

One optimisation method commonly used to solve TSP is simulated annealing (SA). SA is a strong metaheuristic, in which the system is 'heated' and then 'cooled' down slowly in a controlled manner, in search of optima [2–5]. Simulated annealing has an extensive research and many variations, several of which are quantum inspired variations [6–13]. Those are optimisation algorithms, running on classical computers, borrowing ideas from quantum mechanics for better

* Supported by The South African Quantum Technology Initiative
** Supervised by Prof. Tommie Meyer, Prof. Francesco Petruccione and Prof. Ilya Sinayskiy

performance. It is based on the notion that quantum mechanic offers many advantages to classic one [14–16], but it is not bound to industrial developments in the field of quantum computing. In its more general form, the quantum inspired annealing relies on the adiabatic theorem for calculations [17–19], meaning that a quantum system will stay near its instantaneous state of lowest energy (also referred to as ground state) if the operator that governs its evolution varies slowly enough, allowing it to reach the optima embedded in the last ground state. Despite the fact that the adiabatic theorem and the annealing process are seemingly distinct independent processes, this slow controlled evolution of the system makes these two algorithms closely related [20–22]. Due to the relative simplicity of building a quantum annealer, extensive research was directed towards analysing the abilities of such computers (in particular D-Wave [23, 24]) and borrowing inspiration from its behaviour for development of quantum approximation optimisation algorithms [25–27].

With the growing research on quantum computing in general and adiabatic quantum computing in particular [28–36], and its new applications [37–41], this research benchmarks four different algorithms for TSP - simulated and quantum inspired annealing, in order to further explore the applicability and suitability of specific algorithms to specific problems.

2 Methods

Four different algorithms were chosen for this comparison:

Simulated Annealing (SA) - The used algorithm is a variation of the classic simulated annealing, which iteratively explores a solution space by accepting worse solutions with decreasing probability, gradually 'cooling' the system to find near-optimal solutions [3].

Vanilla Quantum Inspired Annealing (QIA) - This algorithm adds qunatum tunneling to the SA, allowing a particle to tunnel through a potential barrier to a non-adjacent state. This helps escaping local minima in the search space and exploring alternative solutions [6, 42, 43].

Quantum Inspired Annealing Using Gradient Descent (GD-QIA) - This algorithm translates TSP to an Ising model, and aims at finding the configuration of spins that minimises the energy of the system. The dynamical evolution is replaced with a gradient descent based method [44].

Coherent Ising Machine Simulator (SimCIM) - This algorithm emulates the behavior of a quantum annealer using CUDA, which allows running on various configurations simultaneously, and choosing from which the minimal evolution of energy [45, 46].

All algorithms ran a dataset of 20k instances of the TSP, each comprising of 15 vertices. The chosen coordinates of each of the vertices for each problem were uniformly distributed [47].

3 Results

In our comparative study, we applied the four algorithms to find optimal solutions to 20k instances of the TSP. The evaluation of these strategies revealed significant differences in their performances.

The **SimCIM** algorithm emerged superior amongst its peers, showcasing remarkable efficacy in optimizing the solutions to the TSP. This method demonstrated a consistent ability to find optimal or near-optimal solutions, indicating its potential as a robust choice for solving TSP efficiently. In over 97% of the cases the algorithm found the optimal solution and was near optima in close to 100% of the cases. In addition to that, convergence was very quick - after less than 1500 iteration in average. The SimCIM's robustness can be attributed to its sophisticated mechanism that leverages quantum mechanics principles to find the shortest possible route, therefore presenting itself as a potentially powerful tool in this computational problem space.

While the **SA** did exhibit good convergence properties, also at less than 1500 iterations in average, it was prone to fluctuations during the optimization process. This propensity for fluctuations indicates that while SA can indeed arrive at satisfying solutions, the path to such solutions may be characterized by periods of instability, potentially limiting its utility in scenarios demanding quick and reliable solutions to TSPs. Despite those fluctuations, the algorithm could achieve an optimal solution in 82% of the cases.

The **QIA** algorithm illustrated a protracted convergence timeline in comparison to the others, after around 8000 iterations, thereby prolonging the solution's attainment. This slower convergence could potentially be a downside when dealing with larger datasets or when time is a critical factor. Despite its slower pace, the Vanilla algorithm eventually managed to converge, offering a solution that, while not the quickest, still held its ground in the competitive landscape of TSP solving techniques, and optimal solution was achieved in 94% of the cases.

The **GD-QIA** was observed to have instability in finding a solid convergence path. This instability translated into inability to find a solution which includes all the cities and inability to pinpoint one solid route. The results suggest a need for further refinement in this method to enhance its reliability and efficiency in solving TSPs, and suggesting, that in its current state for TSP solutions, it may not be the optimum choice for problems demanding high reliability.

4 Conclusion

In conclusion, our analysis shows a clear advantage in using the SimCIM algorithm for TSP, with its superior convergence properties and its ability to reliably find the shortest path. While the SA demonstrated reasonable capability, its fluctuations made it a less reliable choice in comparison to SimCIM. The QIA showed more stability and better success rate, but had a significantly slower convergence. The GR-QIA did not seem suitable, as is, to the problem. This study paves the way for future research centered on using the strengths of the SimCIM algorithm in finding optimal solutions to other combinatorial optimisations.

References

1. Reinelt, G. The traveling salesman: computational solutions for TSP applications. (Springer,2003)
2. Metropolis, Nicholas and Rosenbluth, Arianna W and Rosenbluth, Marshall N and Teller, Augusta H and Teller, Edward: Equation of state calculations by fast computing machines. The journal of chemical physics **21** (6), 1087–1092 (1953)
3. Kirkpatrick, Scott and Gelatt Jr, C Daniel and Vecchi, Mario P: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983)
4. Van Laarhoven, Peter JM and Aarts, Emile HL: Simulated annealing. In Simulated annealing : Theory and applications, pp 7–15. Springer (1987)
5. Ingber, Lester: Simulated annealing: Practice versus theory.Mathematical and computer modelling **18** (11), 29–57 (1993)
6. Apolloni, Bruno and Carvalho, C and De Falco, Diego: Quantum stochastic optimization. Stochastic Processes and their Applications **33** (2), 233–244 (1989)
7. Rutenbar, Rob A: Simulated annealing algorithms: An overview. IEEE Circuits and Devices magazine **5** (1), 19–26 (1989)
8. Kadowaki, Tadashi and Nishimori, Hidetoshi: Quantum annealing in the transverse Ising model. Physical Review E **58** (5), 5355 (1998)
9. Czyżżak, Piotr and Jaszkiewicz, Adrezej: Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization. Journal of multi-criteria decision analysis **7** (1), 34–47 (1998)
10. Szu, Harold and Hartley, Ralph: Fast simulated annealing. Physics letters A **122** (3-4), 157–162 (1987)
11. Moscato, Pablo and Fontanari, José F: Stochastic versus deterministic update in simulated annealing. Physics Letters A **146** (4), 204–208 (1990)
12. Junghans, Lars and Darde, Nicholas: Hybrid single objective genetic algorithm coupled with the simulated annealing optimization method for building optimization. Energy and Buildings **86**, 651–662 (2015)
13. Dong, Lingbo and Bettinger, Pete and Liu, Zhaogang and Qin, Huiyan: A comparison of a neighborhood search technique for forest spatial harvest scheduling problems: A case study of the simulated annealing algorithm. Forest Ecology and Management **356**, 124–135 (2015)
14. Feynman, Richard P and Hey, Tony and Allen, Robin W: Feynman lectures on computation. CRC Press (2018)
15. Steane, Andrew: Quantum computing. Reports on Progress in Physics **61** (2), 117 (1998)
16. Schuld, Maria and Petruccione, Francesco: Machine learning with quantum computers. Springer (2021)
17. Farhi, Edward and Goldstone, Jeffrey and Gutmann, Sam and Sipser, Michael: Quantum computation by adiabatic evolution. arXiv preprint quant-ph/0001106 (2000)
18. Van Dam, Wim and Mosca, Michele and Vazirani, Umesh: How powerful is adiabatic quantum computation?. In: Proceedings 42nd IEEE symposium on foundations of computer science, pp 279–287 (2001)
19. Farhi, Edward and Goldstone, Jeffrey and Gutmann, Sam and Lapan, Joshua and Lundgren, Andrew and Preda, Daniel: A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. Science **292** (5516), 472–475 (2001)

20. Shu, Wanneng: Quantum-inspired genetic algorithm based on simulated annealing for combinatorial optimization problem. International Journal of Distributed Sensor Networks **5** (1), 64–65 (2009)
21. Heim, Bettina and Rønnow, Troels F and Isakov, Sergei V and Troyer, Matthias: Quantum versus classical annealing of Ising spin glasses. Science **348** (6231), 215–217 (2015)
22. Yan, Bin and Sinitsyn, Nikolai A: Analytical solution for nonadiabatic quantum annealing to arbitrary Ising spin Hamiltonian. Nature Communications **13** (1), 1–12 (2022)
23. D-Wave Homepage https://www.dwavesys.com/our-company/meet-d-wave Last accessed 30 January 2023
24. Shin, Seung Woo and Smith, Graeme and Smolin, John A and Vazirani, Umesh: How "quantum" is the D-Wave machine?. arXiv preprint arXiv:1401.7087 (2014)
25. Aharonov, Dorit and Van Dam, Wim and Kempe, Julia and Landau, Zeph and Lloyd, Seth and Regev, Oded: Adiabatic quantum computation is equivalent to standard quantum computation. SIAM review **50** (4), 755–787 (2008)
26. Albash, Tameem and Lidar, Daniel A: Adiabatic quantum computation. Reviews of Modern Physics **90** (1), 015002 (2018)
27. Fedorov, AK and Gisin, N and Beloussov, SM and Lvovsky, AI: Quantum computing at the quantum advantage threshold: a down-to-business review. arXiv preprint arXiv:2203.17181 (2022)
28. Arrazola, Juan Miguel and Delgado, Alain and Bardhan, Bhaskar Roy and Lloyd, Seth: Quantum-inspired algorithms in practice. arXiv preprint arXiv:1905.10415 (2019)
29. Dey, Alokananda and Dey, Sandip and Bhattacharyya, Siddhartha and Snasel, Vaclav and Hassanien, Aboul Ella: 7 Quantum inspired simulated annealing technique for automatic clustering. Intelligent Multimedia Data Analysis **2**, 145 (2019)
30. Sadhu, Tithli and Chowdhury, Somanth and Mondal, Shubham and Roy, Jagannath and Chakrabarty, Jitamanyu and Lahiri, Sandip Kumar: A comparative study of metaheuristics algorithms based on their performance of complex benchmark problems. Decision Making: Applications in Management and Engineering (2022)
31. Fujitsu quantum-inspired Computing Digital annealer user's guide Last accessed 13 November 2022
32. Tatsumura, Kosuke: Large-scale combinatorial optimization in real-time systems by FPGA-based accelerators for simulated bifurcation.In: Proceedings of the 11th International Symposium on Highly Efficient Accelerators and Reconfigurable Technologies, pp 1–6 (2021)
33. Codognet, Philippe and Diaz, Daniel and Abreu, Salvador: Quantum and Digital Annealing for the Quadratic Assignment Problem. In: 2022 IEEE International Conference on Quantum Software (QSW), pp 1–8 (2022)
34. Wang, Hao and Pan, Yu and Cui, Wei: Quantum-Inspired Solvers on Mixed-Integer Linear Programming Problem. arXiv preprint arXiv:2204.04574 (2022)
35. Waidyasooriya, Hasitha Muthumala and Hariyama, Masanori: Temporal and spatial parallel processing of simulated quantum annealing on a multicore CPU. The Journal of Supercomputing **78** (6), 8733–8750 (2022)
36. Oshiyama, Hiroki and Ohzeki, Masayuki: Benchmark of quantum-inspired heuristic solvers for quadratic unconstrained binary optimization. Scientific reports **12** (1), 1–10 (2022)
37. Frausto-Solis, Juan and Liñán-García, Ernesto and Sánchez-Hernández, Juan Paulo and González-Barbosa, J Javier and González-Flores, Carlos and Castilla-

Valdez, Guadalupe: Multiphase simulated annealing based on Boltzmann and Bose-Einstein distribution applied to protein folding problem. Advances in Bioinformatics **2016**, (2016)
38. Hatakeyama-Sato, Kan and Kashikawa, Takahiro and Kimura, Koichi and Oyaizu, Kenichi: Tackling the challenge of a huge materials science search space with quantum-Inspired annealing. Advanced Intelligent Systems **3** (4), 2000209 (2021)
39. Hu, Siyuan and Jordan, Stephen and Boyacioglu, Rasim and Rozada, Ignacio and Troyer, Matthias and Griswold, Mark and McGivney, Debra and Ma, Dan: A Fast MR Fingerprinting Simulator for Direct Error Estimation and Sequence Optimization. arXiv preprint arXiv:2105.11594 (2021)
40. Huang, Tian and Xu, Jun and Luo, Tao and Gu, Xiaozhe and Goh, Rick and Wong, Weng-Fai: Benchmarking Quantum (-inspired) Annealing Hardware on Practical Use Cases. arXiv preprint arXiv:2203.02325 (2022)
41. Jordan, Stephen P and Hu, Siyuan and Rozada, Ignacio and McGivney, Debra F and Boyacioğlu, Rasim and Jacob, Darryl C and Huang, Sherry and Beverland, Michael and Katzgraber, Helmut G and Troyer, Matthias and others: Automated design of pulse sequences for magnetic resonance fingerprinting using physics-inspired optimization. In: Proceedings of the National Academy of Sciences **118** (40), e2020516118 (2021)
42. Apolloni, Bruno and Cesa-Bianchi, Nicolò and De Falco, Diego: A numerical implementation of "quantum annealing". In: Stochastic Processes, Physics and Geometry: Proceedings of the Ascona-Locarno Conference, pp 97–111 (1990)
43. McCaffrey, James: Quantum-Inspired Annealing Using C# or Python. https://visualstudiomagazine.com/articles/2022/01/20/quantum-inspired-annealing.aspx?m= Last accessed 7 July 2023
44. Bowles, Joseph and Dauphin, Alexandre and Huembeli, Patrick and Martinez, José and Acín, Antonio: Quadratic Unconstrained Binary Optimization via Quantum-Inspired Annealing. Physical Review Applied **18** (3), 034016 (2022)
45. Tiunov, Egor S and Ulanov, Alexander E and Lvovsky, AI: Annealing by simulating the coherent Ising machine. Optics express **27** (7), 10288–10295 (2019)
46. Boev, AS and Rakitko, AS and Usmanov, SR and Kobzeva, AN and Popov, IV and Ilinsky, VV and Kiktenko, EO and Fedorov, AK: Genome assembly using quantum and quantum-inspired annealing. Scientific Reports **11** (1), 1–9 (2021)
47. Kaggle: 20k TSPs with optimal tours dataset https://www.kaggle.com/datasets/stephanhocke/15k-tsps-with-optimal-tours, Last accessed 30 August 2023

Reinforcement Learning by Minimizing Constraint Violation

Hortence Nana[1] and Gavin Rens[2]

[1] Stellenbosch University, South Africa
27675173@sun.ac.za
[2] Stellenbosch University, South Africa
gavinrens@sun.ac.za

Abstract. This work contributes to the field of constrained reinforcement learning, focus on the promise of violation measures in encouraging responsible and safe intelligent agent decision-making. We explore the integration of violation measures in constrained reinforcement learning [2] to enhance the decision-making capabilities of intelligent agents in complex environments. Two main approaches are investigated: one involving the incorporation of violation penalization within the reward function and another with external violation measures. Notably, varying parameter configurations and threshold values are used to evaluate the impact on agent performance, which is trained using Q-learning algorithm. Our findings suggest that incorporating violation penalization within the reward function leads to more stable and robust results.

Introduction and Background. This research aims to address the challenge of effectively representing logical safety constraints within the RL framework by introducing a novel violation measure, thereby enhancing the agent's decision-making process in model-free RL using constrained Markov Decision Processes (MDP) to adhere to the safety constraints. A MDP is a formal framework used to model decision-making problems in which an agent interacts with an environment over time [3]. It consists of the tuple (S, A, R, P) [4], where S represents the set of state, A actions, R is the reward function and P represents he transition probabilities from one state to another. Our goal is to find the optimal strategy, known as a policy (π), that dictates which action to choose in each state. An 'environment' in this context refers to the external surroundings and conditions that the agent interacts with during its decision-making process.

In CMDP [5], we introduce an additional element $\mathcal{C}$ which specifies a set of unsafe states that the agent must avoid. The constraint modifies the agent's action space and influences the reward function. The updated tuple $(S, A, R, P, \mathcal{C})$ captures the integration of the safety constraint into the MDP framework. The agent's task now involves finding a policy that maximizes rewards while satisfying the safety constraint. This often requires trade-offs between optimizing rewards and avoiding unsafe actions. The set of constraint is generally expressed as : $\mathcal{C} = \{\mathbf{c_i} : \mathbf{S} \times \mathbf{A} \longrightarrow \mathbb{R} \mid \mathbf{i} = \mathbf{1}, .., \mathbf{k}\}$ [5].

Safety Constraints and Violation Measure. The set of safety constraints $\mathcal{C}$, which are logical conditions [6] in this case, will be defined using some basic statements or propositions that can be true or false, that is, the propositional logic language. These constraints define the criteria that a state must avoid in order to be deemed safe.

Consider we have a set of safety constraints pair $\mathcal{C} = \{c_1, p_1; c_2, p_2; \ldots\}$, where each c_i is a logical expression involving the attributes of the state and p_i the threshold associated to c_i. The general idea is that the robot can transition from s to s' if and only if none of constraints associated with s' are satisfied. If not, the agent should choose an action that does not place him in s'.

The novel violation measure $V^{\mathcal{C}}(a, s)$ evaluates the level of constraint violation. Consider each constraint c_i in $\mathcal{C}$ associated with its threshold p_i. We now define $\mathcal{C}$ to have elements (c, p). Here we define the violation measure by:

$$V^{\mathcal{C}}(a, s) = \frac{1}{|\mathcal{C}|} \sum_{(c,p)\in\mathcal{C}} \max\{0, \widehat{P}(c_i|a, s) - p_i\},$$

where $|\mathcal{C}|$ is the size of $\mathcal{C}$, $\widehat{P}(c_i|a, s)$ is the probability that an action a is unsafe to execute in state s with respect to constraint c_i, and is computed as $\widehat{P}(c_i|a, s) = \frac{N_{a,s,c_i}}{N_{a,s}}$, where N_{a,s,c_i} is the number of times that action a was executed in state s and constraint c_i was satisfied (c_i is true) in the resulting state, and $N_{a,s}$ is the number of time that action a was executed and the constraint was satisfied or not (number of episodes). To incorporate this violation inside the RL framework, we use two method: the first one is the incorporate it directly inside reward and the second one, outside reward. If the difference between the estimated probability and the threshold is positive, it implies a violation.

Methods for Adding constraints. The first method, where we deal with constraints separately from rewards, leverages the combination of Q-values [1] and U-values, where Q-values represent expected cumulative rewards, and U-values represent the expected cumulative violations. The Q and U values are central elements here, and summarise the agent's decision-making process, guaranteeing a balanced compromise between maximising reward and respecting safety.

One popular way to compute function Q is to apply the Q - learning update rule $Q(s_t, a_t) \leftarrow (1 - \alpha)Q(s_t, a_t) + \alpha \cdot [r_t + \gamma\, max_a\, Q(s_{t+1}, a_t)]$ [8], where α is the learning rate and γ the discounted factor. The U-value is updated based on the violation measure $V^C(a, s)$ with the goal of minimizing it. It is defined as:

$$U(s_t, a_t) \leftarrow (1 - \alpha)U(s_t, a_t) + \alpha \cdot \left[V^{\mathcal{C}}(a, s) + \gamma \min_a U(s_{t+1}, a_t)\right].$$

The optimal policy $\pi_*(s)$ that evaluates the trade-off between Q-values and U-values is proposed as: $\pi_*(s) = argmax_{a\in A}\,[Q_*(s_t, a_t) - \beta U_*(s_t, a_t)]$, where β is a weight or scaling factor that determines the trade-off between maximizing rewards and minimizing safety violations.

The second method : "Constraint-Based Reward Approach" aims to discourage unsafe actions by penalizing the agent with negative rewards when safety

constraints are not met. It simply views the constraint violation measure directly in the reward function, and does not update separately U-values as in our first approach.

Preliminary Result. We conducted experiments in two environments, each comprise several essential elements, including states, actions, reward and an exploration-exploitation strategy, where both methods of incorporating constraint violation are used. One is basic and simple, in which we evaluated the agent's performance with several sets of constraints and several values of beta, and the second is a little more complex in terms of size and constraints included, where we evaluated the agent via different values of threshold.

For lack of space, we only present result for the basic environment, with set of one constraint under varying values of β. Outputs are shown in figure 1:

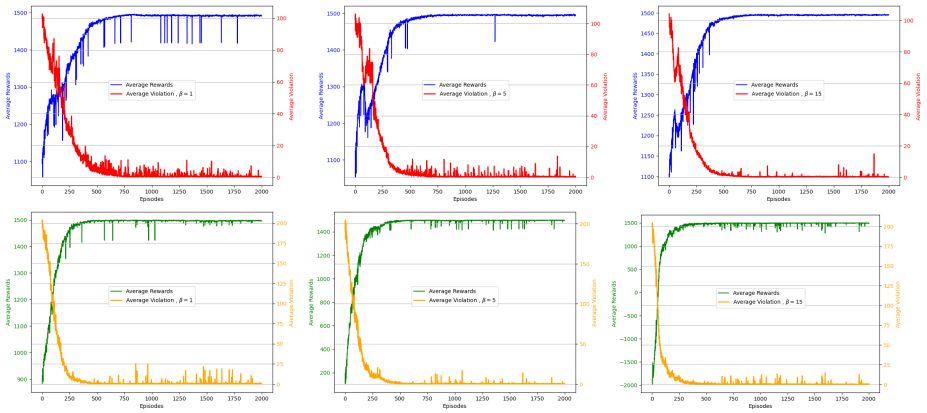


Fig. 1. Average rewards and violations with different β values. Top row: the violation measure is outside the reward function. Bottom row: the violation measure is incorporated inside the reward function.

For both approaches, agent demonstrates an ability to optimise its decisions, by maximising reward and satisfying constraints only after 500 episodes. As we increase β, we notice a significant change in our agent's behaviour. This means that higher values of beta lead to more favorable rewards and violation, indicating that the system is performing better in terms of achieving its objectives. However, the second approach consistently leads to more favorable graphs compared to the first approach for all three beta values.

We also run experiment in this environment with a set of two different constraint, also with different value of β. for the big environment, we have chosen to keep the parameter β fixed while experimenting with two distinct threshold sets. One of these sets contains large values, whereas the other set comprises smaller values.

References

1. Sutton , S., Andrew, G. : Reinforcement learning: An introduction. In: MIT press, (2018).
2. Krishnan, S., Benjamin, E., Sehoon, H., Jie, T., Chelsea, F. : Learning to be safe: Deep rl with a safety critic. arXiv:2010.14603, (2020)
3. Martin, L. :Markov decision processes. Handbooks in operations research and management science, 2, pp.331-434 (1990)
4. Peters, J., Stefan, S. : Reinforcement learning of motor skills with policy gradients. Neural networks, 21(4), pp.682-697 (2008).
5. Yanyang, G., Fei, Z., Xinghong, L., Quan, L.: Safe Q-learning method based on constrained Markov decision processes. In: IEEE Access, 7, pp.165007-165017 (2019)
6. Wen-Chi, Y., Giuseppe, M., Gavin, R., Luc De R. : Safe Reinforcement Learning via Probabilistic Logic Shields. In : 31st International Joint Conference on Artificial Intelligence, (2022)
7. Ding, Z., Yanhua, H., Hang, Y., Hao, D. : Introduction to reinforcement learning. Deep reinforcement learning. fundamentals, research and applications, pp.47-123 (2020)
8. Jang, B., Myeonghwi, K., Gaspard, H., and Jong, K.: Q-learning algorithms: A comprehensive classification and applications. In: IEEE access, 7, pp.133653-133667 (2019)

Predicting South African Protests using Graph Neural Networks

Derwin Ngomane[1][0009−0004−8624−2402] and Vukosi Marivate[1,2,3][0000−0002−6731−6267]

[1] Department of Computer Science, University of Pretoria
[2] Lelapa AI
[3] Masakhane NLP
derwin.ngomane@gmail.com, vukosi.marivate@cs.up.ac.za

Abstract. This study examines the use of Twitter data to predict South African occurrences. Comparing Logistic Regression, Graph Convolutional Network, Dynamic Graph Convolutional Network, and Graph Isomorphism Network. It is revealed that the graph neural network models has the highest F1 score.

Keywords: Protests · Social Media · Graph Neural Networks.

1 Introduction

Social media has the ability to create echo chambers, reinforcing beliefs and opinions of users [7]. This effect is closely related to social movement theory, which focuses on mobilizing individuals for social movements [12]. In South Africa, social media has been accused of facilitating violent protests [20].

This study aims to determine if social media can be manipulated to predict future protests in South Africa. The research aims to use Twitter data to anticipate protests and predict protest-related occurrences using Graph Neural Networks.

2 Related Work

Protests are used by participants to shape public policy and advance democracy [19]. Protest participation is characterised by three (3) factors: injustice towards an in-group, the belief that a protests will result in immediate results, and the in-group identity of protest participants [5, 1]. South Africa has had a significant increase in protests that have been sparked by socioeconomic issues [3]. Although it is a democratic right in South Africa to hold peaceful protests, the protests have become violent and turning into unrests [18].

The prospects of protests turning violent has been evident since 1994 [17]. It has been found that violent protest are more prevalent amongst the youth demographic [2]. The internet has boosted young political engagement and it

has been observed through some prominent social media protests that have been associated with the "mustfall" and "shutdown" twitter hashtags [8].

The consequences of violent protests has a negative impact on infrastructure and security. Hence, ex ante measures are needed to prevent the destruction of important infrastructure [11]. There have been different types of machine learning methods used for event prediction on social media data, such as, logistic regression [4]; deep learning methods, such as, Recurrent Neural Networks [23] and attention-based neural networks [10]; and graph based methods [9]. The methods that have been used for protest prediction in South Africa have relied mostly only on numerical data [23] or generating sparse features from English only text documents [6], which does not account to the non-static dynamic nature of social media data. Additionally, similar work have used post-event knowledge to perform data filtration of the twitter data [6, 9], which is not possible in a real-world context. In this work, we investigate graph-neural network using multilingual social media data. Additionally, we use noise-reduction and a TF-IDF based relevancy score to perform pre-event filtration of the twitter data.

3 Methodology

This study examines a three-year data window between 2019 and 2021 using the Global Dataset of Events, Location, and Tone (GDELT) [15] and Twitter data [14]. The GDELT data used focuses on CAMEO codes related to protests and conflict occurrences, while Twitter is related to protest-related posts. A sliding window approach is used to combine the two datasets (see Figure 1). In this work we explored the potential of processing Twitter data to reveal signals for improved predictive capability. Derivative metrics, from hashtags, links, and mentions, were used for noise filtration in order to improve the signal in the data. Additionally, the removal of stopwords was done using the TF1 method [21] because of its benefit to the feature space and the sparsity degree of the dataset.

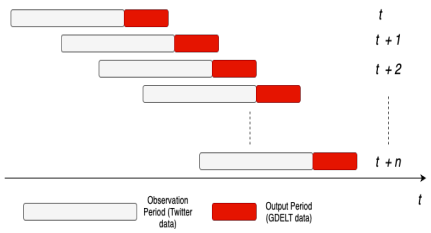


Fig. 1. Sliding window

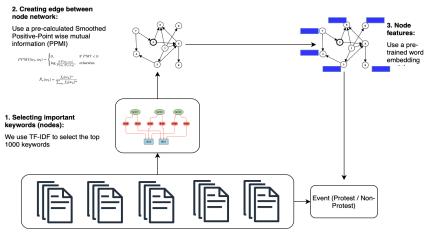


Fig. 2. Graph generation process

The study compares different machine learning methods, including Logistic Regression, Graph Convolutional Network (GCN) [13], Dynamic Graph Convolutional Network (DGCN) [9] and Graph Isomorphism Network (GIN) [22],

to model the data. A feature matrix is generated using TF-IDF values from the Twitter data for logistic regression and the graph representations process as depicted by Figure 2 is used for the graph word networks. Two versions of graph representations are generated, i.e.,dynamic and static. The twitter data is ordered sequentially by post date, thereafter, for each protest-related event occurrence as given by the GDELT data, the prior 5 days of tweets are extracted. Finally, a selection process is used to find relevant keywords using a relevancy score as defined by Equation 1, for each token w_i over all documents k in the corpus that contain w_i. A selection of 1000 keywords are used to construct the word network with nodes as keywords and edges based on smoothed positive pointwise mutual information [16]. Finally, skip-gram word2vec word embeddings are used to represent each node's feature matrix due to the size of the data set.

$$r(w_i) = \frac{1}{k_{w_i}} \sum_{k_{w_i}} \mathbf{x}_k \tag{1}$$

4 Results

Cross-validation is used to estimate mean average performance of the models. Table 1 shows that geometric deep learning methods have higher average precision, with the GCN models having the highest average positive class F1 score. It can be observed that the GCN models tend to have a higher recall, hence detecting a majority of the days with protests. However, we note that the GIN model has a high precision which implies that it less likely to get incorrect predictions than the logistic regression and GCN model.

Table 1. Cross-validation performance

Model	**F1**	**Recall**	**Precision**
Logistic Regression	0.835 ± 0.050	0.927 ± 0.021	0.763 ± 0.076
GCN	0.875 ± 0.056	1.000 ± 0.000	0.781 ± 0.086
GIN	0.820 ± 0.079	0.780 ± 0.078	0.881 ± 0.133
DynamicGCN	0.870 ± 0.047	0.986 ± 0.005	0.782 ± 0.071

5 Conclusion

The geometric deep learning methods tend to quickly overfit training data due to the data size. However, they are more stable due to their expressive graph embeddings, hence allowing them to predict protest-related occurrences more effectively. This can be observed from the high F1 score. Future research should explore data processing and model building in heterogeneous networks. Keyword selection and isomorphism are crucial for network performance, and further investigation should explore dynamic capabilities and the addition of edge weights to isomorphic graph neural networks.

References

1. Barbera, S., Jackson, M.O.: A model of protests, revolution, and information. Revolution, and Information (October 2019) (2019)
2. Bedasso, B.E., Obikili, N.: A dream deferred: The microfoundations of direct political action in pre-and post-democratisation south africa. The Journal of Development Studies **52**(1), 130–146 (2016)
3. Bonga, W.G.: Impact of repetitive protests on economic development: A case of south africa. Quest Journals' Journal of Research in Humanities and Social Science **9**(8), 34–39 (2021)
4. Cadena, J., Korkmaz, G., Kuhlman, C.J., Marathe, A., Ramakrishnan, N., Vullikanti, A.: Forecasting social unrest using activity cascades. PloS one **10**(6), e0128879 (2015), ISBN: 1932-6203 Publisher: Public Library of Science San Francisco, CA USA
5. Chan, M.: Media use and the social identity model of collective action: Examining the roles of online alternative news and social media news. Journalism & Mass Communication Quarterly **94**(3), 663–681 (2017), number: 3 ISBN: 1077-6990 Publisher: SAGE Publications Sage CA: Los Angeles, CA
6. Chinta, A., Zhang, J., DeLucia, A., Dredze, M., Buczak, A.L.: Study of manifestation of civil unrest on twitter. In: Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021). pp. 396–409 (2021)
7. Cinelli, M., Morales, G.D.F., Galeazzi, A., Quattrociocchi, W., Starnini, M.: The echo chamber effect on social media. Proceedings of the National Academy of Sciences **118**(9) (2021)
8. Daniels, G.: Scrutinizing Hashtag Activism in the MustFall Protests in South Africa in 2015, pp. 175–193 (12 2016)
9. Deng, S., Rangwala, H., Ning, Y.: Learning dynamic context graphs for predicting social events. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1007–1016 (2019)
10. Ertugrul, A.M., Lin, Y.R., Chung, W.T., Yan, M., Li, A.: Activism via attention: interpretable spatiotemporal learning to forecast protest activities. EPJ Data Science **8**(1), 5 (2019), ISBN: 2193-1127 Publisher: Springer Berlin Heidelberg
11. Khambule, I., Nomdo, A., Siswana, B.: Burning capabilities: the social cost of violent and destructive service delivery protests in south africa. African Journal of Peace and Conflict Studies **8**(1), 51 (2019)
12. Killian, L.M., Turner, R.H., Smelser, N.J.: "social movement". Encyclopedia Britannica (2020), https://www.britannica.com/topic/social-movement
13. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
14. Kwak, H., Lee, C., Park, H., Moon, S.: What is twitter, a social network or a news media? pp. 591–600 (2010)
15. Leetaru, K., Schrodt, P.A.: Gdelt: Global data on events, location, and tone, 1979–2012. In: ISA annual convention. vol. 2, pp. 1–49. Citeseer (2013)
16. Levy, O., Goldberg, Y., Dagan, I.: Improving distributional similarity with lessons learned from word embeddings. Transactions of the association for computational linguistics **3**, 211–225 (2015)
17. Matebesi, S., Botes, L.: Party identification and service delivery protests in the eastern cape and northern cape, south africa. African Sociological Review/Revue Africaine de Sociologie **21**(2), 81–99 (2017), number: 2 ISBN: 1027-4332

18. Mottiar, S.: From 'popcorn'to 'occupy': protest in durban, south africa **44**(3), 603–619, ISBN: 0012-155X Publisher: Wiley Online Library
19. Passarelli, F., Tabellini, G.: Emotions and political unrest. Journal of Political Economy **125**(3), 903–946 (2017), number: 3 ISBN: 0022-3808 Publisher: University of Chicago Press Chicago, IL
20. Pillay, Y., Mtshali, S.: Social media was 'instrumental' in july unrest expert tells sa human rights commission. IOL (2021)
21. Saif, H., Fernandez, M., He, Y., Alani, H.: On stopwords, filtering and data sparsity for sentiment analysis of Twitter. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14). European Language Resources Association (ELRA), Reykjavik, Iceland (May 2014)
22. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? (2019)
23. Zambezi, S.: Predicting social unrest events in South Africa using LSTM neural networks. Master's thesis, University of Cape Town (2021)

Vision Transformers for Lung Cancer Diagnosis

Abdul Gany Osman[1] and Yuvkia Singh[2]

School of Mathematics, Statistics and Computer Science,
University of KwaZulu-Natal, South Africa
220002001@stu.ukzn.ac.za

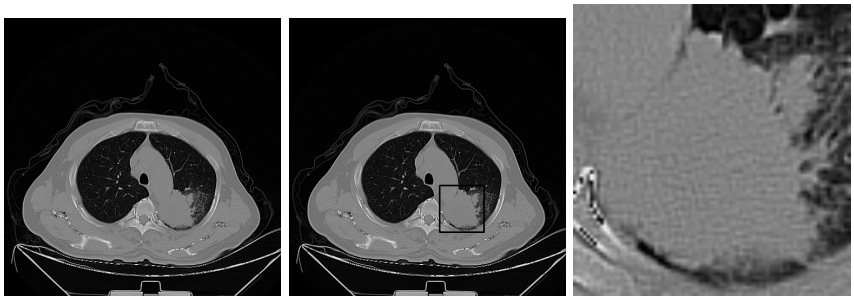
1 Extended Abstract

Vision Transformers (ViTs) are a relatively new Computer Vision Breakthrough. They have been shown to outperform CNNs on a large enough dataset, this project aims to use this information to bring and explore the effectiveness of this technology into the field of lung cancer diagnosis. Lung cancer is the third most common form of cancer and the leading cause of death to cancer[4]. Correctly diagnosing its early stages can significantly improves one's chance of surviving it and as such doing so is vital. CT and PET scans are the most common imaging methods used in cancer detection and diagnosis, however these scans expose patients to incredibly large amounts of radiation which in itself can potentially cause a cancer. As such there is a need to diagnose lung cancer early on and with as few tests as possible hence a project that aids in this is vital. There are different types of lung cancer which require different treatments, as such identifying them correctly and early is of vital importance. This project created a Vision Transformer model to learn how to identify the type of lung cancer from CT and PET scans correctly while also exploring the different ways to prepare the images for the model to find which worked best.

The process of identifying lung cancer type is called histologic classification and is exactly what this model aims to do. It classifies lung cancer nodules into one of the following four lung cancer types:

- Adenocarcinoma
- Small cell carcinoma
- Large cell carcinoma
- Squamos cell carcinoma

To train the model the Lung-PET-CT-Dx [3] dataset was used. This dataset consits of CT and PET-CT Dicom images with xml annotation files segmenting the nodes. This project used the Dicom images and annotations to obtain all the images that has nodules present. These images were then saved in three ways, one was just the image as it was, one was the image with a bounding box drawn around the nodule and lastly the image with just the area of interest cropped. This resulted in 26758 images being available to be used for training and testing of each model. A train-test split of 80-20 was used for the training of each model. Examples of three types of images follow:



(a) Unsegmented Image (b) Segmented Image (c) Cropped Image

Fig. 1: The three different types of images used

The three different models were trained with identical configurations and parameters with just the actual images differing. The images were shrunk down to 72 by 72 greyscale images due to memory constraints. A sliding window of 6 by 6 pixels were used to get the patches. Each model was trained for 100 epochs, batch size of 256, with an Adam optimizer and Sparse Categorical Crossentropy loss function.

The accuracy and top-2-accuracy of each model was obtained with the following results being produced:

Model	Accuracy	Top-2-Accuracy
ViT-Unsegmented-Model	45.83%	72%
ViT-Segmented-Model	51.35%	78.10%
ViT-Cropped-Model	62.73%	90.49%

Greater results were obtained in other papers such as Barbouchi *et al* [1], they were able to get on 96% accuracy on the Lung-PET-CT-Dx dataset, however they left out LCC examples as they focused on PET/CT images only which were not present in the LCC cases of the dataset while this project used all three types of scans.

To conclude, from these results it is observed the model that was fed cropped images worked the best. The model with unsegmented images was slightly behind the segmented one. This however could be due to the fact that more detail in the segmented and unsegmented images were lost in shrinking it due to memory constraints. Another important thing to note is that as proven in the initial Vision Transformer paper [2], to outperform CNNs a very large dataset is required, as such with a larger dataset Vision Transformers can go a long way in solving this problem.

References

[1] Khalil Barbouchi et al. "A transformer-based deep neural network for detection and classification of lung cancer via PET/CT images". en. In: *International Journal of Imaging Systems and Technology* n/a.n/a (). ISSN: 1098-1098. DOI: 10.1002/ima.22858. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/ima.22858 (visited on 05/01/2023).

[2] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. arXiv:2010.11929 [cs]. June 2021. DOI: 10.48550/arXiv.2010.11929. URL: http://arxiv.org/abs/2010.11929 (visited on 05/02/2023).

[3] Ping Li et al. *A Large-Scale CT and PET/CT Dataset for Lung Cancer Diagnosis*. 2020. DOI: 10.7937/TCIA.2020.NNC2-0461. URL: https://wiki.cancerimagingarchive.net/x/WIkvB (visited on 05/05/2023).

[4] Lindsey A. Torre, Rebecca L. Siegel, and Ahmedin Jemal. "Lung Cancer Statistics". en. In: *Lung Cancer and Personalized Medicine: Current Knowledge and Therapies*. Ed. by Aamir Ahmad and Shirish Gadgeel. Advances in Experimental Medicine and Biology. Cham: Springer International Publishing, 2016, pp. 1–19. ISBN: 978-3-319-24223-1. DOI: 10.1007/978-3-319-24223-1_1. URL: https://doi.org/10.1007/978-3-319-24223-1_1 (visited on 05/02/2023).

A Comparative Analysis on Decision Transformers and Behavioural Cloning for Offline Reinforcement Learning in ATARI Environments

Sachin Reddy
Supervisor: Asad Jeewa

School of Mathematics, Statistics and Computer Science,
University of KwaZulu-Natal , Durban, KwaZulu-Natal, South Africa
220025233@stu.ukzn.ac.za

Abstract. In recent years, transformer-based models have demonstrated remarkable success in various natural language processing tasks, leading to their exploration in the field of reinforcement learning. The Decision Transformer, an extension of the GPT architecture to Offline Reinforcement Learning, was shown to outperform standard imitation learning (Behavioural Cloning) and perform comparatively with SOTA algorithms such as Conservative Q-Learning on various reward-dense environments from the ATARI-replay Offline dataset. However, it remains to be seen how well DT performs in comparison to Behavioural Cloning in environments with difficult exploration. Our study leverages two ATARI video-game environments with varying exploration difficulty, to create an evaluation framework. We make deductions about each algorithm's performance in environments where exploration is a challenge. Our results support previous literature that shows Decision Transformer to outperform Behavioural Cloning in GVGP environments, as well as suggesting DT generalizes well to environments that require strategic exploration but have dense rewards.

Keywords: Offline Reinforcement Learning · Transformers · Video Game Environment

1 Introduction

With the success of Transformer models in areas such as Computer Vision and Natural Language Processing, Transformer-based RL approaches have started to gain prominence, such as the Decision Transformer (DT), [3] which abstracts RL as a sequence modelling problem and leverages a generative transformer to output actions.
The Decision Transformer has been shown to outperform conventional Offline RL approaches, such as CQL (Conservative Q-Learning), [6] and Behavioural Cloning [1], in certain reward-dense environments from the Arcade Learning

Environment (ALE) [2], and D4RL benchmark collection. [3]
General Video Game Playing (GVGP) [4] has emerged as a challenging task for Artificial Intelligence agents, as it requires them to learn different game environments and their corresponding rules. In such environments we define exploration to be the process through which an agent interacts with the environment to gain an understanding of it, and experiment with its own features.
Conventional implementations of Reinforcement Learning are known to struggle in environments where such exploration is difficult, due to factors such as reward-sparsity, necessity of strategic play, and movement complexity [7]. We investigated to what advantages/disadvantages Decision Transformer may have held over Behavioural Cloning in such environments.

2 Method

To investigate the exploration ability of Decision Transformer and Behavioural Cloning, we required a baseline environment with low-difficulty exploration, and an environment with high-difficulty exploration for testing.
2 environments from ALE were chosen, namely:
Breakout, and Frostbite.
Breakout is characterized by dense rewards and easy exploration, due to its 1-dimensional movement and repetitive gameplay, whereas Frostbite is characterized by dense rewards and difficult exploration, due to the complex 2-dimensional movement required in traversing the moving platforms, as well as the strategic gameplay in constructing an igloo via jumping on consecutive ice floes.
The hyper-parameters used for each approach are taken from Chen et al, who consistently trained over 5 epochs with a batch size of 128, 50 buffers, and 1% training data for all approaches in all environments.
Each approach is trained and evaluated for three different seeds for each ATARI game. For the training, a 1% sample of all offline data pertaining to that game was used as a training set as is done in Chen et al. [3].
Once every 20 timesteps of learning, we evaluated the model by deploying it within the relevant environments from ALE in order to produce training curves. We used 3 seeds for each algorithm on each environment for better validity. We took the outputted scores from the experiments, and averaged them across the three seeds, per timestep, per algorithm, per environment in order to produce the data necessary for 4 training curves.
From the mean scores, we obtained normalized scores, based on scores obtained from a SOTA Q-Learning algorithm, namely Double DQN [5]. A normalized score of 100 represents being on par with DDQN. From these normalized scores, we plotted the training curves. Due to the extremely high variance seen in the data, a rolling mean of 20 was used for smoothing purposes.

3 Results

	DT	BC
Normalized mean score	19.09	12.69
Normalized std dev	1.61	2.18

Table 1. DDQN Normalized Scores for ATARI Breakout.

	DT	BC
Normalized mean score	70.35	39.37
Normalized std dev	12.09	6.70

Table 2. DDQN Normalized Scores for ATARI Frostbite.

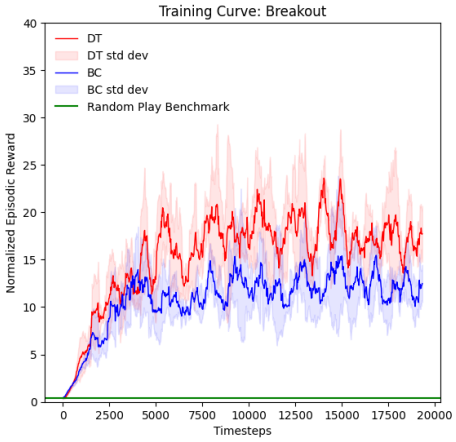


Fig. 1. Training Curve for Breakout

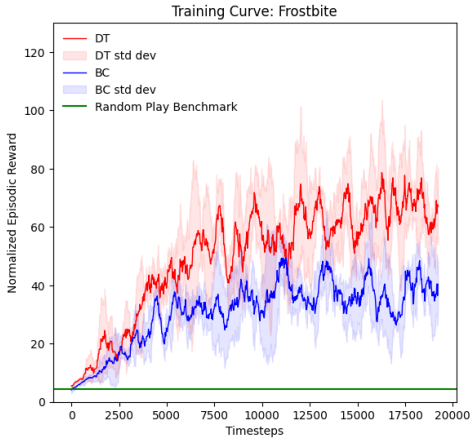


Fig. 2. Training Curve for Frostbite

4 Discussion

4.1 Breakout

The performance seen in Breakout is in line with Chen et al, where DT achieved a DDQN normalized score of 22.26, and BC achieved a DDQN normalized score of 11.56 [3]. The difference in performance between our results and Chen et al's can be attributed to the high variance in performance demonstrated by both algorithms. DT significantly outperforms BC, with a 50% score improvement, as expected from what is seen in literature [3].

4.2 Frostbite

DT continues to outperform Behavioural Cloning, achieving a 79% score improvement over BC. This is a significant increase in disparity than to that seen in Breakout, suggesting that DT is more robust to such dense-reward, difficult-movement, hard-exploration environments.

References

[1] Michael Bain and Claude Sammut. "A Framework for Behavioural Cloning". In: *Machine Intelligence 15*. 1995. URL: https://api.semanticscholar.org/CorpusID:10738655.

[2] M. G. Bellemare et al. "The Arcade Learning Environment: An Evaluation Platform for General Agents". In: *Journal of Artificial Intelligence Research* 47 (June 2013), pp. 253–279. DOI: 10.1613/jair.3912. URL: https://arxiv.org/abs/1207.4708.

[3] Lili Chen et al. *Decision Transformer: Reinforcement Learning via Sequence Modeling*. 2021. arXiv: 2106.01345 [cs.LG].

[4] Michael Genesereth, Nathaniel Love, and Barney Pell. "General Game Playing: Overview of the AAAI Competition". In: *AI Magazine* 26.2 (June 2005), p. 62. DOI: 10.1609/aimag.v26i2.1813. URL: https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/1813.

[5] Hado van Hasselt, Arthur Guez, and David Silver. *Deep Reinforcement Learning with Double Q-learning*. 2015. arXiv: 1509.06461 [cs.LG].

[6] Aviral Kumar et al. *Conservative Q-Learning for Offline Reinforcement Learning*. 2020. arXiv: 2006.04779 [cs.LG].

[7] Nathan Lambert et al. *The Challenges of Exploration for Offline Reinforcement Learning*. 2022. arXiv: 2201.11861 [cs.LG].

Training Graph Neural Networks with Particle Swarm Optimisation

Regan Shen[1], Anna Sergeevna Bosman[1], Arné Schreuder[1], Maciej Krzywda[2], and Szymon Łukasik[2]

[1] Dept. of Computer Science, University Of Pretoria, Pretoria, Gauteng, South Africa
[2] AGH University of Science and Technology, Al. Mickiewicza 30, 30-059, Kraków, Małopolska, Poland

Abstract. Graph neural networks (GNNs) have emerged as a key research area in the neural network field. While numerous advancements have been made, there is limited research on using different optimisation algorithms to train GNNs. This paper investigates the potential of particle swarm optimisation (PSO) for training GNNs. A novel PSO variant, AdamPSO, is proposed, which combines the Adam optimiser with PSO. Preliminary results indicate that AdamPSO can rival or even surpass the performance of Adam, particularly on smaller GNN models. With appropriate parameter adjustments, AdamPSO holds significant potential in the realm of GNN training.

Keywords: metaheuristics · particle swarm optimization · graph convolutional networks · graph attention networks.

1 Introduction

Introduced in 2008 [7], graph neural networks (GNNs) have become a prominent research area due to their ability to model graph-structured data [12]. While the use of metaheuristic optimisation algorithms on neural networks has a considerable amount of research [1, 3, 8], their application to enhance GNN training remains a comparatively unexplored area.

PSO is a population-based optimisation algorithm, where the search is conducted by simulating the social behaviour of a swarm. The population (swarm) consists of candidate solutions (particles) [9]. Each particle's movement in the search space is determined through a velocity and position update. Particles collaboratively use their personal best positions and the swarm's global best position to determine their movement direction and speed [4]. This study uses PSO to train graph convolutional networks (GCN) [5], inspired by convolutional neural networks (CNN), and graph attention networks (GAT) [10], which enhance GCNs by incorporating self-attention mechanisms.

2 Proposed Method

Gradient descent PSO (GDPSO) is a PSO variant that uses gradient information together with the social information to guide the search [2]. GDPSO performs

standard PSO iterations, and at each iteration updates the global best using the stochastic gradient descent (SGD) algorithm. We propose AdamPSO, where the Adam optimiser [6] is used in the GDPSO setting in place of SGD algorithm. By using Adam exclusively for the global best, AdamPSO retains the original PSO's explorative advantages, offering a balance between exploration and exploitation.

The AdamPSO algorithm was tested on three GNN models: two GCN variants and one GAT. The first GCN is more complex and specially tailored for text classification [11], evident in its larger embedding size of 200, in contrast to the second GCN model that was taken from the original GCN architecture, with an embedding size of 4 per layer [5]. Six datasets were used: 20 news group (20NG), Movie Review (MR), Cora, Citeseer, Pubmed, and Karate. AdamPSO was benchmarked against Adam, SGD, PSO, and GDPSO.

3 Results

Table 1 displays the average test accuracy and standard deviation across 10 runs for each algorithm on the different dataset. According to Table 1, Adam marginally outperformed AdamPSO for the text classification GCN and GAT. However, AdamPSO significantly outperformed Adam for the second GCN. Both the text classification GCN and GAT have higher dimensionality compared to the second GCN model. Based on these observations, we conclude that AdamPSO is more effective than Adam for models of lower dimensionality. We have also observed that AdamPSO converged faster than Adam, implying that fewer iterations may be required to reach the solution.

4 Conclusion

In this paper, a selection of PSO algorithm variants were applied to GCN training and benchmarked against standard gradient-based Adam training. AdamPSO was proposed, which applies the Adam update to the global best particle of the swarm. Results indicated that AdamPSO performed comparably to Adam, and outperformed Adam on smaller problems. Future work will include a study of AdamPSO hyperparameters, and will consider other ways of hybridising gradient information with the social information of the swarm.

Table 1. Test accuracy across the optimisation algorithms

Model	GCN 1		GCN 2			GAT		
Dataset	MR	20NG	Cora	Citeseer	Pubmed	Cora	Citeseer	Pubmed
Algorithm								
AdamPSO	72.78 ± 0.39	84.74 ± 0.24	96.59 ± 0.38	89.08 ± 1.85	95.0 ± 0.41	81.7 ± 2.35	77.17 ± 0.39	83.18 ± 0.44
Adam	76.25 ± 0.46	86.26 ± 0.16	66.35 ± 4.3	55.77 ± 5.99	86.31 ± 0.27	83.83 ± 0.48	78.15 ± 0.31	84.86 ± 0.28
SGD	49.71 ± 2.07	5.44 ± 0.23	35.47 ± 5.27	34.22 ± 9.13	39.53 ± 1.41	21.37 ± 5.47	22.14 ± 4.36	40.42 ± 10.99
PSO	50.75 ± 1.01	5.11 ± 0.83	35.8 ± 3.71	27.3 ± 3.83	66.92 ± 5.43	16.62 ± 4.6	18.24 ± 3.34	41.9 ± 6.72
GDPSO	51.3 ± 1.3	6.64 ± 0.67	88.45 ± 3.75	73.48 ± 2.74	86.22 ± 1.62	82.66 ± 0.96	77.2 ± 0.54	83.20 ± 0.37

References

1. Alba, E., Martí, R.: Metaheuristic procedures for training neural networks, vol. 35. Springer Science & Business Media (2006)
2. Barla-Szabó, D., et al.: A study of gradient based particle swarm optimisers. Master's thesis, University of Pretoria (2010)
3. Chong, H.Y., Yap, H.J., Tan, S.C., Yap, K.S., Wong, S.Y.: Advances of metaheuristic algorithms in training neural networks for industrial applications. Soft Computing **25**(16), 11209–11233 (2021)
4. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks. vol. 4, pp. 1942–1948. IEEE (1995)
5. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
6. Luo, D., Cheng, W., Xu, D., Yu, W., Zong, B., Chen, H., Zhang, X.: Parameterized explainer for graph neural network. Advances in neural information processing systems **33**, 19620–19631 (2020)
7. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE transactions on neural networks **20**(1), 61–80 (2008)
8. Si, T., Bagchi, J., Miranda, P.B.: Artificial neural network training using metaheuristics for medical data classification: an experimental study. Expert Systems with Applications **193**, 116423 (2022)
9. Talbi, E.G.: Metaheuristics: from design to implementation. John Wiley & Sons (2009)
10. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., et al.: Graph attention networks. stat **1050**(20), 10–48550 (2017)
11. Yao, L., Mao, C., Luo, Y.: Graph convolutional networks for text classification. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 7370–7377 (2019)
12. Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M.: Graph neural networks: A review of methods and applications. AI Open **1**, 57–81 (2020). https://doi.org/https://doi.org/10.1016/j.aiopen.2021.01.001, https://www.sciencedirect.com/science/article/pii/S2666651021000012