IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Incorporating Deep Q - Network With Multiclass Classification Algorithms

[1]Noopur Zambare, [2]Ravindranath Sawane

[1,2]Student

[1]Department of Mechanical Engineering, [2]Department of Computer Science

[1]Indian Institute of Technology, Jodhpur, India, [2]Western University, Ontario, Canada

Abstract: In this study, we explore how Deep Q-Network (DQN) might improve the functionality of multiclass classification algorithms. We used a benchmark dataset from Kaggle to create a framework incorporating DQN with existing supervised multiclass classification algorithms. They have been used in a number of fields, including image recognition, natural language processing, and bioinformatics. This study is focused on the prediction of financial distress in companies in addition to the wider application of Deep Q-Network in multiclass classification. Identifying businesses that are likely to experience financial distress is a crucial task in the fields of finance and risk management. Whenever a business experiences serious challenges keeping its operations going and meeting its financial responsibilities, it is said to be in financial distress. It commonly happens when a company has a sharp and sustained recession in profitability, cash flow issues, or an unsustainable level of debt.

***Index Terms* - DQN (Deep Q - Network), Deep Reinforcement Learning, Financial Distress, Multiclass Classification, Decision Tree Classifier, Naive Bayes, Random Forest Classifier**

I. INTRODUCTION

1.1 Background

The goal of Reinforcement Learning (RL), is to train agents how to make decisions sequentially in an environment that optimises a reward signal. By interacting with the environment, getting feedback in the form of rewards or penalties, and adapting their behaviour in response, RL algorithms learn through trial and error. The Deep Q-Network (DQN) is a deep reinforcement learning method that combines the Q-learning algorithm and the capability of deep neural networks.

Financial distress refers to a state in which a company faces considerable challenges in meeting its financial obligations. Early indications of financial problems might help proactive actions like restructuring, obtaining more finance, or putting cost-cutting measures into place.

We used a wide range of supervised learning algorithms, such as Decision Tree, Random Forest Classifier, and Naive Bayes, to create the DQN framework. The DQN ensemble's underlying models are represented by these algorithms. We intend to study the potential advantages and performance enhancements that can be achieved by combining supervised learning with the reinforcement learning approach of DQN using supervised learning algorithms as the foundation models. The use of DQN for multiclass classification to forecast financial difficulties in businesses is explored in this study.

2.2 Problem Statement

The goal of this paper is to investigate the use of Deep Q-Network in multiclass classification problems. We intend to adapt and use DQN's skills for resolving multiclass classification issues despite the fact that its typical application is mostly in the field of reinforcement learning. The subject of interest is the application of DQN for multiclass classification to predict financial distress in businesses.

By effectively resolving this problem, we want to open up the possibility of applying reinforcement learning principles to a variety of classification problems.

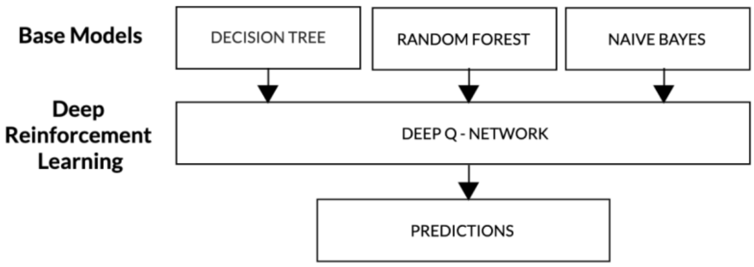

Fig. 1. Deep Q-Network framework with baseline models

II. STATE OF THE ART

In DQN, our goal is to train an action-value function Q(s, a) that calculates the predicted cumulative reward for performing action 'a' in state 's'.

The Bellman Equation or Q-Learning update equation is defined as follows:

$$Q(s, a) = (1-\varepsilon)\, Q(s, a) + \alpha\, [r + \gamma \max Q(s', a') - Q(s, a)] \quad (1)$$

where,
Q(s, a) = Current estimate of the predicted future benefits of action 'a' in state 's'
ε = exploration-exploitation trade-off
α = learning rate
r = immediate received reward
γ = discount factor

A variation of the Q-learning process called Deep Q-Network makes use of neural networks to make approximations of the Q-value function. The expected reward for performing a specific action in a given condition is provided by the Q-value function. The Q-value function is represented as a table in conventional Q-learning but as a neural network in DQN.

Experience replay and a technique called fixed Q-targets are both used by the DQN algorithm to stabilise the learning process. Experience replay involves sampling small batches of experiences for training and storing observed transitions (s, a, r, and s') in a replay buffer. Using a target network with set parameters for a predetermined number of iterations before updating it with the parameters of the online network is recognised as leveraging fixed Q-targets.

III. METHODOLOGY

3.1 Dataset

The study involves the use of a dataset gathered by Kaggle that includes different financial parameters and company characteristics. The dataset, which is accessible in CSV format, includes statistics on the company's performance as well as relevant contextual information. Using methods like label encoding, a preprocessing step is implemented to handle missing data, normalise features, and transform categorical variables. Then, training and testing sets are created from the preprocessed dataset.

3.2 Baseline Multiclass Classification Algorithms

3.2.1 Decision Tree

In this algorithm, the space of features is recursively divided according to a set of criteria in order to generate a decision tree. Information gain or Gini impurity is the most widely used criterion.

They can handle categorical and numerical features, as well as non-linear relationships, and they can capture both. Decision trees, show a tendency to overfit the training set if they are not appropriately regularised or pruned. Overfitting can be reduced using strategies like pruning, establishing a minimum number of samples needed to split a node, or using ensemble methods.

3.2.2 Random Forest Classifier

An ensemble technique called the Random Forest Classifier combines several decision trees to produce predictions. A random subset of features is taken into account at each split of each tree, which is trained on a bootstrap sample of the training data. Random Forest's main idea is to generate a "forest" of decision trees, with each tree trained on a random sample of the data and making independent predictions. By combining the predictions of various trees, either through majority voting or averaging, the final prediction is obtained.

3.2.3 Naive Bayes

The Naive Bayes algorithm is a probabilistic classifier that relies on the Bayes theorem and makes the assumption that features are independent of the class. Given the input features, it calculates the probabilities of each class and chooses the class with the highest probability as the prediction.

Naive Bayes is a computationally efficient algorithm that performs satisfactorily on high-dimensional data. However, when dealing with correlated characteristics, the "naive" assumption of feature independence may prove limiting.

3.3 Multiclass Classification Algorithms with DQN Integration

Combining a Deep Q-Network (DQN) and supervised models involves merging DQN's reinforcement learning framework with supervised learning models' predictive power. This method takes advantage of the agent's ability to learn from incentives while making data-driven decisions, leading to more accurate categorization predictions. This integration is especially valuable when the agent has the opportunity to find complicated patterns that are essential to accurate classification.

- *Defining Agent*
  The DQN class is used to represent the agent. Based on the input features given, it acts as the decision-making entity that learns to categorise the different levels of financial distress. The agent employs a method akin to the DQN, using a group of Decision Tree Classifier, Random Forest Classifier and Naive Bayes models as the Q-network.

- *Defining Environment*
  In this case, the environment is the classification problem itself, which involves determining the levels of economic distress based on the given input features. The agent receives rewards from the environment as feedback, which helps it improve its classification performance.

- *State Representation*
  The input features that were utilised to train the agent define the state representation. In this instance, the features Company, Time, x1, x2, x3, and x4 serve as representations of the state. These features are taken out of the data frame and sent to the classification agent as input.

- *Setting Reward Function*
  The act() method of the DQN class contains a definition of the reward. If any of the true class labels in the y variable match the predicted action (class label), the agent is rewarded with a value of 1. If not, it is rewarded with -1. The goal of the reward system is to encourage the agent to forecast classes correctly.

- *Selection of Action*
  The action selection method makes sure that the model chooses the best class label depending on the situation at hand and previously learned information.
  The class labels that are available in this situation make up the action space. To determine the class for a particular input, the agent will select an action (class label) from this collection. The number of classes in the classification problem and the size of the action space are related.

- *Training*
  Iterating through episodes and the stages in each episode are both parts of the training process. In agreement with an epsilon-greedy exploration-exploitation strategy, the agent chooses a course of action (class label). In accordance with the accuracy of its forecast, it is rewarded, and the ensemble of decision tree models is updated. For the specified number of episodes, training is ongoing.

- *Evaluation*
  By comparing the predicted labels with the actual labels using the test data, it is feasible to assess how accurate the agent's predictions were. The calculated accuracy of the base model and the accuracy of the DQN-based agent after training are compared.
  The various metrics involved in analysis are accuracy, recall score and precision score. However, the performance of models was also analyzed using a confusion matrix.

IV. RESULTS AND ANALYSIS

4.1 Comparison with Baseline Algorithms

On the chosen benchmark datasets, the performance of the proposed framework, which incorporates Deep Q-Network with multiclass classification algorithms, is compared with that of the baseline algorithms.

Comparative Analysis			
Model	**Accuracy**	**Recall**	**Precision**
Decision Tree	0.98	0.50	0.50
Decision Tree with DQN	0.36	0.28	0.34
Random Forest	0.99	0.50	0.50
Random Forest with DQN	0.35	0.29	0.34
Naive Bayes	0.99	0.75	0.67
Naive Bayes with DQN	0.36	0.28	0.34

Table 4.1: Results with supervised models

4.2 Analysis of Computational Efficiency

- The Decision Tree Classifier baseline model trains faster than the DQN-based model. Because they immediately learn decision boundaries and feature splits, decision trees can be trained quickly without iterative optimisation. The DQN-based model trains an ensemble of Decision Tree Classifiers periodically, which is more computationally expensive.
- The baseline model (Random Forest Classifier) takes less training time than the DQN-based model. Random forests are successful because they may construct many decision trees at once utilising parallel computing. Individual decision trees are trained using a random collection of characteristics and data samples. The DQN-based approach requires training for an ensemble of Random Forest Classifiers, which can be computationally intensive.
- Basic Gaussian Naive Bayes is computationally effective for training and prediction due to its simplicity. The ensemble DQN model uses Naive Bayes classifiers, however, it is more sophisticated and requires more processing.
- In conclusion, compared to the DQN-based model, the baseline models are anticipated to be computationally more efficient in terms of training time, inference time, and memory use.
- The intrinsic intricacy of the DQN architecture is the cause of this efficiency disparity. Due to the repetitive nature of reinforcement learning, DQN combines deep reinforcement learning and sequential decision-making, which frequently necessitates longer training cycles. Additionally, slower convergence rates could be a result of the exploration-exploitation trade-off in DQN.

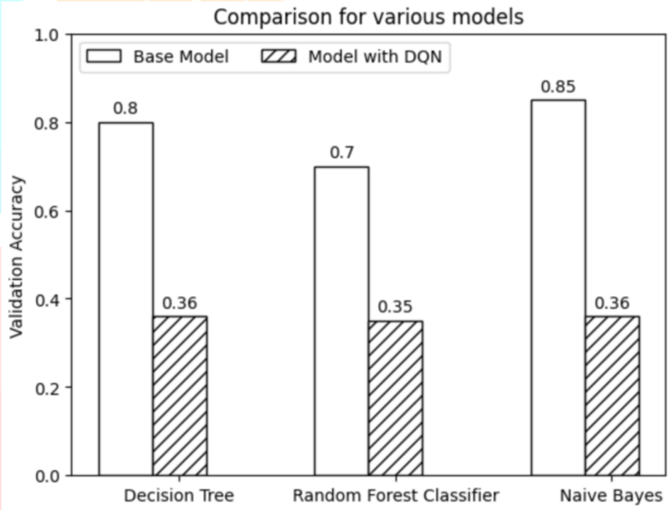

Fig. 2. Comparative analysis for accuracy

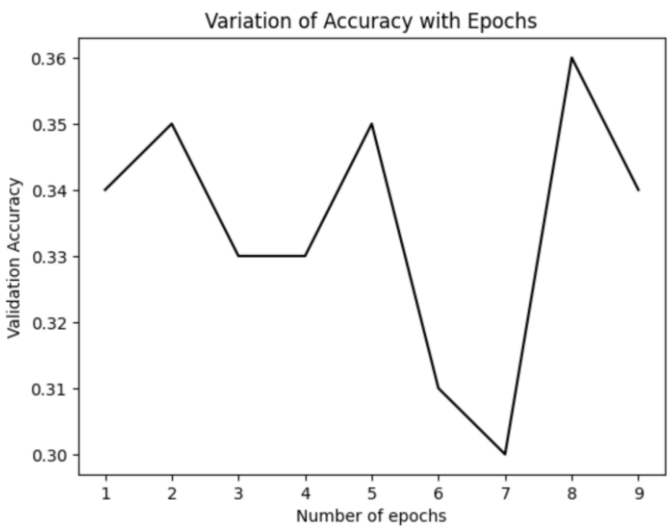

Fig. 3. Variation of accuracy with epochs for Decision Tree - DQN

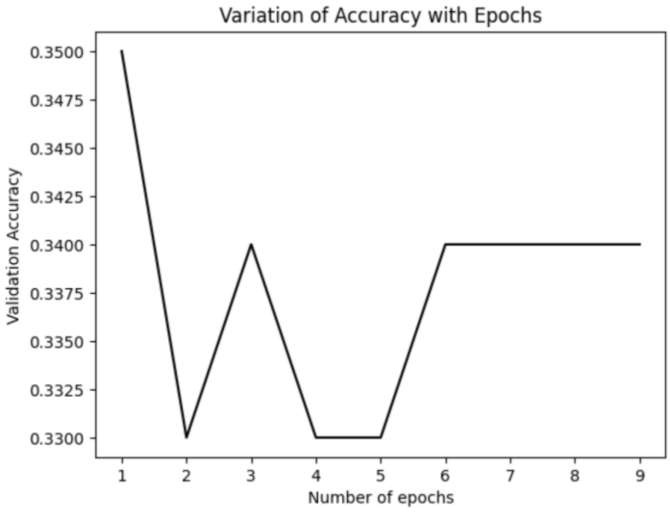

Fig. 4. Variation of accuracy with epochs for Random Forest Classifier with DQN

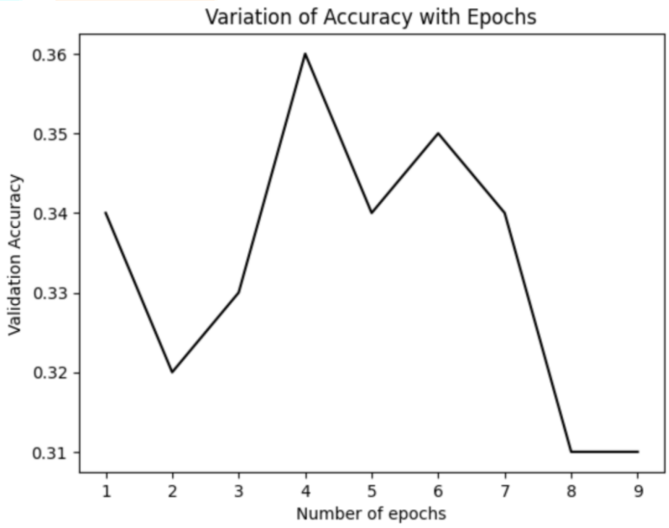

Fig. 5. Variation of accuracy with epochs for Naive Bayes with DQN

V. DISCUSSION

5.1 Advantages

Multiclass classification methods that incorporate Deep Q-Network (DQN) have various benefits and provide special capabilities to the task. Benefits involve:

- *Handling complex decision-making*
  The Q-function captures complicated choice settings in complex environments by giving greater values to state-action combinations that result in better outcomes. DQNs can now manoeuvre through complex scenarios with a wide range of potential actions and states thanks to this. Through successive improvements that reduce the discrepancy between the predicted Q-values and the true rewards received through contact with the environment, the DQN learns the Q-function.
- *Adaptability to dynamic environments*

The DQN adjusts its Q-function as the agent learns and interacts with the environment. This lets the DQN adapt its decision-making technique to external changes, guaranteeing it can still make successful decisions.

- *Handling imbalanced datasets*
  The uneven distribution of various classes or states is referred to as an imbalanced dataset. This might happen in the setting of DQNs when some states or behaviours are observed more frequently than others. Due to the DQN's potential limited exposure to specific state-action combinations, imbalanced datasets can cause biases in the estimate of the Q-function. Techniques like experience replay and prioritised replay can be used to remedy this. By ensuring that the DQN samples experience various states and actions consistently, these techniques lessen the negative effects of imbalanced datasets on learning.
- *Real-time classification*
  Real-time categorization requires rapid, effective action in situations when timeliness is critical. By adopting methods that give consideration to trade-offs between exploitation and exploration, DQNs can be trained to take decisions in the present. Epsilon-greedy exploration is one such technique that aids the agent in striking a balance between selecting activities that are known to be beneficial (exploitation) and investigating novel behaviours to find potential enhancements. DQNs may take decisions rapidly while being able to learn from fresh experiences by striking this equilibrium.

5.2 Limitations

- *Large Memory Requirements*
  Especially when employing experience replay, which includes storing and sampling from a significant replay buffer, DQN often needs a lot of RAM.
- *Curse of Dimensionality*
  Finding the most effective measures and achieving efficient convergence can be more difficult when the DQN training and learning process is impacted by the curse of dimensionality. Consequently, DQN's ability to do multiclass classification well may be constrained by its ability to handle significant feature spaces.
- *Limited Generalization to New Classes*
  It often acquires policies unique to the classes found in the training set. They are efficient at handling well-known classes, but they have a limited ability to generalise to unfamiliar or new classes. In dynamic classification contexts where new classes continually emerge, the technique is less adaptive since incorporating new classes into the model often requires retraining or considerable fine-tuning.

5.3 Future scope

Future prospects are promising when Deep Q-Network is incorporated into multiclass classification algorithms. Transfer Learning and Knowledge Transfer, Real-time Classification, Hierarchical Multiclass Classification, Adaptive Learning and Dynamic Feature Selection, and many others can benefit from this.

VI. CONCLUSION

The study uses multiclass classification to show the significance of using DQN for financial distress prediction in businesses. The study's findings may help businesses, investors, and financial institutions make informed decisions and take preventive action to reduce the risks associated with the financial crisis.

Possible reasons for less accuracy by the DQN model than the base model:

- The classifier for the base model is trained directly on the labelled training data using a traditional supervised learning methodology. In a single step, it learns the probability distributions and class boundaries from the data. While the DQN model iteratively changes its ensemble of classifiers based on the rewards it receives from the environment, it is trained using a reinforcement learning methodology. This recurrent training procedure may generate noise and instability, resulting in less accurate convergence.
- While lowering bias and variance can help ensembles perform better, they also add to the complexity and risk of inconsistencies across the various models. Lower accuracy may be the consequence if the ensemble is unable to fully capture the underlying patterns and relationships.

References

[1] Melrose Roderick, James MacGlashan, Stefanie Tellex "Implementing the Deep Q-Network" arXiv:1711.07478v1 [cs.LG] 20 Nov 2017

[2] Jianqing Fan, Zhaoran Wang, Yuchen Xie, Zhuoran Yang Proceedings of the 2nd Conference on Learning for Dynamics and Control, PMLR 120:486-489, 2020

[3] Z. Gao, Y. Gao, Y. Hu, Z. Jiang and J. Su, "Application of Deep Q-Network in Portfolio Management," 2020 5th IEEE International Conference on Big Data Analytics (ICBDA), Xiamen, China, 2020, pp. 268-275, doi: 10.1109/ICBDA49040.2020.9101333.

[4] Mills P. Solving for multi-class: a survey and synthesis. arXiv preprint arXiv:1809.05929. 2018 Sep 16.

[5] Wen G, Wu K. Building decision tree for imbalanced classification via deep reinforcement learning. Asian Conference on Machine Learning 2021 Nov 28 (pp. 1645-1659). PMLR.

[6] Fu Q, Li K, Chen J, Wang J, Lu Y, Wang Y. Building energy consumption prediction using a deep-forest-based DQN method. Buildings. 2022 Jan 27;12(2):131.

[7] Reddy EM, Gurrala A, Hasitha VB, Kumar KV. Introduction to Naive Bayes and a Review on Its Subtypes with Applications. Bayesian Reason. Gaussian Process. Mach. Learn. Appl. 2022 Apr 19:1-4.

[8] Whitaker RB. The early stages of financial distress. Journal of Economics and Finance. 1999 Jun;23(2):123-32.

[9] Lau AH. A five-state financial distress prediction model. Journal of Accounting Research. 1987 Apr 1:127-38.

[10] Mselmi N, Lahiani A, Hamza T. Financial distress prediction: The case of French small and medium-sized firms. International Review of Financial Analysis. 2017 Mar 1;50:67-80.

[11] Grandini M, Bagli E, Visani G. Metrics for multi-class classification: an overview. arXiv preprint arXiv:2008.05756. 2020 Aug 13.

[12] Toupas P, Chamou D, Giannoutakis KM, Drosou A, Tzovaras D. An intrusion detection system for multi-class classification based on deep neural networks. In2019 18th IEEE International Conference on machine learning and Applications (ICMLA) 2019 Dec 16 (pp. 1253-1258). IEEE.

[13] Li J, Liu Y, Yin R, Zhang H, Ding L, Wang W. Multi-class learning: From theory to algorithm. Advances in Neural Information Processing Systems. 2018;31.