Automatic Fraud Detection in e-Commerce Transactions using Deep Reinforcement Learning and Artificial Neural Networks

Yuanyuan Tang*
School of Economics and Management, Tianjin Vocational Institute, Tianjin 300410, China

***Abstract*—Fraud is a serious issue that has plagued e-commerce for many years, and despite significant efforts to combat it, current fraud detection strategies only catch a small portion of fraudulent transactions. This results in substantial financial losses, with billions of dollars being lost each year. Given the expected surge in the volume of online transactions in the upcoming years, there is a critical need for improved fraud detection strategies. To tackle this problem, the article proposes a deep reinforcement learning approach for the automatic detection of fraudulent e-commerce transactions. The architecture's policy is built on the implementation of artificial neural networks (ANNs). The classification problem is viewed as a step-by-step decision-making procedure. The implementation of the model involves the use of the artificial bee colony (ABC) algorithm to acquire initial weight values. After that, in each step, the agent obtains a sample and performs a classification, with the environment providing a reward for each classification action. To encourage the model to concentrate on detecting fraudulent transactions precisely, the reward for identifying the minority class is higher than that for the majority class. With the aid of a supportive learning setting and a specific reward system, the agent ultimately determines the best approach to achieve its objectives. The performance of the suggested model is assessed utilizing a publicly available dataset contributed by the Machine Learning group at the Université Libre de Bruxelles. The experimental outcomes, determined using recognized evaluation measures, indicate that the model has attained a high level of accuracy. As a result, the suggested model is considered appropriate for identifying deceitful transactions in e-commerce.**

Keywords*—Fraud detection; reinforcement learning; artificial neural network; artificial bee colony; imbalanced classification

I. Introduction

Fraud refers to intentional dishonesty or deception by an individual or group of people with the aim of obtaining financial benefits. As a result of the increase in online transactions such as shopping and insurance claims, there is a new level of fraudulent activity that individuals and businesses must be wary of. Reports indicate that the increase in fraudulent activities in e-commerce transactions during the first quarter of 2018 was significantly higher than the growth rate of e-commerce transactions in 2016. The e-Commerce Fraud Index revealed that in 2017, account takeover fraud in online department stores rose to 0.23%, a significant increase from 0.06% in 2016 and accounting for over 10% of fraud losses. While credit card fraud only makes up 0.1% of all card transactions, fraudulent transactions involving large sums of money have led to significant financial losses. However, even with the surge in credit card transactions in recent times, the ratio of fraud cases has not changed [1].

Numerous institutions and industries have invested significant resources in developing efficient techniques to combat fraudulent activities by leveraging advanced technologies, especially machine learning [2, 3]. As a result of these endeavors, a plethora of solutions have been developed that can differentiate between valid credit card transactions and those that are fraudulent without human intervention. Irrespective of the method employed, there are certain shared issues that can hinder its effectiveness. The prevalent issue is the imbalanced distribution of training data, a feature of past transactions. This creates various challenges, such as overfitting, and results in low-accuracy classifiers used. Imbalanced classification is a common challenge in machine learning, where one class has significantly more data compared to other classes. Due to this disproportion, recognizing minority specimens becomes difficult because of their infrequency and randomness, resulting in a poorer outcome [4-6]. The problem of imbalanced classification in machine learning can be tackled using one of two methods, namely the data level and algorithmic level [7, 8]. At the data level, balancing the class distribution can be achieved through various methods like oversampling and under-sampling. Oversampling generates new samples by linear interpolation between adjacent minority samples, while under-sampling removes some majority examples utilizing the nearest neighbor algorithm [9]. However, these methods risk overfitting and loss of significant information [10, 11]. At the algorithmic level, techniques such as cost-sensitive learning, decision threshold adjustment, and ensemble learning can be applied to increase the significance of the minority class. Ensemble learning trains multiple sub-classifiers and combines them to improve performance, while cost-sensitive learning assigns varying costs to incorrect classification, with the misclassification of the minority class given a higher cost. Threshold adjustment techniques change the decision-making threshold during testing. Some proposed deep models for imbalanced data classification utilize innovative loss functions that consider errors in classifying minority and majority groups [12]. These techniques can retain the specific attributes of a dataset that has imbalanced classes or clusters while also safeguarding the margins between samples from different classes or clusters [5, 13, 14].

While traditional machine learning approaches use a rigid feature extraction strategy that often leads to poor generalization ability, long processing time, and low accuracy, deep learning algorithms have emerged as promising alternative for classification tasks [8]. With their layered structure, deep learning algorithms can capture complex patterns and relationships within data, making them highly adept at learning high-level features. One such popular algorithm is the Multilayer Perceptron (MLP), a universal approximation that excels at handling nonlinear problems. Originally developed to solve the XOR problem, MLP has since been successfully applied to various combinatorial optimization problems, including pattern recognition, classification, image processing, and linear and nonlinear optimization. MLP processes input signals by passing them through interconnected layers of processing nodes, each of which receives a set of input values, sums them, and then applies an activation function to determine its output, mimicking the behavior of a human neuron. MLP, through its layers of interconnected nodes, is capable of acquiring intricate associations between the input and output variables. Additionally, the lack of interconnections between nodes in the same layer helps reduce computational complexity, making processing more efficient [15].

The paper presents a novel approach to detecting fraud in e-commerce transactions by combining the ABC algorithm [16] and reinforcement learning. The model being proposed views the classification task as a process of making sequential decisions, which is analogous to playing a game of guessing. In this game, the agent is presented with a training instance and has to classify it using a policy. The agent's performance is evaluated based on the rewards received for correct and incorrect classifications, with a higher reward assigned to correctly identifying the minority class. The main goal of the agent is to achieve the highest possible cumulative reward by correctly identifying the largest number of samples. This technique has the potential to overcome the challenges of imbalanced classification discussed earlier in the paper. The suggested model frames the classification issue as a series of sequential decisions, enabling the agent to acquire knowledge and adjust its strategy according to responses from the surroundings. The use of reinforcement learning further enhances the agent's ability to explore and exploit the search space efficiently. This article makes three significant contributions. 1) The approach taken to address the challenge of imbalanced classification is to view the prediction problem as a sequential decision-making process and a reinforcement learning-based algorithm is introduced. 2) An encoding method based on the ABC algorithm was devised to obtain the best initial value instead of assigning weights randomly, and 3) the proposed model's performance was evaluated through experiments, and a comparison was made between this model and other methods that use random weight initialization, which encounter challenges in dealing with imbalanced classification.

The format of the article is organized in the following manner. In the second section, the paper presents an overview of existing research in the relevant area. In the third section, a succinct explanation of the ABC algorithm and its functioning is presented. In the fourth section, the model proposed in the study is introduced, and in the fifth section, the evaluation criteria, dataset, and analysis of the results are presented. The concluding section of the paper discusses the study's findings and draws conclusions, as well as outlining potential avenues for future research.

II. Related Work

Initially, fraud detection was associated with the utilization of Information Retrieval or the Rule-based method. The details of each transaction were scrutinized by hand, and decisions were made concerning fraudulence or reliability based on strict criteria. Each transaction causes the development of a feature vector [2]. A feature vector is composed of various attributes and parameters such as Transaction identifier, Transaction amount, Cardholder data, Site of the transaction, and Time of transaction. This vector is assigned points according to the scoring criteria determined by human investigators. As an example, if a transaction has taken place on another continent within an hour, then the fraud score would be 0.95 [17]. This system depends on the addition of more and more regulations in order to remain one step ahead of scammers that seek to exploit and bypass current rules.

Big Data Analytics, through the use of machine learning, is more wide-reaching, economical, precise, and automated [18, 19]. This powerful combination of Big Data and machine learning has opened up new possibilities for businesses and organizations across various industries, enabling them to extract valuable insights, make data-driven decisions, and optimize their operations like never before. One of the remarkable applications of Big Data Analytics is the construction of sophisticated models capable of forecasting, classifying, or estimating the authenticity of transactions, especially when it comes to identifying fraudulent activities [20]. Such models, empowered by the wealth of data collected from digital datasets containing numerous transactions, have revolutionized fraud detection methodologies. By leveraging machine learning algorithms and tapping into large and diverse datasets, these data-informed models have achieved impressive results in accurately differentiating between genuine and fraudulent transactions. The abundance of data provides these models with a rich source of information, enabling them to discern complex patterns and anomalies that would be challenging for traditional rule-based systems to detect. The training process of these models involves exposing them to vast quantities of labeled data, where each transaction is tagged as either authentic or fraudulent. Through iterative learning and optimization, the models adapt and fine-tune their parameters, continuously improving their performance and generalization abilities. Different data-driven models have emerged in the realm of Big Data Analytics, each employing a variety of methods and algorithms tailored to specific use cases and data characteristics. These models can include traditional machine learning approaches like SVM, Random Forest, and Gradient Boosting, as well as state-of-the-art deep learning techniques like Neural Networks and Transformer-based models. The choice of model and method depends on the nature of the data, the complexity of the problem, and the desired level of interpretability [21, 22].

The commonly utilized approach involves using machine learning techniques to create a model based on the data. This data-driven model is generally more versatile and dependable, enabling it to achieve high accuracy levels, often reaching up to 87% or even higher, depending on the specific problem and dataset. The success of machine learning models can be attributed to their ability to uncover complex patterns and relationships within the data that might not be easily discernible through traditional rule-based systems. Among the well-known machine learning algorithms, several have proven to be highly effective in various domains. K-means is a popular clustering algorithm used for grouping data points into clusters based on their similarity, making it useful for segmentation and pattern discovery tasks. Regression Analysis, on the other hand, is widely employed for predicting numerical values and understanding the relationships between variables. SVM have been extensively utilized in both classification and regression tasks. SVM is particularly suitable for binary classification problems, and with appropriate kernel functions, it can handle complex decision boundaries efficiently. Similarly, Random Forest and Decision Trees are powerful ensemble methods that can be applied to both classification and regression tasks, providing robustness and reducing overfitting [23]. Recent advancements in deep learning have introduced breakthroughs in the field of fraud prevention and detection. Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) have demonstrated significant potential in carrying out diverse prediction and classification tasks, including those related to fraud detection. RNNs are well-suited for sequential data, making them ideal for processing time-series data or textual data with temporal dependencies. On the other hand, CNNs excel at processing grid-like data, such as images and other structured representations, and are capable of automatically learning relevant features from the input. When applied to fraud prevention and detection, RNNs can effectively capture temporal patterns in transaction histories and user behaviors, enabling them to identify unusual or fraudulent activities. CNNs, on the other hand, can be employed for tasks like image-based fraud detection, where they can learn to recognize visual patterns associated with fraudulent behavior [24].

The approach of supervised machine learning involves first training the learning algorithm with labeled data and then evaluating its accuracy on a test set [25]. Labeled data is a prerequisite for using the supervised learning method to train a classifier. This labeling procedure is both time-consuming and costly. Various classifiers, such as one-class SVM [26], decision tree [27, 28], random forest [29, 30], and logistic regression [8], have shown a good level of accuracy in their performance. SVM can be used for solving both regression and classification problems [15]. One-class SVM is especially useful in scenarios where the data distribution is imbalanced, which is similar to the particular issue. The system gains the ability to deduce the characteristics of the dominant class while simultaneously being able to identify anomalies or the less prevalent class. Decision trees are structures resembling flow charts that enable the classification of input data points or the prediction of output based on an input. The Random Forest technique involves a strong methodology that incorporates numerous decision trees, which are subsequently combined to produce their outputs.

Unsupervised learning has been widely adopted in various domains and finance due to the flexibility and cost-effectiveness it offers [31]. In contrast to supervised learning, unsupervised learning methods can derive insights from a dataset without the need for labeled data. This makes it a useful tool in situations where data labeling is expensive or impractical, especially when dealing with large datasets. With the growth of big data, unsupervised learning has become increasingly important as it can help us to identify patterns, anomalies, and hidden structures in large datasets that may not be easily noticeable through manual inspection. Nearest neighbor, clustering, and outlier detection are three commonly used unsupervised learning algorithms for fraud detection [32-35]. The nearest neighbor algorithm determines the authenticity of a transaction by measuring the distance between it and its nearest neighbor in the dataset [36]. This allows for the identification of data points that are considered to be abnormal or fraudulent [37]. Clustering algorithms group similar data points together, which is particularly useful for identifying groups of transactions that exhibit similar behavior or characteristics. Peer group analysis is one application of clustering algorithms used in fraud detection [38, 39]. Outlier detection algorithms aim to locate data points that deviate from the norm in a dataset, which can be valuable in detecting fraudulent transactions [1, 40]. Credit fraud detection often deals with imbalanced data, where the number of genuine transactions is significantly larger than the number of fraudulent transactions. This can result in learning algorithms underperforming as they tend to prioritize accuracy on the majority class. Therefore, resampling methods such as oversampling or undersampling need to be used to balance the data before training the learning algorithm.

Oversampling pertains to generating artificial data points for the underrepresented class, whereas undersampling entails reducing the number of data points in the overrepresented class [12]. Careful consideration must be given to the choice of resampling method, as oversampling can lead to overfitting while undersampling can lead to loss of information. One challenge in fraud detection is the delay caused by the need for human investigators to label transactions before they can be used for training the algorithm. This delay is known as verification latency and can be reduced by automating the labeling process through the use of semi-supervised or active learning approaches [41]. Oversampling pertains to generating artificial data points for the underrepresented class, whereas undersampling entails reducing the number of data points in the overrepresented class [42]. Active learning is a method that entails selecting the most informative data points for labeling in an iterative process, which can decrease the required amount of labeled data [43, 44]. Lastly, it is essential to consider the issues of concept change over time and biased sample selection when creating a machine learning algorithm for credit fraud analysis [45]. Concept drift refers to the tendency of transaction behavior to change over time, which can lead to the algorithm becoming outdated and inaccurate [46]. Sample selection bias occurs when the distribution of data used for

training and testing the algorithm is different, which can lead to the algorithm performing poorly on unseen data [47].

III. Artificial Bee Colony Algorithm

ABC algorithm is a type of optimization algorithm that draws inspiration from the foraging behavior of honeybee colonies [48, 49]. The procedure emulates the nourishment-gathering demeanor of honeybees and employs a populace-centered strategy to explore the supreme resolution to a specified enhancement predicament. In ABC, the populace of bees is segregated into three factions: occupied bees, spectator bees, and scout bees. The assignment of the occupied bees is to probe the resolution expanse and unearth advantageous resolutions. The onlooker bees assess the solutions discovered by the employed bees according to their quality and conduct additional evaluations. The scout bees explore new regions of the search space that have not been explored by the employed and onlooker bees. ABC is based on the idea of random search, where candidate solutions are generated randomly in the search space. The quality of the solutions is evaluated using a fitness function that measures how well the solution performs on the optimization problem. ABC iteratively generates new candidate solutions by modifying the existing solutions based on the foraging behavior of honeybees. ABC has been successfully applied to various optimization issues in different fields, including engineering, economics, and bioinformatics. It has been shown to be efficient and effective in finding optimal or near-optimal solutions in many real-world optimization issues. The optimization process of ABC is summarized below:

1) Initialization: The algorithm starts by randomly generating an initial population of candidate solutions (employed bees) within the search space.

2) Employed bee phase: Each employed bee independently explores the search space by modifying its solution using a neighborhood search algorithm. The new solution is evaluated using a fitness function and compared to the current solution. If the new solution is better, it replaces the current solution. This process is repeated for all employed bees.

3) Onlooker bee phase: The employed bees communicate with the onlooker bees by performing a waggle dance to indicate the quality of their solutions. The onlooker bees select the solutions based on the quality of the dance and evaluate them using the fitness function. The onlooker bees choose the solutions with higher fitness values and use them for further exploration.

4) Scout bee phase: Some of the employed and onlooker bees become scout bees with a small probability. These scout bees randomly explore new solutions in the search space that have not been explored by the other bees.

5) Update: The algorithm updates the population by replacing the worst solutions with new solutions generated by the employed and scout bees. The algorithm terminates when a stopping criterion is met, such as reaching a maximum number of iterations or a satisfactory solution is found.

6) Output: The output of the optimization process is the best solution found by the algorithm.

IV. Model Architecture

According to Fig. 1, the proposed model encompasses ABC and RL for fraud detection.

A. Pre-training

In this stage, the suggested algorithm incorporates a mutual learning-based ABC technique to initialize the weights of the MLP. The organization of these weights into a vector mirrors the bees' positions in the ABC algorithm. This process of converting the weights into a vector is commonly referred to as encoding. Finding the optimal layout for this encoding presents a challenging task, but researchers have diligently conducted numerous experiments to develop the most effective encoding strategy. As shown in Fig. 2, all the bias terms and weights within the MLP are carefully arranged into a vector, essentially forming a potential solution within the ABC algorithm. Each element of this vector corresponds to a specific weight or bias term in the neural network. By treating the vector as a candidate solution, the ABC algorithm can explore and refine its position in the solution space through the process of iterative optimization. The mutual learning-based approach in the ABC algorithm plays a crucial role in this phase. As bees in the colony share information and collectively learn from one another, the weights in the vector evolve based on the knowledge gathered from different bees' experiences. This collaboration allows the ABC algorithm to efficiently navigate the vast solution space and converge towards more promising weight configurations. The process of encoding and using a vector representation of weights provides several advantages. Firstly, it enables a seamless integration of the ABC algorithm with the neural network training process, effectively initializing the model for further optimization. Additionally, the vector-based representation facilitates the implementation of various search and optimization strategies, making it easier to explore and exploit different regions of the solution space. With the weights and bias terms encoded in a vector, the ABC algorithm operates in a population-based manner, emulating the collective intelligence of bees in a real colony. The population of candidate solutions evolves over iterations, and through the exploration and exploitation of different weight configurations, the algorithm progressively refines the MLP's parameters, ultimately leading to improved performance and convergence towards better solutions.

The quality of a candidate solution is determined by defining the fitness function as

$$Fitness = \frac{1}{\sum_{i=1}^{N}(y_i - \tilde{y}_i)^2} \quad (1)$$

where N represents the total number of training samples, with y_i and $\tilde{y}_i$ denoting the i-th target and model-predicted output, respectively.

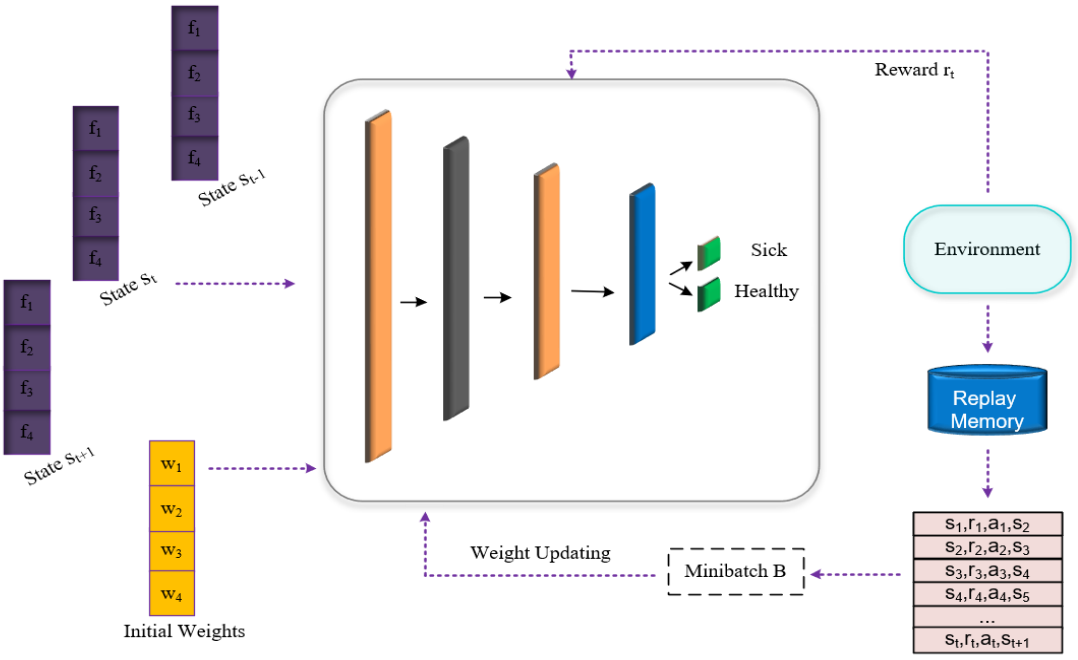

Fig. 1. An outline of the suggested approach.

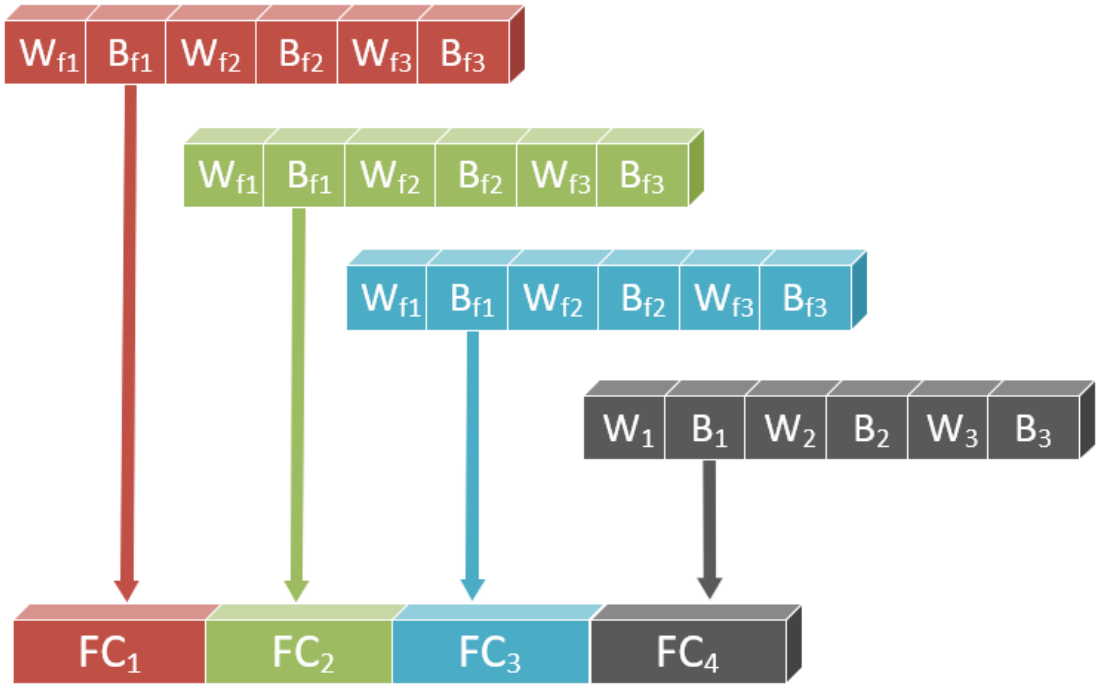

Fig. 2. The method of encoding used in the suggested algorithm.

B. Classification

The issue of imbalanced classification arises when there is a significant difference in data volume between the two classes. To tackle this challenge, we have employed a sequential decision-making process utilizing an RL approach. The RL approach involves an agent striving to maximize its score through effective decision-making and actions within the environment, eventually leading to the discovery of an optimal policy. In the proposed framework, at every time step, the actor acquires an exemplar from the collection and executes a categorization duty. Afterward, the actor obtains prompt responsiveness from the milieu, where a precise categorization yields an affirmative grade, whereas an erroneous one produces a pessimistic grade. This feedback mechanism serves to guide the agent towards making more informed decisions and improving its performance over time. The RL algorithm plays a central role in the approach as it seeks to achieve the optimal policy by maximizing the cumulative rewards obtained throughout the decision-making process. The goal is to find the

most favorable strategy that results in the highest rewards and, ultimately, the best classification performance. To further illustrate the intended configurations, we utilize a dataset containing N samples, each with corresponding labels. These samples are represented as $D = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_N, y_N)\}$, where x_i denotes the i-th image, and y_i represents its corresponding label. These configurations are vital for setting up the environment in which the RL agent operates, guiding it towards achieving optimal classification results through the decision-making process. The following describes the intended configurations:

- Policy π_θ : Policy π is a mapping function that associates states (S) with actions (A). In this context, $\pi_\theta(s_t)$ denotes the action taken in a specific state s_t. The method employing the classifier with weights θ is denoted as π_θ.
- State s_t: An instance x_t extracted from the dataset, D, is associated with a corresponding state s_t. The first data point x_1 represents the initial state s_1. To avoid the model from learning a fixed sequence, D is randomized in each episode.
- Action a_t: The action a_t is performed to predict the label x_t, with binary classification, and the available choices for a_t are limited to either 0 or 1. In this context, the minority class is represented by 0, while the majority class is denoted by 1.
- Reward r_t : The reward is contingent on the outcome of the action performed. Upon performing the correct classification, the agent receives a positive reward, whereas an incorrect classification results in a negative reward. The bonus value needs to vary for each class. Appropriately calibrated rewards can significantly improve the model's performance by ensuring that the reward level corresponds to the action taken. In this study, the reward for an action is defined using the following formula:

$$r_t(s_t, a_t, y_t) = \begin{cases} +1\,, a_t = y_t \; and \; s_t \in D_S \\ -1\,, a_t \neq y_t \; and \; s_t \in D_S \\ \lambda\,, a_t = y_t \; and \; s_t \in D_H \\ -\lambda\,, a_t \neq y_t \; and \; s_t \in D_H \end{cases} \quad (2)$$

where D_S , and D_H represent the majority (“sick”) and minority (“healthy”) classes, respectively. Correctly/incorrectly classifying a sample from the majority class yields a reward of $+\lambda/-\lambda$, where $0 < \lambda < 1$.

- Terminal E: In every instructional session, the teaching procedure concludes at diverse concluding conditions. A progression of situation-action duets $\{(s_1, a_1, y_1), (s_2, a_2, y_2), (s_3, a_3, y_3), \ldots, (s_t, a_t, y_t)\}$ from a starting situation to an ultimate situation is denoted as an instructional session. In the circumstance, the culmination of an occurrence is ascertained by either categorizing all the instruction data or by inaccurately categorizing a specimen from the underrepresented class.
- Transition probability P: The agent transitions to the next state, s_{t+1}, from the current state, s_t, based on the sequential order of the read data. The probability of transitioning to state s_{t+1} from state s_t, given the action a_t, is denoted as $p(s_{t+1}|s_t, a_t)$.

V. Experimental Results

The dataset employed in the project is publicly accessible and was provided by the Machine Learning group of Université Libre de Bruxelles [50]. The data used in this study comprises credit card transactions made by cardholders in Europe during September 2013. This particular dataset consists of 284,807 transactions made by European cardholders over a span of two days, with 492 of them identified as fraudulent. The dataset is characterized by a significant class imbalance, where the number of positive cases (fraudulent transactions) constitutes only 0.172% of the total transactions. The dataset comprises solely numerical input variables, which are the product of a PCA transformation. The original features and additional contextual details about the data were not released to the public due to confidentiality concerns. The attributes V1 to V28 are the primary features produced through PCA, while 'Time' and 'Amount' are the only characteristics that have not been subjected to PCA transformation. The attribute 'Time' represents' the time interval in seconds between a particular transaction and the initial transaction registered in the dataset. The 'Amount' characteristic pertains to the amount of the transaction and can be applied to tasks such as cost-sensitive learning, which depend on the transaction value. The variable 'Class' serves as the output or target variable and is assigned the value of 1 if the transaction is a fraud and 0 if it is not. The dataset of credit card transactions may also include summarized characteristics. Various summarized characteristics can be extracted from the credit card transaction dataset, such as the average monthly transaction amount per cardholder, the average number of transactions per month, the average monthly spending on fuel, the time and distance between the present and previous transactions, and others.

In this article, we have incorporated a batch normalization layer to ensure data normalization and facilitate smoother training. By processing the data before feeding it into the MLP, the batch normalization layer effectively addresses the issue of internal covariate shift, allowing the network to learn more effectively and expedite convergence. To introduce non-linearity and enhance the network's capacity to model complex relationships within the data, we apply the ReLU (Rectified Linear Unit) activation function between the layers. Regarding the training configuration, we have carefully chosen a batch size of 32. This decision is made to strike a balance between computational efficiency and gradient accuracy during the optimization process. The batch size of 32 enables efficient parallel processing on modern hardware while retaining sufficient samples to ensure a stable gradient estimation during backpropagation.

The proposed approach was subjected to a rigorous evaluation by comparing it with six different machine learning models. These models included SVM [51], Naïve Bayes [52], KNN [53], Random forests [54], Logistic Regression [55], and Decision tree [56], which are all popular and widely used in the

field of machine learning (See Table I). In addition, the performance of two smaller versions of the proposed approach was also tested. These versions were designed to use random weights and RL for classification. For evaluating the outcomes of the approach, standard performance metrics, including F-measure and geometric mean, known to be dependable for assessing imbalanced data, were employed. The approach outperformed all other models across all evaluation criteria, surpassing even the top-performing model, Decision tree. Specifically, the approach achieved a reduction in the error rate by more than 65% and 28% in F-measure and G-means, respectively. Additionally, a comparison was conducted between the performance of the approach and the smaller versions, namely Proposed (random weights) and Proposed (random weights and RL). This comparison revealed that the approach significantly reduced the error rate by approximately 72%. These results highlight the importance and effectiveness of the improved artificial bee colony and RL techniques utilized in the suggested approach.

In the next experiment, the objective is to evaluate ABC against various metaheuristic optimization algorithms. To achieve this, different metaheuristics are utilized to obtain the initial model parameters while keeping other model components constant. The six algorithms used in this experiment are HMS [57], FA [58], BA [59], DE [60], GWO [61], and COA [62]. For all algorithms, default settings have been used (Table II). Table III presents the results obtained from this comparison. The results indicate that the proposed ABC approach outperforms other algorithms in terms of error reduction, with a decrease of approximately 33% compared to the second-best algorithm, HMS.

TABLE I. RESULTS OF VARIOUS CLASSIFICATION ALGORITHMS

	accuracy	recall	precision	F-measure	G-means
Naïve Bayes	0.695 ± 0.160	0.610 ± 0.180	0.560 ± 0.100	0.580 ± 0.041	0.695 ± 0.160
Random forests	0.705 ± 0.015	0.580 ± 0.035	0.570 ± 0.107	0.580 ± 0.035	0.705 ± 0.015
SVM	0.825 ± 0.165	0.790 ± 0.025	0.730 ± 0.000	0.760 ± 0.263	0.825 ± 0.255
KNN	0.800 ± 0.015	0.680 ± 0.078	0.720 ± 0.097	0.700 ± 0.120	0.800 ± 0.000
Decision tree	0.855 ± 0.105	0.840 ± 0.105	0.780 ± 0.485	0.820 ± 0.056	0.855 ± 0.269
Logistic Regression	0.830 ± 0.110	0.750 ± 0.090	0.790 ± 0.059	0.770 ± 0.025	0.830 ± 0.142
Proposed (random weights)	0.810 ± 0.120	0.820 ± 0.140	0.800 ± 0.041	0.790 ± 0.129	0.810 ± 0.012
Proposed (random weights and RL)	0.860 ± 0.005	0.870 ± 0.105	0.850 ± 0.200	0.850 ± 0.012	0.860 ± 0.035
Full model	0.890 ± 0.015	0.910 ± 0.120	0.892 ± 0.0312	0.890 ± 0.003	0.898 ± 0.055

TABLE II. METAHEURISTICS PARAMETER SETTINGS

algorithm	parameter	value
HMS	minimum mental processes	2
	maximum mental processes	5
	C	1
FA	light absorption coefficient	1
	attractiveness at r = 0	0.1
	scaling factor	0.25
BA	constant for loudness update	0.5
	constant for an emission rate update	0.5
	initial pulse emission rate	0.001
DE	scaling factor	0.5
	crossover probability	0.9
COA	discovery rate of alien solutions	0.25

TABLE III. OUTCOMES OF DIFFERENT METAHEURISTIC ALGORITHMS

	accuracy	recall	precision	F-measure	G-means
HMS	0.872±0.058	0.860±0.103	0.874±0.041	0.862±0.008	0.842±0.082
FA	0.861±0.138	0.850±0.093	0.862±0.231	0.849±0.014	0.820±0.021
BA	0.847±0.004	0.835±0.113	0.830±0.251	0.839±0.065	0.800±0.000
DE	0.830±0.014	0.820±0.006	0.821±0.061	0.825±0.145	0.782±0.120
GWO	0.812±0.159	0.792±0.014	0.806±0.261	0.792±0.165	0.761±0.150
COA	0.760±0.140	0.744±0.004	0.760±0.000	0.750±0.211	0.710±0.110

A. Effect of the Reward Function

Rewards of ±1 and ±λ are used in this study to indicate correct/incorrect classifications of the majority and minority classes, respectively. The optimal value of λ is influenced by the proportion of the majority to minority samples, with a lower value expected as the ratio increases. The performance of the suggested model with λ initialized using a set of values ranging from 0 to 1 in increments of 0.1 is evaluated while keeping the bonus for the majority class constant. The results, which demonstrate that a λ value of 0.4 yields the best model performance across all metrics are shown in Fig. 3. When λ = 0, the impact of the majority class is nullified, while a value of 1 result in equal impact from both majority and minority classes. It is important to note that while adjusting λ is necessary to mitigate the effect of the majority class, setting the value too low can negatively impact the overall performance of the model.

B. Investigating the Impact of the Number of Layers in MLP

The number of layers in MLP affects the model complexity, with a higher number of layers resulting in increased complexity that may lead to over-fitting. Conversely, a model with too few layers may not capture important features in the training data. To address these issues, the impact of the number of layers on the proposed approach was evaluated by testing eight values between 1 and 12. Table IV shows the results, which demonstrate a decreasing trend for values from 1 to 8, followed by an increasing trend for values from 8 to 12. Therefore, the optimal number of layers for MLP is 8 to balance the representation of important features and model complexity.

C. Effect of the Loss Function

Traditional methods, such as adjusting data augmentation and the loss function, can be utilized to address data imbalances. The loss function, which can assign more weight to the minority class, is considered particularly important. Various loss functions, including (WCE) [63], balanced cross-entropy (BCE)] [64], Dice loss (DL) [65], Tversky loss (TL) [66], and Combo Loss (CL) [67], were tested to determine their impact on the model. In the BCE and WCE functions, equal weight is assigned to positive and negative examples. The CL function, which assigns less weight to simple examples and more weight to complex ones, is useful for handling unbalanced data. The results presented in Table V show that CL performs better than TL, with a 31% and 42% reduction in error for accuracy and F-measure, respectively. However, RL performs 71% better than the CL function.

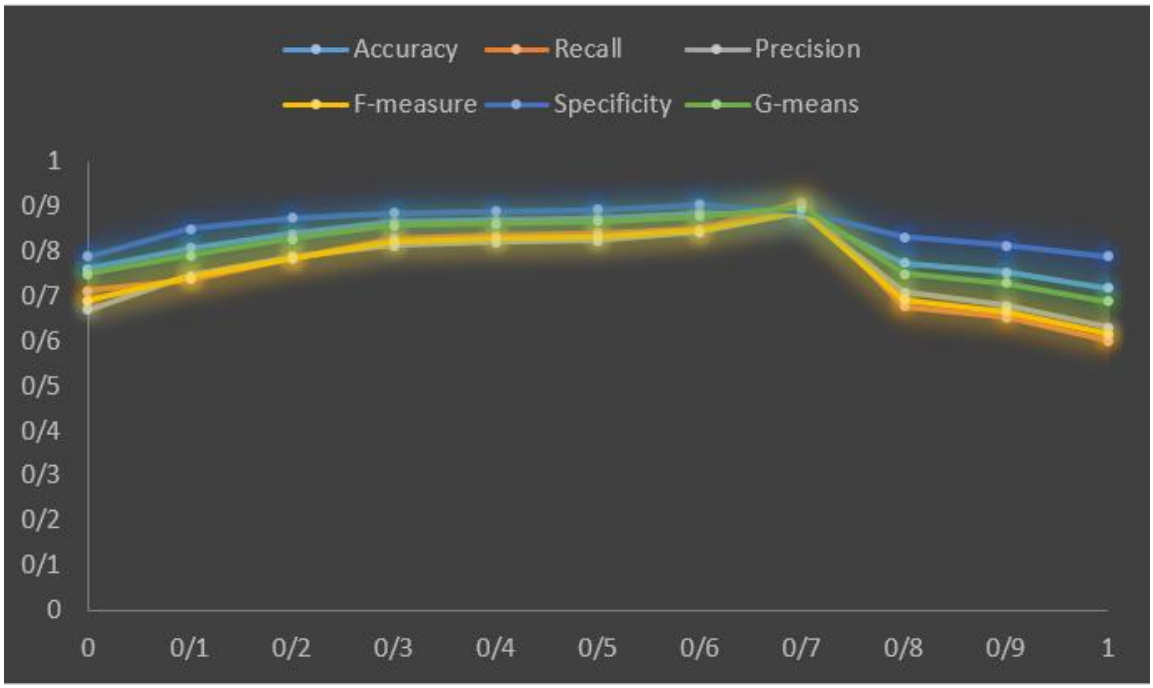

Fig. 3. The performance metrics of the proposed model are graphed against different values of λ in the reward function.

TABLE IV. THE PERFORMANCE METRICS PLOTTED VS. THE DIFFERENT NUMBER OF LAYERS IN MLP

Number of layers	accuracy	recall	precision	F-measure	G-means
1	0.758±0.001	0.792±0.205	0.741±0.017	0.742±0.037	0.773±0.100
2	0.872±0.010	0.883±0.163	0.864±0.007	0.869±0.032	0.883±0.155
4	0.852±0.020	0.861±0.111	0.850±0.233	0.830±0.012	0.862±0.024
8	0.888±0.031	0.905±0.113	0.883±0.023	0.883±0.021	0.905±0.036
10	0.724±0.135	0.740±0.121	0.702±0.211	0.720±0.017	0.670±0.036
12	0.510±0.025	0.612±0.011	0.510±0.043	0.546±0.013	0.440±0.110

TABLE V. RESULTS OF VARIED LOSS FUNCTIONS

	accuracy	recall	precision	F-measure	G-means
WCE	0.765 ± 0.032	0.755 ± 0.016	0.746 ± 0.125	0.751 ± 0.005	0.777 ± 0.038
BCE	0.815 ± 0.027	0.807 ± 0.055	0.786 ± 0.171	0.784 ± 0.016	0.825 ± 0.003
DL	0.826 ± 0.038	0.815 ± 0.031	0.794 ± 0.032	0.812 ± 0.010	0.834 ± 0.002
TL	0.844 ± 0.129	0.838 ± 0.009	0.814 ± 0.021	0.827 ± 0.042	0.857 ± 0.071
CL	0.874 ± 0.006	0.866 ± 0.218	0.854 ± 0.009	0.853 ± 0.053	0.876 ± 0.156

D. Discussion

The article proposed a deep reinforcement learning approach for fraud detection in e-commerce transactions. It acknowledged the serious issue of fraud in e-commerce and the limitations of current detection strategies. The proposed model utilized ANNs and the ABC algorithm to acquire initial weight values. The model viewed fraud detection as a step-by-step decision-making process, with the agent receiving rewards for each classification action. To prioritize detecting fraudulent transactions, the model assigns higher rewards to identifying the minority class.

Fraudsters are known for their adaptability and creativity in devising new strategies to evade detection. As a result, the effectiveness of any fraud detection model, including the proposed deep reinforcement learning approach, hinges on its ability to generalize well to previously unseen fraud patterns. The process of generalization refers to the model's capacity to make accurate predictions on data that differs from the training set, encompassing novel and evolving fraud scenarios. To ensure the robustness of the proposed model, rigorous testing on diverse and evolving fraud scenarios is imperative. Here are some key aspects to consider for assessing the model's generalization capabilities:

- Diverse Testing Datasets: Apart from the publicly available dataset used during model development, it is crucial to evaluate the model on multiple datasets, including those collected from different sources and time periods. Diverse datasets can represent a broader range of fraud patterns and help uncover potential weaknesses in the model's detection capabilities.
- Cross-Domain Evaluation: Fraud patterns may differ across various industries and regions. Evaluating the model's performance on datasets from different domains, such as e-commerce, banking, insurance, etc., helps assess its ability to handle variations in fraud characteristics and attack vectors.
- Time-based Evaluation: Fraud patterns evolve over time, necessitating the model's ability to adapt to emerging fraud tactics. Testing the model on data from different time periods can reveal its responsiveness to temporal changes in fraud behavior.
- Transfer Learning: Applying transfer learning techniques allows leveraging knowledge gained from one dataset to improve performance on another. Pre-trained models can serve as a starting point for fine-tuning on specific fraud detection tasks, potentially enhancing the model's generalization capacity.
- Data Augmentation: To expose the model to a wider array of fraud patterns, data augmentation techniques can be employed. Synthetic fraud scenarios can be generated by perturbing existing data or by using generative models, thus enriching the training data and improving generalization.
- Continuous Monitoring and Feedback: Real-world deployment of the model demands continuous monitoring of its performance and feedback from analysts and fraud experts. This feedback loop helps identify potential misclassifications and enables timely updates to the model to account for evolving fraud patterns.
- Adversarial attacks pose a significant challenge to fraud detection systems, as they can lead to severe financial losses and reputational damage. Malicious actors exploit vulnerabilities in the model's decision boundaries, making subtle changes to input data that are imperceptible to humans but can mislead the model into producing incorrect outputs. For instance, attackers may modify features in a transaction or manipulate user behavior to hide fraudulent activities. To ensure the reliability and effectiveness of the proposed deep reinforcement learning model in real-world scenarios, it is essential to assess its robustness against various types of adversarial attacks. There are several approaches to evaluate the model's susceptibility to such attacks:
- Adversarial Testing: Conducting rigorous adversarial testing involves generating adversarial samples and evaluating how the model responds to them. Adversarial samples can be crafted using techniques like Fast Gradient Sign Method (FGSM), Projected Gradient Descent (PGD), or Genetic Algorithms. By testing the model's performance on these samples, researchers can identify vulnerabilities and areas of improvement.
- Robustness Metrics: Various robustness metrics have been proposed to quantify a model's resilience against adversarial attacks. Examples include robust accuracy, fooling rate, and adversarial training loss. These metrics help in comparing the model's performance under normal and adversarial conditions, providing insights into its vulnerability.
- Adversarial Training: Adversarial training is a popular technique to enhance a model's robustness. It involves augmenting the training dataset with adversarial samples, forcing the model to learn from both clean and adversarial data. This process can improve the model's

ability to detect fraudulent activities in the presence of adversarial examples.

- Transferability Analysis: It is crucial to examine whether adversarial attacks generated for one model can also fool other models. Transferability analysis helps in understanding the generalizability of adversarial attacks across different fraud detection models. If attacks transfer across models, it indicates a common vulnerability that needs to be addressed.
- Use of Certified Defenses: Certified defenses, such as certified robustness and provable security techniques, provide mathematical guarantees against adversarial attacks. By incorporating such defenses into the model, the system can offer formal guarantees of security and robustness.
- Real-world Adversarial Testing: It is essential to conduct adversarial testing using real-world data and attack scenarios. This involves simulating how actual attackers might attempt to evade the model's detection mechanisms. This testing should include both known and novel adversarial strategies to ensure comprehensive evaluation.

VI. Conclusion

The suggested model offers a hopeful solution to the issue of identifying fraud in e-commerce transactions, a significant apprehension for companies and customers. The use of multilayer perceptron, RL, and ABC allows for a more robust and accurate fraud detection system. Pre-training the network weights with the evolutionary ABC algorithm is a crucial measure to prevent being trapped in local optima. By using this initialization method, the model is better able to converge to a global optimum, leading to improved accuracy and reliability. The utilization of RL to address dataset imbalance is another notable aspect of the proposed model. Imbalanced datasets are a common challenge in machine learning, and traditional approaches often fail to provide satisfactory results. RL offers a new way to address this issue, allowing the model to learn how to handle imbalanced data on its own, leading to improved performance. Based on the results obtained from the experiments conducted on the utilized dataset, it can be inferred that the proposed model outperforms other existing machine learning models in terms of common metrics. The superior performance of the ABC algorithm and RL over other metaheuristic initialization algorithms and loss functions demonstrates the effectiveness of the suggested approach.

Potential future work includes testing the proposed model on a broader and more varied dataset of e-commerce transactions to assess its ability to generalize. This would allow assessing whether the model is robust enough to identify fraud patterns in different scenarios and datasets, which is crucial for its practical applicability. Additionally, it would be interesting to explore the interpretability of the model, as understanding how it identifies fraudulent transactions can provide valuable insights for fraud detection in e-commerce. Finally, further research could investigate the scalability of the proposed approach, as it may become computationally expensive when dealing with large datasets.

Acknowledgment

This work was supported by Tianjin 2018 Philosophy and Social Sciences Planning Project "Research on the impact of the matching of personal, organizational values of the new generation of college teachers on constructive deviant behavior."

References

[1] U. Porwal and S. Mukund, "Credit card fraud detection in e-commerce: An outlier detection approach," arXiv preprint arXiv:1811.02196, 2018.

[2] R. Jhangiani, D. Bein, and A. Verma, "Machine learning pipeline for fraud detection and prevention in e-commerce transactions," in 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), 2019, pp. 0135-0140: IEEE.

[3] A. Saputra, "Fraud detection using machine learning in e-commerce," International Journal of Advanced Computer Science and Applications, vol. 10, no. 9, 2019.

[4] S. V. Moravvej, S. J. Mousavirad, D. Oliva, and F. Mohammadi, "A Novel Plagiarism Detection Approach Combining BERT-based Word Embedding, Attention-based LSTMs and an Improved Differential Evolution Algorithm," arXiv preprint arXiv:2305.02374, 2023.

[5] S. Danaei et al., "Myocarditis Diagnosis: A Method using Mutual Learning-Based ABC and Reinforcement Learning," in 2022 IEEE 22nd International Symposium on Computational Intelligence and Informatics and 8th IEEE International Conference on Recent Achievements in Mechatronics, Automation, Computer Science and Robotics (CINTI-MACRo), 2022, pp. 000265-000270: IEEE.

[6] S. Moravvej, M. Maleki Kahaki, M. Salimi Sartakhti, and M. Joodaki, "Efficient GAN-based method for extractive summarization," Journal of Electrical and Computer Engineering Innovations (JECEI), vol. 10, no. 2, pp. 287-298, 2022.

[7] I. Mani and I. Zhang, "kNN approach to unbalanced data distributions: a case study involving information extraction," in Proceedings of workshop on learning from imbalanced datasets, 2003, vol. 126, pp. 1-7: ICML.

[8] S. V. Moravvej et al., "RLMD-PA: A reinforcement learning-based myocarditis diagnosis combined with a population-based algorithm for pretraining weights," Contrast Media & Molecular Imaging, vol. 2022, 2022.

[9] M. S. Sartakhti, M. J. M. Kahaki, S. V. Moravvej, M. javadi Joortani, and A. Bagheri, "Persian language model based on BiLSTM model on COVID-19 corpus," in 2021 5th International Conference on Pattern Recognition and Image Analysis (IPRIA), 2021, pp. 1-5: IEEE.

[10] S. V. Moravvej, A. Mirzaei, and M. Safayani, "Biomedical text summarization using conditional generative adversarial network (CGAN)," arXiv preprint arXiv:2110.11870, 2021.

[11] S. V. Moravvej, M. Joodaki, M. J. M. Kahaki, and M. S. Sartakhti, "A method based on an attention mechanism to measure the similarity of two sentences," in 2021 7th International Conference on Web Research (ICWR), 2021, pp. 238-242: IEEE.

[12] !!! INVALID CITATION !!! {}.

[13] L. Hong et al., "GAN - LSTM - 3D: An efficient method for lung tumour 3D reconstruction enhanced by attention - based LSTM," CAAI Transactions on Intelligence Technology, 2023.

[14] X. Hu, Q. Kuang, Q. Cai, Y. Xue, W. Zhou, and Y. Li, "A Coherent Pattern Mining Algorithm Based on All Contiguous Column Bicluster," Journal of Artificial Intelligence and Technology, vol. 2, no. 3, pp. 80-92, 2022.

[15] S. Zhang, C. Tjortjis, X. Zeng, H. Qiao, I. Buchan, and J. Keane, "Comparing Data Mining Methods with Logistic Regression."

[16] S. Vakilian, S. V. Moravvej, and A. Fanian, "Using the artificial bee colony (ABC) algorithm in collaboration with the fog nodes in the Internet of Things three-layer architecture," in 2021 29th Iranian Conference on Electrical Engineering (ICEE), 2021, pp. 509-513: IEEE.

[17] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: a realistic modeling and a novel learning

strategy," IEEE transactions on neural networks and learning systems, vol. 29, no. 8, pp. 3784-3797, 2017.

[18] Q. Yang, X. Hu, Z. Cheng, K. Miao, and X. Zheng, "Based big data analysis of fraud detection for online transaction orders," in Cloud Computing: 5th International Conference, CloudComp 2014, Guilin, China, October 19-21, 2014, Revised Selected Papers 5, 2015, pp. 98-106: Springer.

[19] X. Wang, S. Wang, P.-Y. Chen, X. Lin, and P. Chin, "Block switching: a stochastic approach for deep learning security," arXiv preprint arXiv:2002.07920, 2020.

[20] N. Shakeel and S. Shakeel, "Context-Free Word Importance Scores for Attacking Neural Networks," Journal of Computational and Cognitive Engineering, vol. 1, no. 4, pp. 187-192, 2022.

[21] J. Shaji and D. Panchal, "Improved fraud detection in e-commerce transactions," in 2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA), 2017, pp. 121-126: IEEE.

[22] S. V. Moravvej, S. J. Mousavirad, D. Oliva, G. Schaefer, and Z. Sobhaninia, "An improved de algorithm to optimise the learning process of a bert-based plagiarism detection model," in 2022 IEEE Congress on Evolutionary Computation (CEC), 2022, pp. 1-7: IEEE.

[23] V. Dheepa and R. Dhanapal, "Behavior based credit card fraud detection using support vector machines," ICTACT Journal on Soft computing, vol. 2, no. 4, pp. 391-397, 2012.

[24] S. Wang, C. Liu, X. Gao, H. Qu, and W. Xu, "Session-based fraud detection in online e-commerce transactions using recurrent neural networks," in Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part III 10, 2017, pp. 241-252: Springer.

[25] A. Gasparin, S. Lukovic, and C. Alippi, "Deep learning for time series forecasting: The electric load case," CAAI Transactions on Intelligence Technology, vol. 7, no. 1, pp. 1-25, 2022.

[26] C. Whitrow, D. J. Hand, P. Juszczak, D. Weston, and N. M. Adams, "Transaction aggregation as a strategy for credit card fraud detection," Data mining and knowledge discovery, vol. 18, pp. 30-55, 2009.

[27] A. C. Bahnsen, D. Aouada, and B. Ottersten, "Example-dependent cost-sensitive decision trees," Expert Systems with Applications, vol. 42, no. 19, pp. 6609-6619, 2015.

[28] P. Save, P. Tiwarekar, K. N. Jain, and N. Mahyavanshi, "A novel idea for credit card fraud detection using decision tree," International Journal of Computer Applications, vol. 161, no. 13, 2017.

[29] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, and C. Jiang, "Random forest for credit card fraud detection," in 2018 IEEE 15th international conference on networking, sensing and control (ICNSC), 2018, pp. 1-6: IEEE.

[30] T. Jemima Jebaseeli, R. Venkatesan, and K. Ramalakshmi, "Fraud detection for credit card transactions using random forest algorithm," in Intelligence in Big Data Technologies—Beyond the Hype: Proceedings of ICBDCC 2019, 2021, pp. 189-197: Springer.

[31] R. R. Popat and J. Chaudhary, "A survey on credit card fraud detection using machine learning," in 2018 2nd international conference on trends in electronics and informatics (ICOEI), 2018, pp. 1120-1125: IEEE.

[32] A. S. Hussein, R. S. Khairy, S. M. M. Najeeb, and H. T. ALRikabi, "Credit Card Fraud Detection Using Fuzzy Rough Nearest Neighbor and Sequential Minimal Optimization with Logistic Regression," International Journal of Interactive Mobile Technologies, vol. 15, no. 5, 2021.

[33] A. Kannagi, J. G. Mohammed, S. S. G. Murugan, and M. Varsha, "Intelligent mechanical systems and its applications on online fraud detection analysis using pattern recognition K-nearest neighbor algorithm for cloud security applications," Materials Today: Proceedings, 2021.

[34] T. K. Behera and S. Panigrahi, "Credit card fraud detection: a hybrid approach using fuzzy clustering & neural network," in 2015 second international conference on advances in computing and communication engineering, 2015, pp. 494-499: IEEE.

[35] M. Zamini and G. Montazer, "Credit card fraud detection using autoencoder based clustering," in 2018 9th International Symposium on Telecommunications (IST), 2018, pp. 486-491: IEEE.

[36] V. R. Ganji and S. N. P. Mannem, "Credit card fraud detection using anti-k nearest neighbor algorithm," International Journal on Computer Science and Engineering, vol. 4, no. 6, pp. 1035-1039, 2012.

[37] S. Nami and M. Shajari, "Cost-sensitive payment card fraud detection based on dynamic random forest and k-nearest neighbors," Expert Systems with Applications, vol. 110, pp. 381-392, 2018.

[38] P. Srikanth, "An efficient approach for clustering and classification for fraud detection using bankruptcy data in IoT environment," International Journal of Information Technology, vol. 13, no. 6, pp. 2497-2503, 2021.

[39] W. Min, W. Liang, H. Yin, Z. Wang, M. Li, and A. Lal, "Explainable deep behavioral sequence clustering for transaction fraud detection," arXiv preprint arXiv:2101.04285, 2021.

[40] N. Malini and M. Pushpa, "Analysis on credit card fraud identification techniques based on KNN and outlier detection," in 2017 third international conference on advances in electrical, electronics, information, communication and bio-informatics (AEEICB), 2017, pp. 255-258: IEEE.

[41] B. Lebichot, F. Braun, O. Caelen, and M. Saerens, "A graph-based, semi-supervised, credit card fraud detection system," in Complex Networks & Their Applications V: Proceedings of the 5th International Workshop on Complex Networks and their Applications (COMPLEX NETWORKS 2016), 2017, pp. 721-733: Springer.

[42] S. Elshaar and S. Sadaoui, "Semi-supervised classification of fraud data in commercial auctions," Applied Artificial Intelligence, vol. 34, no. 1, pp. 47-63, 2020.

[43] L. Cui et al., "ALLIE: Active Learning on Large-scale Imbalanced Graphs," in Proceedings of the ACM Web Conference 2022, 2022, pp. 690-698.

[44] N. Tax et al., "Machine learning for fraud detection in e-Commerce: A research agenda," in Deployable Machine Learning for Security Defense: Second International Workshop, MLHat 2021, Virtual Event, August 15, 2021, Proceedings 2, 2021, pp. 30-54: Springer.

[45] D. Malekian and M. R. Hashemi, "An adaptive profile based fraud detection framework for handling concept drift," in 2013 10th International ISC Conference on Information Security and Cryptology (ISCISC), 2013, pp. 1-6: IEEE.

[46] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," ACM computing surveys (CSUR), vol. 46, no. 4, pp. 1-37, 2014.

[47] C. Winship and R. D. Mare, "Models for sample selection bias," Annual review of sociology, vol. 18, no. 1, pp. 327-350, 1992.

[48] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," Journal of global optimization, vol. 39, pp. 459-471, 2007.

[49] S. Vakilian, S. V. Moravvej, and A. Fanian, "Using the cuckoo algorithm to optimizing the response time and energy consumption cost of fog nodes by considering collaboration in the fog layer," in 2021 5th International Conference on Internet of Things and Applications (IoT), 2021, pp. 1-5: IEEE.

[50] M. Hallin, D. Paindaveine, and M. Šiman, "Université Libre de Bruxelles," The Annals of Statistics, vol. 38, no. 2, pp. 694-703, 2010.

[51] M. A. de Almeida, "DATA MINING: DETERMINAC AO DE AGRUPAMENTOS EM GRANDES BASES DE DADOS," Universidade Federal do Rio de Janeiro, 2013.

[52] G. I. Webb, E. Keogh, and R. Miikkulainen, "Naïve Bayes," Encyclopedia of machine learning, vol. 15, pp. 713-714, 2010.

[53] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "KNN model-based approach in classification," in On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings, 2003, pp. 986-996: Springer.

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

[55] M. P. LaValley, "Logistic regression," Circulation, vol. 117, no. 18, pp. 2395-2399, 2008.

[56] A. J. Myles, R. N. Feudale, Y. Liu, N. A. Woody, and S. D. Brown, "An introduction to decision tree modeling," Journal of Chemometrics: A Journal of the Chemometrics Society, vol. 18, no. 6, pp. 275-285, 2004.

[57] S. J. Mousavirad and H. Ebrahimpour-Komleh, "Human mental search: a new population-based metaheuristic optimization algorithm," Applied Intelligence, vol. 47, pp. 850-887, 2017.

[58] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," International journal of bio-inspired computation, vol. 2, no. 2, pp. 78-84, 2010.

[59] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," Nature inspired cooperative strategies for optimization (NICSO 2010), pp. 65-74, 2010.

[60] S. J. Mousavirad, D. Oliva, S. Hinojosa, and G. Schaefer, "Differential evolution-based neural network training incorporating a centroid-based strategy and dynamic opposition-based learning," in 2021 IEEE Congress on Evolutionary Computation (CEC), 2021, pp. 1233-1240: IEEE.

[61] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," Advances in engineering software, vol. 69, pp. 46-61, 2014.

[62] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in 2009 World congress on nature & biologically inspired computing (NaBIC), 2009, pp. 210-214: Ieee.

[63] Ö. Özdemir and E. B. Sönmez, "Weighted cross-entropy for unbalanced data with application on covid x-ray images," in 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), 2020, pp. 1-6: IEEE.

[64] F. Huang, J. Li, and X. Zhu, "Balanced Symmetric Cross Entropy for Large Scale Imbalanced and Noisy Data," arXiv preprint arXiv:2007.01618, 2020.

[65] X. Li, X. Sun, Y. Meng, J. Liang, F. Wu, and J. Li, "Dice loss for data-imbalanced NLP tasks," arXiv preprint arXiv:1911.02855, 2019.

[66] S. S. M. Salehi, D. Erdogmus, and A. Gholipour, "Tversky loss function for image segmentation using 3D fully convolutional deep networks," in Machine Learning in Medical Imaging: 8th International Workshop, MLMI 2017, Held in Conjunction with MICCAI 2017, Quebec City, QC, Canada, September 10, 2017, Proceedings 8, 2017, pp. 379-387: Springer.

[67] S. A. Taghanaki et al., "Combo loss: Handling input and output imbalance in multi-organ segmentation," Computerized Medical Imaging and Graphics, vol. 75, pp. 24-33, 2019.