Bee-inspired knowledge transfer: synthesizing data for enhanced deep learning explainability

Kritanat Chungnoy[1], Tanatorn Tanantong[1,2], Pokpong Songmuang[1,2]
[1]Department of Computer Science, Faculty of Science and Technology, Thammasat University, Pathum Thani, Thailand
[2]Thammasat University Research Unit in Data Innovation and Artificial Intelligence, Thammasat University, Pathum Thani, Thailand

Article Info

Article history:

Received Mar 13, 2024
Revised Jul 3, 2024
Accepted Jul 14, 2024

Keywords:

Bee algorithm
Deep learning
Explainability
Explainable AI
Synthesize data

ABSTRACT

This paper presents the generation method for an explainable model based on the given information of a black box model using a concept of knowledge transfer to synthesize a dataset. The proposed method applies with GAN and Bee algorithm (BA) for data synthesis technique to synthesize a dataset by considering loss value in a knowledge transferring process to inherit the significance of features. The synthesized dataset is used to train for a proxy model as an explainable model. The result of the experiment indicates that knowledge transfer from Bee algo better than generative adversarial network (GAN) in terms of the coefficient of determination R2. In addition, explainable models from the synthesized data of the Bee-based method obtains F1 score superior to those from the GAN-based method in all datasets and settings. The dataset synthesized from the Bee-based method produces the explainable prediction model that has similar top-10 features according to similarity score of 0.6718 using shapley additive explanations (SHAP) feature importance which is higher than those from GAN-based method for 0.4218 in average. Additionally, experimental result to evaluate accuracy shows that F1 score from explainable models from the Bee-based method are closed to F1 score from a model generated from the original dataset.

Corresponding Author:

Tanatorn Tanantong
Thammasat University Research Unit in Data Innovation and Artificial Intelligence, Thammasat University
Pathum Thani, Thailand
Email: tanatorn@sci.tu.ac.th

1. INTRODUCTION

Artificial intelligence (AI) has been developed and used in many tasks and domains [1]. Among many approaches, deep learning approach has gained significant popularity and widespread adoption due to scalability, versatility, and state-of-the-art performance [2]–[12]. However, models from deep learning are a black box model with deep, computationally expensive layers and have been recently found vulnerable to spoofing with well-designed input samples in many safety-critical applications [13]. The black box model refers to the lack of transparency and interpretability from containing a large number of complex parameters, making it challenging to understand why the model chooses particular predictions. Unfortunately, transparency and accountability are ones of crucial factors towards ethics guidelines for trustworthy AI [14] by European Commission standard. Addressing the black box nature of deep learning models is an ongoing research area, and one of the solutions is explanatory artificial intelligence (XAI) which is a technique to make these models more interpretable without sharply sacrificing their performance.

XAI refers to techniques and approaches that aim to make AI systems, particularly deep learning models, more transparent and understandable [15]. The goal of XAI is to provide insights into why AI systems make specific decisions or predictions, enhancing their interpretability and enabling users to trust and verify their outputs. XAI thus plays a role in bridging the gap between the technical complexity of AI models and the need for human understanding, accountability, and trust. As the use of deep learning models continues to integrate into critical applications, the demand for XAI techniques grows accordingly and has become one of focused task in AI development [16]. There are several XAI methods such as rule-based explanations, attention mechanism, and feature importance. However, most of the existing method requires an original dataset to identify which input features or variables have the most significant impact on a model's predictions. Techniques such as feature importance scores, permutation importance, and SHAP (shapley additive explanations) values can help to quantify the contribution of each feature to the model's output for explanation. Some existing research focus on explaining individual predictions using local interpretable model-agnostic explanations (LIME) to approximate the model's behavior for a specific input by generating a simpler and interpretable model (such as linear regression and decision tree (DT)) around that input to explain its prediction. Unfortunately, most of the practical AI models may not share their training data due to copy right or trade secret. Thus, it becomes difficult to generate an XAI model without the training dataset.

In this paper, we propose a method to generate XAI model based on the concept of knowledge transfer inspired by knowledge distillation (KD) architecture. The method does not require the training dataset of the black box model, but the complex model itself as a teacher model. The method involves transferring knowledge directly from a teacher model (large machine learning model) to generate a synthesized dataset that ensembles the characteristics of a teacher model for training for a simpler and interpretable prediction model as an explainable model towards the original complex model. The obtained explainable model hence can be used to explain the prediction and trace for bias in decision-making of deep learning-based AI. The core hypothesis of this method is that knowledge within a complex model contains can be transferred to its synthesized data, and the knowledgeable synthesized data can be used to generate a prediction model that resembles to the original complex model in terms of similarity of feature importance and capability to predict similar to the original model. The chosen data synthesis method in this work includes the frequently used generative adversarial network (GAN) and Bee algorithm (BA). The scope of data types in this work limits to tabular and numerical data.

The contributions of this paper includes:

- We propose a novel method to generate an XAI model from a complex black box model bases on KD-based data synthesis call "knowledge-transferred data synthesis for explainable AI" (KT-XAI).
- We demonstrate on how to apply GAN and BA for knowledge-transferred data synthesis, called KT-GAN and KT-Bee, respectively.
- We demonstrate that the proposed KT-Bee contributes to synthesize a dataset that inherits knowledge from a teacher model superior to KT-GAN.
- We demonstrate that synthesized datasets from KT-Bee are systematically similar to the model generated from the original dataset by sharing top-k important features and can be used as an explainable model to the black box model.
- We observe the knowledge transfer parameters including data distribution, transfer accuracy and sharing of top-k important features and found that the size of synthesized dataset does not affect the performance of knowledge-transferred data synthesis.

2. BACKGROUND

2.1. Explainable artificial intelligent

XAI is an approach to AI that aims to make machine learning models and their decisions more understandable and interpretable by humans since many AI models, particularly deep learning models, are difficult to understand how they arrive at their conclusions. XAI thus helps to identify and mitigate bias in AI models by revealing which features or data points are influencing decisions to improve transparency and allows for the identification and rectification of unfair or discriminatory outcomes [17]. DTs is another approach often used as a proxy model. There are several works mentioned on decomposing neural network models into DT models. Deep rule extraction from DTs (DeepRED) [18] demonstrates the extending of the comprehensible rule extraction from DTs (CRED) [19] algorithm which is designed for shallow networks to arbitrarily many hidden layers. DeepRED aims to simplify using RxREN [20] to reduce unnecessary input and applies algorithm C4.5 [21] to create a DT. DeepRED is able to construct complete trees closely faithful to the original neural network model, but the generated trees are large and requires time and memory; thus, it is difficult in terms of scalability.

Artificial neural network-decision tree (ANN-DT) [22] is another DT-based proxy model. The ANN-DT extracts binary DT from a trained neural network and generate outputs for samples interpolated from the training dataset. The criterion of an attribute selection is based on a significance analysis of the variables on the neural-network output. The ANN-DT is able to extract rules from feedforward neural networks with continuous outputs. These extracted rules are from the neural network without making assumptions about the internal structure of the neural network or the features of the data.

2.2. Knowledge distillation

KD [23] is a technique in machine learning to compress a large machine learning model as teacher model to a smaller and more understandable model as student model but able to maintain the behavior of the original model. KD is specifically used in scenarios where processing power, memory, and storage are limited. The goal of KD is to transfer the knowledge and insights learned by the teacher model to the student model, resulting in a more compact and efficient model. The teacher model is typically a large complex model that has been trained on a large dataset using a deep learning and achieves acceptable accuracy but is computationally expensive and memory-intensive. The student model is a smaller and simpler model that is designed to mimic the behavior of the teacher model with fewer parameters to be more lightweight and suitable for deployment on resource-restricted devices. During the training, the loss function is used for training the student model aiming to compare the soft targets produced by the teacher model with the student model's predictions. The distillation loss encourages the student model to align its predictions with those of the teacher model. Through the training process, the student model learns from the teacher model's decision boundaries, pattern recognition, and generalization capabilities. This allows the student model to capture the underlying knowledge of the teacher model. With the ability to transfer knowledge from a complex model to a more compact model of KD, we are interested to apply the method to synthesize a dataset from a complex model as a teacher model. The synthesized data with knowledge from the teacher model thus can be a representative as an explainable model of the black box model.

2.3. Generative adversarial network

A GAN is a machine learning technique well-known for the generation of new data instances that resemble a given dataset as synthesized data [24]. GAN consists of two neural networks as generator and discriminator. The former is responsible for generating new data instances that mimic the data by considering random noise and include it into data samples. The latter then observes both original data samples and generated data samples and attempts to distinguish between them whether which samples are real, and which are generated. Upon checking the discriminating results, loss value is calculated to improve generator criteria until the generator becomes proficient at creating data that is difficult for the discriminator to distinguish. In practical, Addepalli *et al.* [25] applied GAN to generate data enriching dataset and solve imbalanced data issues for training data for deep learning. For its ability to reliably generate resembled data, conditional tabular-GAN (CT-GAN) algorithm [26] is thus selected to apply for knowledge transferred data synthesis for this work.

2.4. Bee algorithm

BA is a population-based optimization technique inspired by the foraging behavior of bees to find optimal solutions for optimization problems [27], [28]. It is famous for an ability to explore the search space efficiently and handle complex problems successfully. As the concept of food foraging behavior, the parameters in the algorithm include the number of scout bees (n), elite bees (e), number of food areas for foraging (m), number of target food areas, and number of bees for foraging [27]. The steps of BA are as follows.

The algorithm starts by generating random bee population and environment based on setting parameters. Then, scout bees are sent out to scout for solutions, and the solutions are evaluated using fitness function. Elite bees select the solution based on fitness scores where the higher the score, the more probability the solution is selected. The neighborhood of the selected solution is searched for expanding the possible solutions for scout bees to forage, and they are also evaluated using fitness function. Then the solution among all solutions is selected as representative solution.

In 2019, Chungnoy *et al.* [29] adapted BA to solve missing data by imputing the data systematically. The BA-based data imputation showed the ability to generate missing data effectively. The imputed dataset was evaluated to yield the highest accuracy and superior to other techniques for 23% in average. It signifies that BA has capability to generate a data value with high accuracy and impact to a machine learning model. Hence, we select BA to be used as one of the techniques for knowledge transferred data synthesis for this work.

3. MATERIALS AND METHODS

This paper presents a method for developing XAI for a black box model based on KD concepts and data synthesis. The synthesized data that are transferred from a black box model are used to train for AI can become a model that can explain the reason why the AI chooses the output. Thus, the method consists of two main parts as knowledge transfer and explainable model generation as shown in Figure 1.

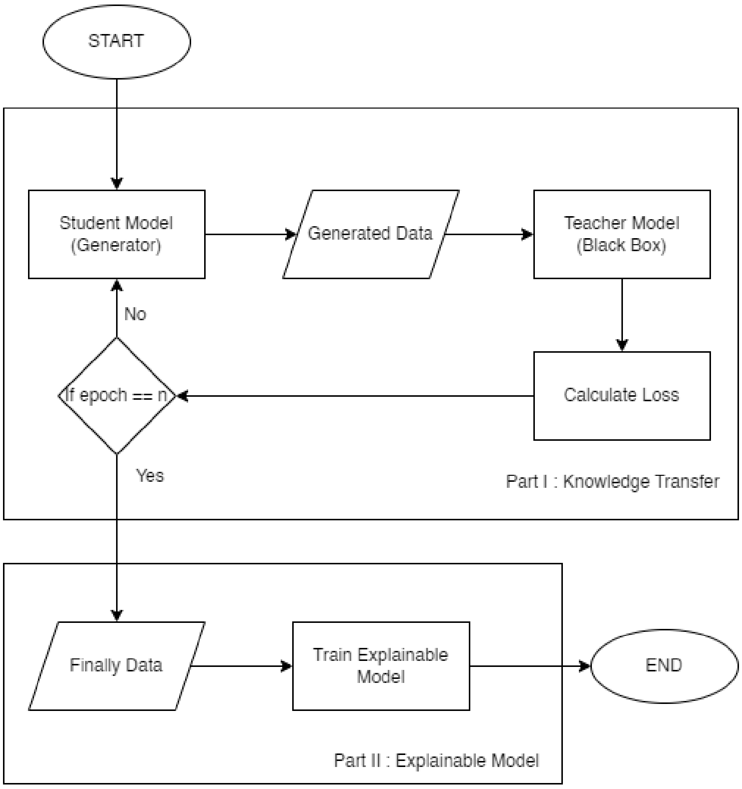

START
Student Model (Generator)
Generated Data
Teacher Model (Black Box)
Calculate Loss
If epoch == n
No
Yes
Part I : Knowledge Transfer
Finally Data
Train Explainable Model
END
Part II : Explainable Model

Figure 1. An overview of the process

3.1. Knowledge transfer

For the knowledge transfer part, we generate synthesis data from the teacher model which is a black box model developed from a deep learning classification. The synthesized data as a student model are used for prediction along with a prediction from the teacher model. Loss function calculates for loss value and uses it to adjust the student model for better prediction performance. The method is repeated until the assigned epoch iteration is met. The result of this part is the final synthesized dataset to be used for developing an XAI model.

In this paper, two techniques of data synthesis with knowledge transfer from a teach model are presented. First, we present knowledge transfer based on GAN (KT-GAN), which applies model generator from CT-GAN [26]. This technique synthesizes tabular data. Second, we propose a knowledge transfer based on Bee algorithm (KT-Bee) in data synthesis.

3.1.1. Knowledge transfer based on generative adversarial network

KT-GAN applies the method of a model generator from CT-GAN [26]. With the method, the tabular data are synthesized as a student model. Instances in synthesized dataset are used in prediction and they are compared to a prediction from the teacher model to find difference. Loss then is calculated using binary cross entropy loss (1).

$$Loss = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i)) \tag{1}$$

Where y is the label (whereas 1 refers to matching label between predictions of a teacher model and a student model and 0 for not-matching label) and p(y) is a probability of label being 1. In step-wise, KT-GAN processes are given in Figure 2. The pseudo code of all steps is as follows.

```
i)    Initial step
        – Assign classes
        – Assign $i$ number of instances and $f$ number of features for synthesis for all assigned classes
        – Create class label for all instances (in this work, we create class labels as balance class type)
        – Generate data for each feature at random within known boundary (min-max) for all $i$ instances and $f$ features
ii)   Use the generated dataset to test for prediction result using the teacher model and collect class labels for all instances
iii)  Compare the result of predictions from teacher model and class label from the generated data whether they are matched or not
iv)   Assign matching label if they are matched as 1 and unmatched as 0
v)    Apply matching label to calculation with loss function in (1) and use loss result to calculate for adjust weight for generator model.
vi)   Use generator model to synthesize generated dataset
vii)  Repeat 2 - 6 for assigned epoch
viii) END
```

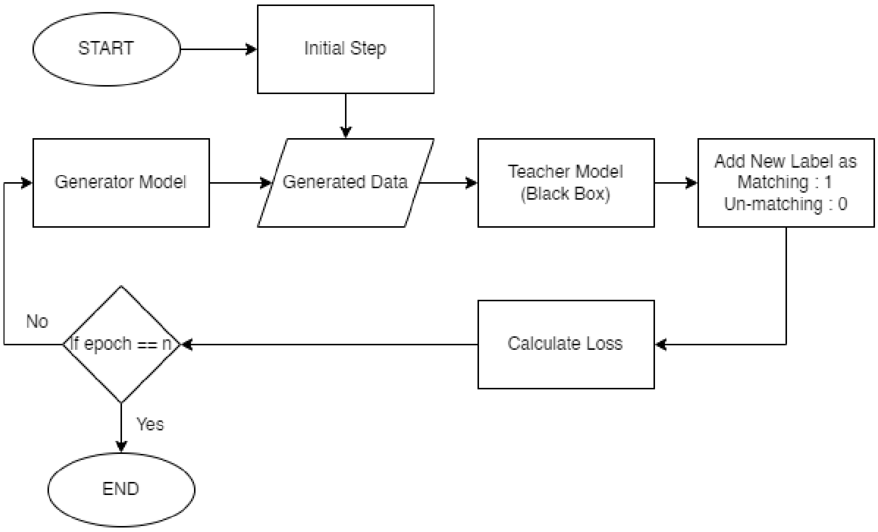


Figure 2. Processes of KT-GAN

3.1.2. Knowledge transfer based on Bee algorithm

KT-Bee exploits BE to synthesize data according to given information in the teacher model. The core process is to let bees to find the best solution of having the lowest loss value. There are two types of bees including scout bee and follower bee. The scout bees are tasked to randomly generate a data value based on a specified data type as a solution. The follower bees are to choose a solution by considering fitness function of a scout bee based on probability. The higher score from fitness function represents the higher quality of a solution and has more probability to be chosen, and vice versa. Once a follower bee chooses a solution, the data value as the chosen solution is adjusted. After all solutions are adjusted, fitness function will be updated to reflect the change. A new set of scout bees then begin another round of foraging to generate a solution from the updated dataset until the assigned iteration is met. The objective function of this process is to minimize the loss value from binary cross entropy loss using (2).

$$Loss_{bee(j)} = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(m_i) + (1 - y_i) \cdot log(1 - m_i) \tag{2}$$

Where y is the label (the y value is always 1, where 1 means matching), and m is a label is assigned to if predictions of a teacher model and a student model are matched (1) and for not-matching (0) label.

In step-wise (Figure 3), KT-Bee processes are given in Figure 3(a). It can be split into 2 phases as process of scout bees in Figure 3(b) and process of follower bees in Figure 3(c). The pseudo code of all steps is as follows.

1) Initial step
   (a) Assign classes
   (b) Assign i number of Instances and f number of features for synthesis for all assigned classes
   (c) Create class label for all instances (in this work, we create class labels as balance class type)
2) Scout bee step (Figure 3(b))
   (a) Generate data for each feature at random within known boundary (Min-Max) for all i instances and f features of b number of scout bees
   (b) Use the generated dataset to test for prediction result using the teacher model and collect class labels for all instances
   (c) Compare the result of predictions from teacher model and class label from the generated data whether they are matched or not
   (d) Assign matching label if they are matched as 1 and unmatched as 0
   (e) Apply matching label to calculation with loss function in (2) and use Loss result to calculate for fitness function of each scout bee using (3). The higher fitness value of a solution from a scout bee, the more probability it is chosen by follower bees.
3) Following bee step (Figure 3(c))
   (a) m number of follower bee chooses a solution regarding fitness value based on probability calculated from (4) where $p(bee_j)$ represents probability of choosing by each follower bee.
   (b) Each follower bee randomly selects a feature to change based on probability calculated given in (7) where $p(feature_k)$ is a probability of feature k, and k refers to k^{th} feature while m represents a number of all features. In the first iteration, weight of all features is set as 1, and the chosen solution for data change is limited to the instances that are under the matching label of 0 (not matching).
   (c) Find top k similarity instance between unmatching group and matching group with results from (8) where p is a considered vector, and q is a target vector to find similarity. r represents r^{th} feature, and s is a number of all features.
   (d) Use the top K instances (from 3(c)) to train for regression model to find prediction values to change data for the selected feature (from 3(b))
   (e) Repeat step 3(c) and 3(d) for all instances with matching label of 0
   (f) Use the changed solution to predict class label using the teacher model and assign the new matching label. Then re-calculate for loss and fitness function and update the weight of each feature using (6). If the new weight is less than 0, the weight is set as 0.
4) Repeat 2 and 3 according to the assigned iteration and if iteration is greater than n, the following bee has a probability to randomly copy solution from the previous iteration.
5) Repeat 2, 3, and 4 until all designated classes are processed
6) END

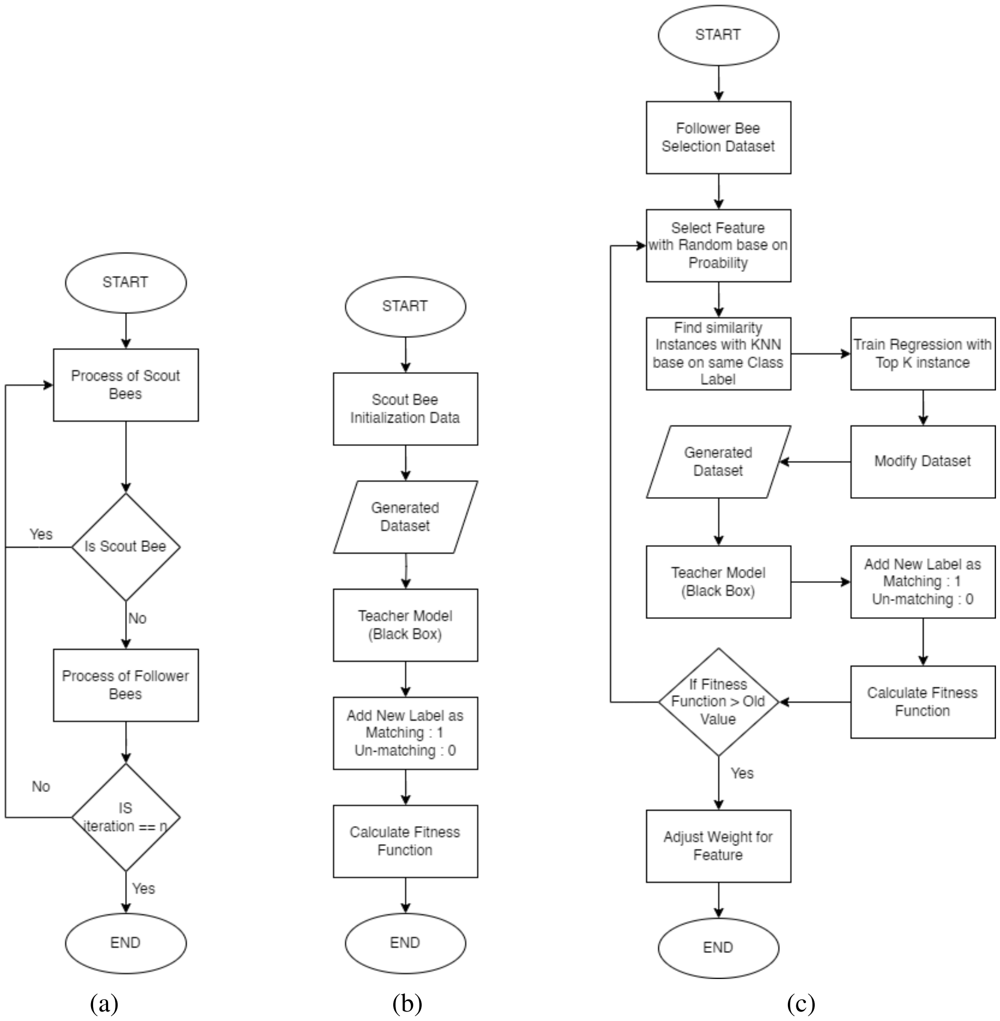

START
Process of Scout Bees
Is Scout Bee
Yes
No
Process of Follower Bees
IS iteration == n
No
Yes
END
(a)
START
Scout Bee Initialization Data
Generated Dataset
Teacher Model (Black Box)
Add New Label as Matching : 1 Un-matching : 0
Calculate Fitness Function
END
(b)
START
Follower Bee Selection Dataset
Select Feature with Random base on Proability
Find similarity Instances with KNN base on same Class Label
Train Regression with Top K instance
Modify Dataset
Generated Dataset
Teacher Model (Black Box)
Add New Label as Matching : 1 Un-matching : 0
Calculate Fitness Function
If Fitness Function > Old Value
Yes
Adjust Weight for Feature
END
(c)

Figure 3. Processes of KT-Bee: (a) overview processes of KT-Bee, (b) processes of scout bees, and (c) processes of following bees

$$fitness(bee_j) = \frac{1}{Loss(bee_j)} \tag{3}$$

$$p(bee_j) = \frac{fitness(bee_j)}{\sum_{j=1}^{n} fitness(bee_j)} \tag{4}$$

$$fitness_{diff} = fitness(bee_j)_{new} - fitness(bee_j)_{old} \tag{5}$$

$$weight_{new} = weight_{current} + fitness_{diff} \tag{6}$$

$$p(feature_k) = \frac{weight(feature_k)}{\sum_{l=1}^{m} weight(feature_l)} \tag{7}$$

$$d_{pq} = dist(x_p, x_q) = \sqrt{\sum_{r=1}^{s} (x_{pr} - x_{qr})^2} \tag{8}$$

3.2. Training for explainable model

To obtain an explainable model that represents the complex AI model, the synthesized dataset is trained for classification to resemble the original model. To keep an explainable model simple and inherently interpretable, DT technique is chosen as the model generated by DT representing decisions and decision-making processes in a human-readable tree-like structure, with each node representing a decision based on a feature, and each branch representing the outcome of that decision. Furthermore, the transparency of DT model makes it easy to follow the logic behind a model's predictions. Thus, it will help user to visually trace the path from the root node to a leaf node to understand how the model arrived at a particular decision. However, we are aware that DT might be too simplistic to capture intricate patterns in the data and may not offer the same predictive accuracy as more complex models like deep neural networks.

4. EXPERIMENT SETTING

In this work, an experiment is separated into 5 parts as shown in Figure 4 including: i) data preparation, ii) training teacher model, ii) knowledge transfer to synthesis dataset using KT-GAN or KT-Bee, iv) develop an explainable model from synthesized data, v) evaluation of the explainable model, and vi) evaluation comparison.

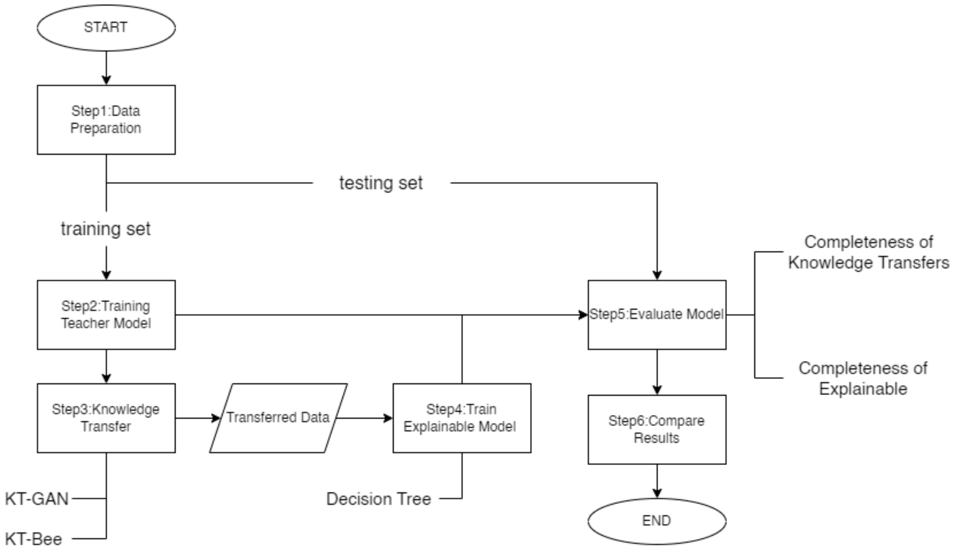


Figure 4. Experiment overview

4.1. Data preparation

In this experiment, we selected 6 datasets including Higgs dataset (HI) [20], Jannis dataset (JA) [21], HLS4ML LHC Jet dataset (HLS) [30], [31], MagicTelescope (MT) [32], MiniBooNE (MB) [33] and numerai (NU) [34]. The data were prepared by discarding a class with less than 15% of the total data. We also removed instances with a missing value(s) and processed the dataset to become a balance dataset as a training set for a teacher model. Statistics of prepared datasets are shown in Table 1. For the testing set, 20% of datasets are separated for model evaluation.

Table 1. Dataset details

Dataset	Number instances	Number Features	Number class
Higgs dataset	92,446	28	2
Jannis dataset	44,202	54	3
HLS4ML LHC Jet dataset	805,330	16	5
MiniBooNE	72,998	50	2
Numerai	95,324	21	2
MagicTelescope	13,376	10	2

4.2. Training teacher models

To obtain teacher models, we train each dataset with general deep learning for classification. The obtained models are considered as a teacher model that represents a high quality classification model in terms of accuracy in a form of complex and large black box model. Parameter setting for training the teacher models is as follows: i) layer: 5 layer (32,16,8,4,2); ii) activation function: rectified linear unit (ReLU); iii) weight optimization: Adam; iv) learning rate: constant; and v) epoch: 10,000 epoch. The teacher models were evaluated for performance using 10 fold-cross-validation and yielded the accuracy result as shown in Table 2.

Table 2. Accuracy performance of teacher models trained with 10-fold cross-validation

Dataset	Higgs	Jannis	HLS4ML LHC Jet	MiniBooNE	Numerai	MagicTelescope
General deep	0.7054	0.6231	0.6748	0.8695	0.5090	0.7397

4.3. Knowledge transfer

To obtain the synthesized data with knowledge-transferred from the teacher model, we applied either KT-GAN or KT-Bee separately to compare their performances. The parameter setting for data synthesis and knowledge transferring is as follows.

i) KT-GAN
   - Using “ctgan” library in python version 0.7.3 [26] as data generator with all default parameter settings and epoch is set to 100

ii) KT-Bee
   - Number of scout bee: 20
   - Number of follower bee: 100
   - Top k similarity: 10
   - Iteration: 100

The knowledge-transferred synthesized data are generated into 3 sizes regarding number of instances as i) same instance number to the original dataset (original size); ii) 50% less instances than the original dataset (50% less size); and iii) 50% more instances than the original dataset (50% more size). The sizes are to study whether the size of synthesized data have the effect on classification performance or not.

4.4. Training explainable model

The synthesized datasets are generated to train for an explainable model to enhance interpretability of prediction decision of a black box model. In this experiment, we chose the DT classification approach based on scikit-learn version 1.3.0 [35] to represent the classification model as the model from DT is easy to interpret and follow. The DT classification has the parameter setting shown in Table 3. We then evaluated performance of the generated explainable models using 10 fold-cross-validation as showed in Table 4.

Table 3. Parameter setting for explainable model

Parameter	Values
Criterion	Gini
Splitter	Best
Maximum depth	None
Minimum samples split	2
Minimum samples leaf	1
Minimum weight fraction leaf	0.0
Maximum features	None
Random state	None
Maximum leaf nodes	None
Minimum impurity decrease	0.0
Class weight	None
Cost-complexity pruning alpha	0.0

Table 4. Performance of explainable models with knowledge-transferred synthesized data

Dataset		KT-Bee	KT-GAN
HI	50% less size	0.8184	1.000
	Original size	0.8268	1.000
	50% more size	0.8364	1.000
JA	50% less size	0.7363	0.9932
	Original size	0.7651	0.9966
	50% more size	0.7733	0.9979
HLS	50% less size	0.8975	0.9997
	Original size	0.8897	0.9999
	50% more size	0.9999	0.9999
MT	50% less size	0.8686	0.9983
	Original size	0.8766	0.9991
	50% more size	0.8728	0.9994
MB	50% less size	0.8466	1.000
	Original size	0.8677	1.000
	50% more size	0.8536	1.000
NU	50% less size	0.7689	1.000
	Original size	0.7742	1.000
	50% more size	0.7652	1.000

4.5. Evaluation and comparison

To evaluate and compare the explainable models from the two presented methods, two aspects are considered including completeness of knowledge transferring and completeness the models as shown in Figure 5. The knowledge transfer method presented in this work draws inspiration from KD, a technique for transferring knowledge from a large deep learning model (teacher model) to a smaller deep learning model (student model). The goal is to enable the student model to achieve performance close to that of the teacher model, allowing it to make predictions similar to the teacher model and thus be suitable for deployment on mobile devices. To evaluate the effectiveness of the proposed knowledge transfer method, we developed performance metrics tailored to assess the successful transfer of knowledge from a black-box model (teacher model) to synthetic dataset believed to represent the dataset used to train the black-box model.

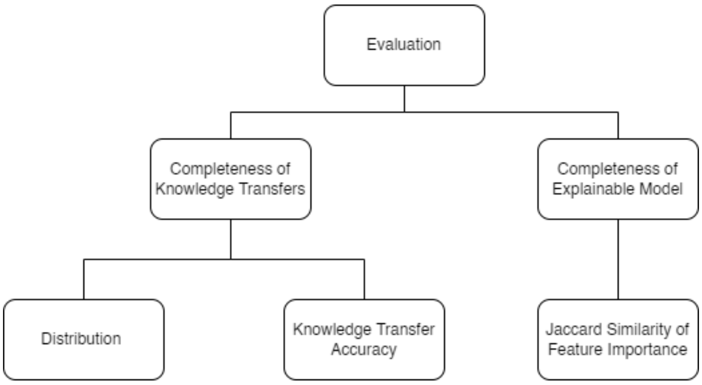

Evaluation
Completeness of Knowledge Transfers
Completeness of Explainable Model
Distribution
Knowledge Transfer Accuracy
Jaccard Similarity of Feature Importance

Figure 5. Evaluation overview

The aspect of completeness of knowledge transferring considers distribution of the data of each feature and performance of the knowledge transferring (KTP). For distribution of data, we assume that the distribution of data of transferred data should be similar or close to the original data, and we evaluated by comparing the distribution of the original dataset and the synthesized dataset. KTP: this metric involves using the dataset obtained through the knowledge transfer process to train a prediction model and then evaluating its performance on testset. The prediction results should ideally match those of the teacher model. A higher KTP value indicates that the synthetic dataset captures the knowledge of the teacher model more accurately. This aligns with the hypothesis that if knowledge transfer is successful, the model trained using synthetic dataset should produce prediction results similar to those of the teacher model. For performance of the knowledge transferring, KTP can be calculated using (9).

$$KTP = \frac{\sum_{i=1}^{n}(\hat{y}_t - \hat{y}_s) \times 100}{N} \tag{9}$$

Where $\hat{y}_t$ refers to number of instances that a prediction of teacher model and student model is matched, while $\hat{y}_s$ is a number of unmatched predictions. N is a number of all instances.

For the completeness of the generated explainable model, we consider the similarity aspect of feature importance between the black box model and its explainable model. The used similarity calculation in this experiment is Jaccard similarity. We consider top-10 ranking of feature importance from both datasets using (10).

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cup B|} \tag{10}$$

Where A is a set of feature importance of the explainable model, and B refers to a set of feature importance of the black box model. Furthermore, we consider SHAP [36], [37] value calculated default settings to determine feature importance and relationship between the explainable model and the black box model. Our hypothesis is that the higher the similarity score, the better completeness of the explainable model.

4.6. Experiment tool

In this experiment, all experiments were run in a single computer in the same environment as follows.

i) Computer spec
   - Processor: Intel(R) Core(TM) i5-3450 CPU @ 3.10GHz 3.10 GHz
   - RAM DDR3 : 32.0 GB
   - Graphic Card: NVIDIA GeForce GTX 1060 6 GB
   - Hard disk: 1 TB
   - System type : Window 10 64-bit operating system

ii) Programing language
   - Knowledge transfer
     - KT-Bee: Python (3.9.1)
     - KT-GAN: Python (3.9.1) with “ctgan” library in python version 0.7.3
   - Explainable model
     - Decision tree: Python (3.9.1) library from scikit-learn version 1.3.0

5. RESULTS

5.1. Completeness of knowledge transferring of synthesized data

In this part, we provide an experiment result of completeness of knowledge transferring between synthesized datasets from KT-GAN and KT-Bee technique. The two aspects in consideration are distribution and accuracy of knowledge transferring.

5.1.1. Distribution comparison results

The measurement used to determine distribution between the synthesized dataset and the original dataset is R2. The distribution result comparison is given in Table 5. Table 5 show the experimental results demonstrate that the KT-Bee method outperforms the KT-GAN method in synthesizing new datasets that closely resemble the original dataset. This aligns with our hypothesis that knowledge transfer from a teacher model to a dataset should result in a synthesized dataset that is highly similar to the original dataset.

Table 5. Result R2 to determine the distribution difference between the synthesized dataset and the original dataset

Datasets	KT-Bee	KT-GAN
HI	**-1.5513**	-2.8540
JA	**-0.9150**	-1.6074
HLS	**-1.0365**	-1.5139
MT	**-0.9810**	-1.2929
MB	**-1.6717**	-1.8484
NU	**-0.8599**	-1.0091

For accuracy of explainable model, we use F1-score of classification result between the synthesized dataset and the original dataset. Comparison is also made using different sizes of synthesized datasets from KT-GAN and KT-Bee. The F1-score results are given in Table 6. The results indicate that synthesized data from KT-Bee method has better performance than KT-GAN in all datasets although the R2 score is in negative. Furthermore, the F1-scores of the explainable models from KT-Bee are closed to the F1-score from a classification model of its corresponding original dataset. The results also signify that size difference do not have a noticeably effect on accuracy performance on a classification task.

Table 6. Comparison results of F1 score from explainable model and classification model from original data

Dataset	Data size	KT-Bee	KT-GAN	Original data
HI	50% less size	**0.6119**	0.5000	0.6257
	Original size	**0.6312**	0.5057	
	50% more size	**0.6184**	0.5000	
JA	50% less size	0.5009	**0.5026**	0.5863
	Original size	0.4646	**0.4667**	
	50% more size	**0.4889**	0.4697	
HLS	50% less size	**0.4110**	0.3884	0.6591
	Original size	**0.3793**	0.3420	
	50% more size	**0.4308**	0.3270	
MT	50% less size	**0.6816**	0.6288	0.7945
	Original size	**0.7155**	0.6160	
	50% more size	**0.7300**	0.6115	
MB	50% less size	**0.7837**	0.6084	0.8801
	Original size	**0.7927**	0.5705	
	50% more size	**0.8163**	0.6686	
NU	50% less size	**0.5114**	0.4965	0.5019
	Original size	**0.5085**	0.5019	
	50% more size	**0.5105**	0.4927	

5.1.2. Accuracy results of knowledge transferring

Accuracy results are an evaluation of how accurate knowledge is transferred from the teacher model to an explainable model using KTP calculation (given in (9)). The KTP results are given in Table 7. Table 7 show the KT-Bee method effectively transfers knowledge from a teacher model to a synthesized dataset more comprehensively compared to the KT-GAN method. This is evident from the consistently higher KTP values achieved by the KT-Bee method across all data sizes for each dataset.

Table 7. Result of KTP score

Datasets	Data size	KT-Bee	KT-GAN
HI	50% less size	**99.9134**	57.8292
	Original size	**99.8972**	58.5747
	50% more size	**99.3156**	61.3638
JA	50% less size	**100.0000**	54.4455
	Original size	**99.9343**	59.4973
	50% more size	**99.8914**	58.1074
HLS	50% less size	**99.7509**	69.7378
	Original size	**98.7093**	69.1699
	50% more size	**98.8140**	69.6694
MT	50% less size	**100.0000**	59.0909
	Original size	**100.0000**	58.9937
	50% more size	**100.0000**	58.1937
MB	50% less size	**100.0000**	79.9561
	Original size	**100.0000**	82.8392
	50% more size	**100.0000**	80.5983
NU	50% less size	**94.0749**	50.6462
	Original size	**93.8661**	50.1573
	50% more size	**95.2456**	51.6854

Moreover, KTP from KT-Bee are 98.85 in average and higher than KT-GAN for 36.05 in average. The one-way analysis of variance (ANOVA) was conducted to determine if the difference in KTP results from each method was significant or not. With Alpha value of 0.05, there was p-value of 5.26×10^{-15} which signified that at least one pair among methods have KTP results with significant difference.

5.2. Completeness of explainable model

This part shows the results of similarity between the original blackbox model and the generated explainable models from KT-GAN and KT-Bee. Two aspects were investigated including similarity of important features from SHAP values. In this work, we chose Jaccard similarity to calculate for similarity of feature importance. In addition, SHAP values are calculated to explain contribution of the features to the classification outcome. In similarity calculation, top-10 features based on feature importance score are selected to represent highest significant features from the classification models. The similarity score is calculated using (10) and the similarity results are given in Table 8.

Table 8. Result of Jaccard similarity for comparison of top-10 important feature similarity

Datasets	Data size	KT-Bee	KT-GAN
HI	50% less size	**0.6667**	0.0526
	Original size	**0.5385**	0.1111
	50% more size	**0.6667**	0.1111
JA	50% less size	**0.8182**	0.3333
	Original size	**0.6667**	0.1765
	50% more size	**0.6667**	0.1765
HLS	50% less size	**0.5385**	0.5385
	Original size	**0.5385**	0.4286
	50% more size	**0.6667**	0.5385
MT	50% less size	**0.6667**	0.4286
	Original size	**0.6667**	0.4286
	50% more size	**0.6667**	0.1111
MB	50% less size	**0.5385**	0.000
	Original size	**0.8182**	0.000
	50% more size	**0.6667**	0.000
NU	50% less size	**0.6667**	0.000
	Original size	**0.8182**	0.0900
	50% more size	**0.8182**	0.3333

The results in Table 8 show that explainable models generated from data synthesis method of KT-Bee are superior to those from KT-GAN in all datasets and settings. Furthermore, sizes of data inconsistently affect the similarity score, and this can be concluded that sizes of training dataset do not involve in model similarity. To analyze further, we selected a dataset as MB with SHAP value for comparison of top-10 features between explainable models and the teacher model (the original blackbox model) in Figure 6. As we found that KT-Bee had higher similarity, we thus show the mapping of same features with their SHAP value from explainable model of KT-Bee based on synthesized dataset size in Figure 7. The KT-Bee method demonstrates superior performance in transferring knowledge from a teacher model to a synthesized dataset compared to the KT-GAN method. This is evident in the higher KTP values achieved by the KT-Bee method. Consequently, when the synthesized dataset generated by the KT-Bee method is utilized to construct a proxy model for explaining the teacher model, the resulting proxy model exhibits importance features that closely resemble those of the teacher model. Figures 7(a) to 7(c), which depict the number of matching pairs among the top 10 importance features between the proxy model and the teacher model for the MB dataset. Figures 7(a) to 7(c) reveal 7, 8, and 9 matching pairs of importance features, respectively. The one-way analysis of variance (ANOVA) was conducted to determine if the difference in Jaccard similarity results from each method was significant or not. With Alpha value of 0.05, there was p-value of 1.68×10^{-10} which signified that at least one pair among methods have Jaccard similarity results with significant difference.

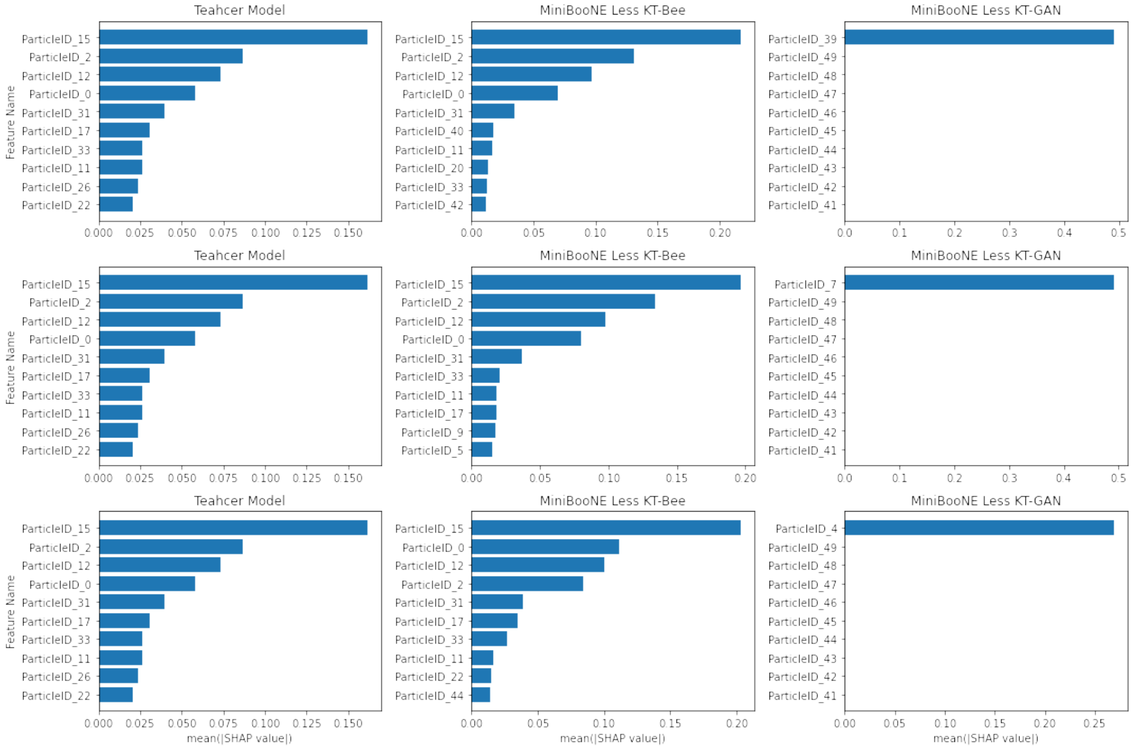
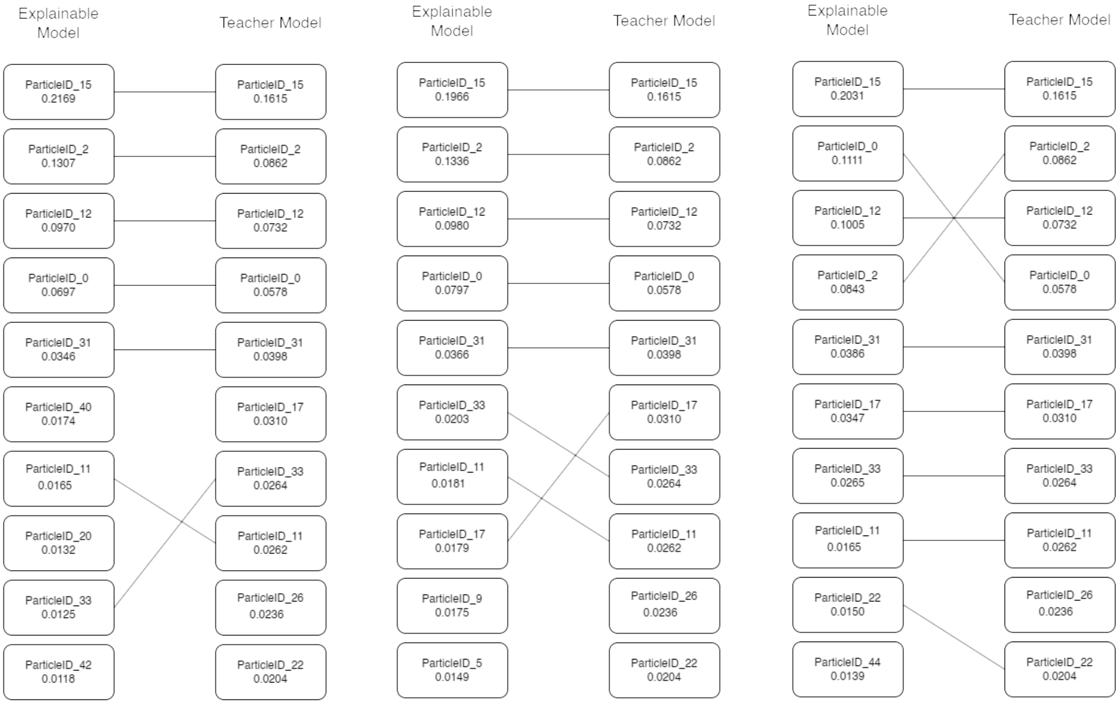


Figure 6. MiniBooNE (MB) feature importance score using SHAP value



(a) (b) (c)

Figure 7. Mapping the same features of 10-top features of MB dataset from the teacher model and explainable model from KT-bee based synthesized data whereas: (a) 50% less size, (b) original size, and (c) 50% more size

6. DISCUSSION

This study demonstrates the effectiveness of a Bee-inspired algorithm (KT-Bee) and GAN-based approach (KT-GAN) for KD in generating explainable models. While the existing approaches including GAN-based method can be applied to generate data to solve missing data or imbalanced data issues, they were not studied for their usage as transferring knowledge inspired by the concept of KD for generating explainable models from complex deep learning models. This work thus proposes to apply GAN-based approach and Bee-inspired algorithm to synthesize a dataset from complex models and trains the obtained dataset as an explainable model.

From the experiment, we investigate the knowledge transfer rate by examining distribution and accuracy results between the synthesized dataset and original dataset. Compared to a GAN-based approach (KT-GAN), KT-Bee achieved significantly higher knowledge transfer performance (98.85% vs. 62.8%). This translates to explainable models with features more closely resembling the original black-box model, as evidenced by the higher similarity score achieved by KT-Bee. These findings suggest that the BE's optimization process is more suitable at capturing the underlying relationships within the black-box model compared to GANs, which primarily focus on generating realistic data distributions.

While KT-Bee demonstrates superior performance, limitations related to its knowledge transfer process warrant further exploration. The method relies on regression to adjust feature values during data synthesis. The regression approach can lead to problematic synthesis with features with a vast range of possible values, as seen in the MiniBooNE dataset (Table 9). From the experiment, features like ”Particle ID 11” and ”Particle ID 22” have a significant gap between minimum and maximum values (-6.401 to 537.262 and 0.0 to 1428.59, respectively). Regression applied to such features might generate unrealistic values outside the expected range, impacting the quality of the synthesized data. In addition, KT-Bee's dependance solely on the teacher model's predictions for knowledge transfer introduces another limitation of the approach. If the teacher model struggles with imbalanced data or prioritizes specific classes, the transferred knowledge might be biased and lead to lower knowledge transfer performance. Since the proposed method does not have access to the original dataset for statistical analysis (e.g., class ratios and data distribution), it relies solely on the teacher model's knowledge, potentially hindering explainability.

To address these limitations, further study should focus on incorporating additional statistics like mean and standard deviation during feature value adjustment in KT-Bee in helping to tackle the issue of the vast range of features. This could provide a better context for the regression process and lead to more realistic synthesized data, especially for features with a wide value range. Furthermore, exploring alternative methods for knowledge transfer that go beyond relying solely on the teacher model's predictions could be a valuable area for future research. This might involve incorporating techniques that can analyze the original data distribution even when it's not directly accessible. By addressing these limitations, we can enhance the robustness and generalizability of the KT-Bee method for generating explainable models from complex black-box systems. Future research directions of this study also include investigating the impact of different BE variants on performance. Additionally, exploring various explainable model types beyond DTs could provide broader insights. Testing the method on real-world applications across diverse domains would further solidify its generalizability. By addressing these areas, we can refine the KT-Bee method and expand its applicability for XAI development.

In conclusion, the findings highlight the potential of Bee-inspired algorithms for explainability in AI for its ability to synthesize a dataset from a complex deep learning model. The synthesized dataset show great resemblance in terms of data distribution and similarity of classification results which indicate that the classifier model trained from synthesized dataset and the original model are similar. Thus, the comprehensible model from synthesized dataset based on Bee-inspired algorithm can be used as explainable model to the original complex model effectively.

Table 9. Example of statistical data for MiniBooNE dataset

	Particle ID 11		Particle ID 22	
	Original data	Synthesized data	Original data	Synthesized data
Mean	162.777	189.395	104.180	134.921
Std	116.686	257.096	102.901	354.792
Min	-6.401	-45.437	0.0	-181.183
Max	537.262	8340.629	1428.59	15119.142

7. CONCLUSION

This paper presents the method to generate an explainable model based on the given information of a well-preformed blackbox model. The proposed method is inspired by KD to transfer knowledge from a complex and incomprehensible model to synthesize a dataset to train for an explainable model. The proposed methods include the use of GAN and BE to synthesize a dataset by considering loss value in a knowledge transferring process to keep the significant features. The synthesized dataset then is used to train for an explainable prediction model as a proxy model using DT classification. The experiment involves 6 public datasets to generate datasets with 3 different sizes as an original size, a 50% less size, and a 50% more size. The results of the study signify that knowledge transfer from BE is better than GAN in terms of the coefficient of determination R2, as the Bee-based knowledge transfer achieved the lowest R2 of -0.895, and the biggest gap between the two methods is -0.692 in JA dataset. Moreover, explainable models from the synthesized data of the Bee-based method show higher F1 score than those from the GAN-based method in all datasets and settings, while F1 score from explainable models from the Bee-based method are closed to F1 score from the original blackbox model. The experimental results indicate that the size of the synthesized data does not affect the distribution of data since the data distribution is similar to the original dataset regardless of the size of the generated dataset. In terms of knowledge transfer performance, the Bee-based method achieves the higher score in average of 98.85% than the GAN-based method that yields an average of 62.8% accuracy score. The dataset synthesized from the Bee-based method produces the explainable prediction model that has similar top-10 features according to similarity score of 0.6718 using SHAP feature importance which is higher than those from GAN-based method for 0.4218 in average. Thus, it is conclusive that an explainable model using Bee-based data synthesis method is noticeably superior to the one from GAN-based data synthesis method. The proposed method is also proved to be usable to generate a proxy model to explain decision-making for a complex blackbox model, especially the case of inaccessibility of an original dataset but the blackbox model. Our findings highlight the potential of Bee-inspired algorithms as a powerful tool for XAI development. The ability to generate explainable models even when the original dataset is unavailable represents a significant advancement. This capability is particularly valuable in scenarios where data privacy concerns or limitations prevent access to the original data.

ACKNOWLEDGEMENTS

This work was supported by a Scholarship for research promotion student for international and education in Faculty of Science and Technology Thammasat University, Contract NO.TB 2/2563 and Thammasat University Research Unit in Data Innovation and Artificial Intelligence.

REFERENCES

[1] M. A. Gulum, C. M. Trombley, and M. Kantardzic, “A review of explainable deep learning cancer detection models in medical imaging,” *Applied Sciences*, vol. 11, no. 10, p. 4573, May 2021, doi: 10.3390/app11104573.

[2] T. Kooi, B. van Ginneken, N. Karssemeijer, and A. Den Heeten, “Discriminating solitary cysts from soft tissue lesions in mammography using a pretrained deep convolutional neural network,” *Medical Physics*, vol. 44, no. 3, pp. 1017–1027, 2017, doi: 10.1002/MP.12110.

[3] A. Akselrod-Ballin, L. Karlinsky, S. Alpert, S. Hasoul, R. Ben-Ari, and E. Barkan, “A region based convolutional network for tumor detection and classification in breast mammography,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10008 LNCS, 2016, pp. 197–205.

[4] X. Zhou *et al.*, “Automated assessment of breast tissue density in non-contrast 3D CT images without image segmentation based on a deep CNN,” *Medical Imaging 2017: Computer-Aided Diagnosis*, vol. 10134, p. 101342Q, 2017, doi: 10.1117/12.2254320.

[5] F. Gao *et al.*, “SD-CNN: a shallow-deep CNN for improved breast cancer diagnosis,” *Computerized Medical Imaging and Graphics*, vol. 70, pp. 53–62, 2018, doi: 10.1016/j.compmedimag.2018.09.004.

[6] J. Li, M. Fan, J. Zhang, and L. Li, “Discriminating between benign and malignant breast tumors using 3D convolutional neural network in dynamic contrast enhanced-MR images,” *Medical Imaging 2017: Imaging Informatics for Healthcare, Research, and Applications*, vol. 10138, p. 1013808, 2017, doi: 10.1117/12.2254716.

[7] A. S. Becker, M. Marcon, S. Ghafoor, M. C. Wurnig, T. Frauenfelder, and A. Boss, “Deep learning in mammography diagnostic accuracy of a multipurpose image analysis software in the detection of breast cancer,” *Investigative Radiology*, vol. 52, no. 7, pp. 434–440, 2017, doi: 10.1097/RLI.0000000000000358.

[8] F. Jiang *et al.*, “Artificial intelligence in healthcare: past, present and future,” *Stroke and Vascular Neurology*, vol. 2, no. 4, pp. 230–243, 2017, doi: 10.1136/svn-2017-000101.

[9] T. C. Chiang, Y. S. Huang, R. T. Chen, C. S. Huang, and R. F. Chang, “Tumor detection in automated breast ultrasound using 3-D CNN and prioritized candidate aggregation,” *IEEE Transactions on Medical Imaging*, vol. 38, no. 1, pp. 240–249, 2019, doi: 10.1109/TMI.2018.2860257.

[10] C. P. Langlotz *et al.*, "A roadmap for foundational research on artificial intelligence in medical imaging: from the 2018 NIH/RSNA/ACR/The Academy workshop," *Radiology*, vol. 291, no. 3, pp. 781–791, 2019, doi: 10.1148/radiol.2019190613.

[11] J. Gao, Q. Jiang, B. Zhou, and D. Chen, "Convolutional neural networks for computer-aided detection or diagnosis in medical image analysis: An overview," *Mathematical Biosciences and Engineering*, vol. 16, no. 6, pp. 6536–6561, 2019, doi: 10.3934/mbe.2019326.

[12] N. K. T Kooi *et al.*, "Large scale deep learning for computer aided detection of mammographic lesions," *Medical Image Analysis*, vol. 35, pp. 303–312, 2017.

[13] X. Bai *et al.*, "Explainable deep learning for efficient and robust pattern recognition: a survey of recent developments," *Pattern Recognition*, vol. 120, p. 108102, Dec. 2021, doi: 10.1016/j.patcog.2021.108102.

[14] "Ethics guidelines for trustworthy ai," *European Commission*, 2024. [Online]. Available: https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai (accessed Sep. 17, 2023).

[15] K. Kadam, S. Ahirrao, and K. Kotecha, "AHP validated literature review of forgery type dependent passive image forgery detection with explainable AI," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 5, pp. 4489–4501, 2021, doi: 10.11591/ijece.v11i5.pp4489-4501.

[16] M. Nauta *et al.*, "From anecdotal evidence to quantitative evaluation methods: a systematic review on evaluating explainable AI," *ACM Computing Surveys*, vol. 55, no. 13s, pp. 1–42, Dec. 2023, doi: 10.1145/3583558.

[17] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: an overview of interpretability of machine learning," in *Proceedings - 2018 IEEE 5th International Conference on Data Science and Advanced Analytics, DSAA 2018*, 2018, pp. 80–89, doi: 10.1109/DSAA.2018.00018.

[18] R. Saleem, B. Yuan, F. Kurugollu, A. Anjum, and L. Liu, "Explaining deep neural networks: a survey on the global interpretation methods," *Neurocomputing*, vol. 513, pp. 165–180, Nov. 2022, doi: 10.1016/j.neucom.2022.09.129.

[19] J. Angwin, J. Larson, S. Mattu, and L. Kirchner, "Machine bias," *ProPublica*. 2016.

[20] P. Baldi, P. Sadowski, and D. Whiteson, "Searching for exotic particles in high-energy physics with deep learning," *Nature Communications*, vol. 5, no. 1, p. 4308, Jul. 2014, doi: 10.1038/ncomms5308.

[21] I. Guyon *et al.*, "Analysis of the AutoML challenge series 2015–2018," *Automated Machine Learning*, vol. 177, pp. 177–219, 2019, doi: 10.1007/978-3-030-05318-5_10.

[22] J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders, "The Amsterdam library of object images," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 103–112, 2005, doi: 10.1023/B:VISI.0000042993.50813.60.

[23] Y. Pei, Y. Qu, and J. Zhang, "Self-boosting for feature distillation," in *IJCAI International Joint Conference on Artificial Intelligence*, 2021, pp. 945–951, doi: 10.24963/ijcai.2021/131.

[24] C. Wang, C. Xu, X. Yao, and D. Tao, "Evolutionary generative adversarial networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 6, pp. 921–934, 2019, doi: 10.1109/TEVC.2019.2895748.

[25] S. Addepalli, G. K. Nayak, A. Chakraborty, and R. V. Babu, "Degan: data-enriching gan for retrieving representative samples from a trained classifier," in *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, 2020, pp. 3130–3137, doi: 10.1609/aaai.v34i04.5709.

[26] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional GAN," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[27] D. T. Pham, A. Ghanbarzadeh, E. Koç, S. Otri, S. Rahim, and M. Zaidi, "The bees algorithm - a novel tool for complex optimisation problems," in *Intelligent Production Machines and Systems - 2nd I*PROMS Virtual International Conference 3-14 July 2006*, 2006, pp. 454–459, doi: 10.1016/B978-008045157-2/50081-X.

[28] D. T. Pham and M. Castellani, "The bees algorithm: modelling foraging behaviour to solve continuous optimization problems," in *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 2009, vol. 223, no. 12, pp. 2919–2938, doi: 10.1243/09544062JMES1494.

[29] K. Chungnoy, L. Paisarnworrapatch, A. Suebsriwichai, R. Kongkachandra, and P. Songmuamg, "Improving bees-based imputation using nearest neighbor for heuristic function in imputing data," in *ACM International Conference Proceeding Series*, 2019, pp. 20–25, doi: 10.1145/3375959.3375974.

[30] C. N. Coelho *et al.*, "Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors," *Nature Machine Intelligence*, vol. 3, no. 8, pp. 675–686, 2021, doi: 10.1038/s42256-021-00356-5.

[31] B. Hawks, J. Duarte, N. J. Fraser, A. Pappalardo, N. Tran, and Y. Umuroglu, "Ps and Qs: quantization-aware pruning for efficient low latency neural network inference," *Frontiers in Artificial Intelligence*, vol. 4, Jul. 2021, doi: 10.3389/frai.2021.676564.

[32] R. K. Bock *et al.*, "Methods for multidimensional event classification: a case study using images from a Cherenkov gamma-ray telescope," *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 516, no. 2–3, pp. 511–528, 2004, doi: 10.1016/j.nima.2003.08.157.

[33] B. P. Roe, H. J. Yang, J. Zhu, Y. Liu, I. Stancu, and G. McGregor, "Boosted decision trees as an alternative to artificial neural networks for particle identification," *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 543, no. 2–3, pp. 577–584, 2005, doi: 10.1016/j.nima.2004.12.018.

[34] "Encrypted stock market data from numerai," Kaggle.com, 2016. [Online]. Available: https://www.kaggle.com/datasets/numerai/encrypted-stock-market-data-from-numerai (accessed Sep. 17, 2023).

[35] L. Buitinck *et al.*, "Scikit-learn: machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

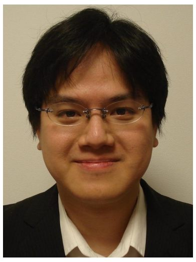
[36] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in Neural Information Processing Systems*, pp. 4765–4774, 2017.

[37] T. Suresh, T. A. Assegie, S. Ganesan, R. L. Tulasi, R. Mothukuri, and A. O. Salau, "Explainable extreme boosting model for breast cancer diagnosis," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 5, pp. 5764–5769, 2023, doi: 10.11591/ijece.v13i5.pp5764-5769.

BIOGRAPHIES OF AUTHORS

Kritanat Chungnoy holds both a Bachelor's and Master's degree in Computer Science from Thammasat University. Currently, he is pursuing a Ph.D. in Computer Science at the same university. In addition to his academic endeavors, Kritanat is actively engaged in programming and event management at Planter Group Co., Ltd. His research interests encompass artificial intelligence, optimization utilizing BE, and explainable AI. He can be contacted at email: kittanutc@gmail.com.

Tanatorn Tanantong received the B.Eng. and M.Eng. degrees in computer engineering from Suranaree University of Technology, Thailand,in 2005 and 2008, respectively, and the Ph.D. degree in computer science from the Sirindhorn International Institute of Technology (SIIT), Thammasat University, Thailand, in 2015, through scholarship from Thailand Research Fund under The Royal Golden Jubilee Ph.D. Program. He is currently an Associate Professor with the Department of Computer Science (CS), Faculty of Science and Technology, Thammasat University. Furthermore, he as the Head of Thammasat University Research Unit in Data Innovation and Artificial Intelligence. In 2016, he contributed as a Visiting Professor with the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, USA. Additionally, in 2017, he gained experience as a Postdoctoral Researcher with Japan Advanced Institute of Science and Technology, Japan. His research interests include artificial intelligence, data mining, machine learning, medical informatics, and hospital information systems. For further communication, he can be contacted at email: tanatorn@sci.tu.ac.th.



Pokpong Songmuang received the B.Eng degree from Thammasat University in 2003, the M.Eng degree from Nagaoka University of Technology in 2006, and the Ph.D. degree in Computer Science from the University of Electro-Communications in 2010. He was an Assistant Professor at Waseda University. He is currently an Assistant Professor at Thammasat University. His research interests include optimization algorithms, e-testing, social network analytics, data mining, and education technology. He can be contacted at email: pokpongs@tu.ac.th.