Communication Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 95–101
DOI: 10.15439/2025F6104
ISSN 2300-5963 ACSIS, Vol. 45

A Framework for Machine Unlearning Using Selective Knowledge Distillation into Soft Decision Tree

Sangmin Kim
Chung-Ang University,
84, Heukseok-ro, Dongjak-gu,
06974 Seoul, South Korea
Email: kimddol98@cau.ac.kr

Byeongcheon Lee
Chung-Ang University,
84, Heukseok-ro, Dongjak-gu,
06974 Seoul, South Korea
Email: qudcjs0208@cau.ac.kr

Sungwoo Park
Chung-Ang University,
84, Heukseok-ro, Dongjak-gu,
06974 Seoul, South Korea
Email: psw5574@cau.ac.kr

Miyoung Lee
Chung-Ang University,
84, Heukseok-ro, Dongjak-gu,
06974 Seoul, South Korea
Email: miylee@cau.ac.kr

Seungmin Rho
Chung-Ang University,
84, Heukseok-ro, Dongjak-gu,
06974 Seoul, South Korea
Email: smrho@cau.ac.kr

***Abstract*—With growing privacy regulations, removing user-related information from machine learning models has become essential. Machine unlearning addresses this by enabling selective removal of learned information, but most existing methods rely on deep learning models, which are computationally expensive and lack interpretability. To overcome these limitations, we propose a novel machine unlearning framework using selective knowledge distillation into a Soft Decision Tree (SDT). A convolutional neural network (ConvNet) is first trained to generate soft labels and intermediate features, which are transferred to the SDT. During distillation, an unlearning algorithm adjusts specific leaf node distributions and routing weights using soft redistribution and path pruning. This enables class-specific forgetting without retraining and preserves accuracy on non-target classes. Experiments on MNIST and CIFAR-10 demonstrate that our framework effectively removes class-specific knowledge while maintaining overall model performance. The interpretable SDT structure also allows for clear visualization of model changes before and after unlearning.**

***Index Terms*—Machine Unlearning, Knowledge Distillation, Convolutional Neural Network, Soft Decision Tree, Privacy.**

I. Introduction

IN RECENT years, data protection regulations aimed at protecting user privacy and increasing personal data control have been introduced and enforced around the world. For example, the General Data Protection Regulation (GDPR) in the European Union (EU) [1] and the California Consumer Privacy Act (CCPA) in the United States [2] impose strict legal obligations on organizations to delete personal information when requested by users. These regulations demonstrate the growing social and legal recognition of the "right to be forgotten" and make it clear that individuals can demand the removal of their digital traces. Accordingly, there is a growing need for a technical solution that can effectively reflect an individual's deletion request, even in already trained machine learning models. As a solution, machine unlearning is gaining much attention [3]. This technology enables models to comply with the latest privacy regulations by selectively removing the impact of specific data that has already been learned.

Most existing machine unlearning approaches have mainly focused on deep learning (DL) models and are typically implemented by retraining the entire model or mitigating the influence of specific data through methods such as gradient ascent or fine-tuning. However, these approaches are costly in terms of computations, and there is a risk of inadvertently affecting non-target data [4, 5]. In such cases, useful learned representations are unintentionally changed, resulting in a reduction of the model's stability and generalizability. In addition, these limitations are further exacerbated in DL models. It is difficult to selectively remove the influence of specific data from a deep learning model's complex and distributed internal representation because even minor adjustments to the representation can unintentionally change the overall result of the model [6]. Furthermore, the opacity of the internal representation complicates the verification of the removal of specific information.

Recently, machine unlearning research on traditional machine learning models has also been actively conducted, especially for tree-based models such as gradient boosting decision tree (GBDT) [7, 8] and random forest (RF) [9, 10, 11]. The advantage of these tree-based models is that the decision process is clear and easy to interpret, and data deletion can be easily verified. However, since these models are primarily designed for discrete, low-dimensional input data, they are not effective for handling high-dimensional continuous data, such as images. In addition, machine unlearning for the tree-based models requires the explicit maintenance of instance-leaf mappings during both training and inference, which can lead to high memory and computational overhead. Furthermore, although tree-based

models are effective for verifying instance-level deletion, they are less suitable for class-level unlearning, where an entire semantic category must be removed. In practical scenarios, such class-level deletion requests frequently occur due to policy revisions, semantic redefinitions, or ethical and legal considerations. Class-level unlearning is gaining attention as a necessary capability in modern machine learning systems, where models are expected to respond to changing category definitions and increasing regulatory demands [12].

To address these limitations, we propose a machine unlearning framework using selective knowledge distillation that can preserve the generalization performance of the original DL model. In the proposed framework, knowledge containing soft labels and intermediate features is extracted from a high-performing convolutional neural network (ConvNet) and transferred to a soft decision tree (SDT). In addition, the proposed unlearning algorithm adjusts the routing probabilities and class distributions of leaf nodes in SDT to selectively suppress information related to the target class while preserving overall model accuracy. The main contributions of this study are as follows:

- Instead of retraining the entire model, we propose an efficient machine unlearning framework that selectively adjusts only routing probabilities and class distributions of specific leaf nodes in SDT. This enables the selective removal of information on target classes while maintaining stable classification performance on non-target classes.
- By utilizing the tree-based structure in which both the branching decisions and leaf node class distributions are explicitly interpretable, the proposed framework enables visual analysis of model changes before and after unlearning.
- The proposed framework was evaluated using multiple image benchmark datasets MNIST and CiFAR-10, and the results showed robust unlearning effectiveness and generalization performance.

II. Preliminaries

A. Machine Unlearning

Machine unlearning refers to the process of removing or weakening the influence of specific training data within a trained model. A learning algorithm is formally defined as a function A: D→ H, where D is a dataset and H is a hypothesis space. In summary, the algorithm A returns a model A(D) ∈ H, which is trained on the dataset D. Unlearning is performed through a removal R: (A(D), D, (x, y)) → H, which takes as inputs the trained model A(D), the original dataset D, and a data instance (x, y) to remove. Exact machine unlearning refers to the ideal scenario in which the resulting model is indistinguishable from one trained from scratch on the dataset with (x, y) removed [13]. This condition is formally expressed as:

$$R\big(A(D), D, (x,y)\big) = A(D \setminus \{(x,y)\}) \quad (1)$$

B. Knowledge Distillation

Knowledge distillation is a technique for model compression and optimization that transfers the knowledge learned by a large and complex teacher model to a small and simple student model so that the student model performs as well as the teacher model [14, 15]. Typically, knowledge distillation utilizes soft labels (probability distributions for each class) produced by the teacher model to train the student model. Soft labels contain more information than hard labels (1 for the correct class and 0 for the rest), such as similarity relationships and uncertainty between classes. In order to leverage additional information from the output distribution of the teacher model, a temperature parameter T is incorporated into the softmax function to smooth the output distribution and emphasize class similarity. The smoothed probability for class i, denoted as q_i, is calculated as follows:

$$q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)} \quad (2)$$

where z_i is the logit for class i, and T controls the smoothness of the distribution. A higher temperature leads to softer probability outputs, allowing the student to capture the teacher's nuanced generalization behavior.

III. Methodology

A. Datasets

In this study, we utilize two widely used benchmark image classification datasets, MNIST [16] and CIFAR-10 [17], in our experiment. The MNIST dataset consists of grayscale images, while CIFAR-10 dataset is composed of color images. Each dataset is divided into train, validation, and test sets to facilitate both model training and evaluation. The number of samples in each subset is summarized in Table I and an example of each dataset is shown in Fig. 1.

TABLE I
CONSTRUCTION OF DATASETS

	Datasets	
	MNIST	**CIFAR-10**
Train	50,000	40,000
Validation	10,000	10,000
Test	10,000	10,000

Fig 1. Example of MNIST and CIFAR-10 dataset.

To ensure consistent input scaling and improve training stability, we apply different preprocessing strategies for each dataset. For the MNIST dataset, we apply Min-Max normalization to scale pixel values to the [0, 1] range. For the CIFAR-10 dataset, we perform Z-score normalization by subtracting the mean and dividing by the standard deviation of each RGB channel. As a final preprocessing step, all target labels are converted into one-hot encoded vectors to support the use of the cross-entropy loss function and enable soft label distillation in subsequent stages.

B. Model Design

This section presents the architecture and training procedures of the two main components in our framework: a ConvNet that acts as the teacher model and SDT is trained as a student model through knowledge distillation.

a) ConvNet: To generate soft labels for knowledge distillation, we design two ConvNet architectures optimized respectively for the MNIST dataset and the CIFAR-10 dataset.

For the MNIST dataset, we implement a compact ConvNet composed of two convolutional blocks. Each block consists of two 3×3 convolutional layers with ReLU activation and the same padding, followed by 2×2 max pooling and dropout. The extracted features are then flattened and passed through a fully connected layer with 512 units, followed by a softmax output layer. The model is trained using the Adam optimizer with a learning rate of 3×10^{-4}, and categorical cross-entropy is used as the loss function.

For the CIFAR-10 dataset, we adopt a deeper ConvNet consisting of three convolutional blocks. Each block contains two 3×3 convolutional layers with ReLU activation, batch normalization, L2 regularization, max pooling, and dropout. The number of filters increases with depth to capture complex spatial features. The final feature map is flattened and passed through a dense layer with 128 units and dropout, followed by a softmax classification layer. The model is trained using the Adam optimizer with an initial learning rate of 1×10^{-3}, along with learning rate scheduling and data augmentation via the ImageDataGenerator framework to enhance generalization.

b) SDT: The SDT is designed to replicate the predictive behavior of a ConvNet trained within this framework while offering a transparent decision-making process based on explicit branching rules [18]. The operational structure of the SDT is illustrated in Fig. 2, which depicts a simple configuration consisting of one internal node and two leaf nodes.

We design two SDT configurations optimized respectively for the MNIST and CIFAR-10 datasets. For the MNIST dataset, which consists of grayscale images, the output from the final convolutional layer of the ConvNet is flattened and used as input to the SDT. In contrast, for the CIFAR-10 dataset, which contains color images, simple flattening may lead to loss of spatial information. To address this, the output of the final max-pooling layer is passed through global average pooling to produce a semantically meaningful one-dimensional feature vector. These feature vectors, along with soft labels generated by the ConvNet, are used to train the SDT. To further enhance representational capacity and training stability, we introduce four key hyperparameters:

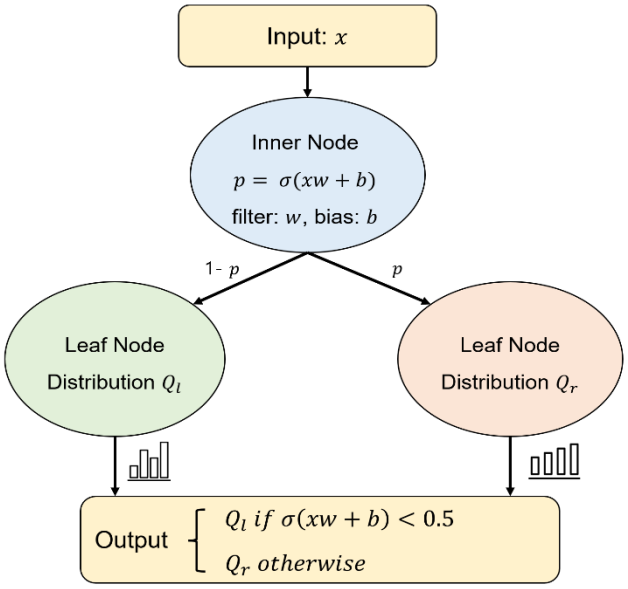

Fig 2. Simple configuration of SDT.

- Penalty strength: Controls the strength of a regularization term that prevents internal nodes from consistently branching in the same direction. Higher values promote more balanced left-right splits, which increases structural diversity and tree utilization.
- Penalty decay: Gradually reduces the influence of the regularization term as the node depth increases. This allows shallow nodes to learn general decision rules, while deeper nodes specialize in more fine-grained distinctions.
- Exponential moving average (EMA) window size: The branching direction of internal nodes may fluctuate across mini-batches, which can potentially destabilize learning. We apply an EMA to smooth routing behavior over time. A larger window emphasizes long-term stability, while a smaller window allows quicker adaptation at the cost of higher variance.
- Inverse temperature (β): Adjusts the sharpness of the sigmoid function used at internal nodes to make branching decisions more decisive. A higher β makes splits more decisive, with probabilities closer to 0 or 1, making the model behave more like a hard decision tree.

Given the sensitivity of these hyperparameters to model performance, we use the Optuna framework for hyperparameter optimization [19]. The search space includes tree depths and learning rate in addition to the four parameters above. The optimal configuration is selected based on

validation accuracy. As a result, the SDT trained under this framework successfully emulates the predictions of the teacher model while maintaining interpretability through its hierarchical and transparent decision structure.

C. Unlearning Algorithm

In this study, we propose a novel unlearning algorithm designed to selectively remove knowledge relevant to a particular class from a trained SDT model. The method utilizes two primary mechanisms, soft redistribution and path pruning, to locally adjust a subset of model parameters, enabling class-level forgetting while preserving the overall structure and performance of the model. The proposed unlearning algorithm is presented in Algorithm 1.

The process of the algorithm is described as follows:

- 1) Identify the target leaf node: Find leaf nodes whose predicted class distribution is dominated by the target class c to be forgotten.
- 2) Redistribute class probabilities: Take a fraction (α) of the class c probability of the target leaf node and distribute it to the top k leaf nodes with high cosine similarity. The proportion of the distribution is proportional to the similarity and is only distributed to leaves that do not strongly predict the target class.
- 3) Soft Pruning: The internal node weights of the decision path to each target leaf are attenuated according to the remaining class c probability (ω) and pruning ratio (γ).
- 4) Residual Suppression: If class c still has the highest probability in all leaves, set its probability to zero and re-normalize the distribution.

Algorithm 1 Target Class Unlearning

Require: Trained tree model T, session S, target class c, top-k, redistribution rate α, pruning rate γ
Ensure: Modified tree with target class c forgotten
Retrieve all leaf nodes L and obtain $\phi_\ell \leftarrow S(\phi_\ell)$ for each $\ell \in L$
$L_{target} \leftarrow \{\ell \in L \mid \text{argmax}(\phi_\ell) = c\}$
for all $\ell t \in L_{target}$ **do**
    Find top-k most similar leaves $N(\ell_t)$ using cosine similarity
        $\delta \leftarrow \phi_{\ell t}[c] \cdot \alpha$
        $\phi_{\ell t}[c] \leftarrow \phi_{\ell t}[c] \cdot (1 - \alpha)$
        for all $\ell_j \in N(\ell_t)$ **do**
            $\phi_{\ell j}[c] \leftarrow \phi_{\ell j}[c] + \delta \cdot \text{sim}(\ell_t, \ell_j)$
            Normalize $\phi_{\ell j}$ and update in session
        end for
        Normalize $\phi_{\ell t}$ and update in session
        Compute importance $\omega \leftarrow \phi_{\ell t}[c]$
        Compute pruning factor $\rho \leftarrow 1 - \gamma \cdot \omega$
        Identify internal nodes P on the path to ℓ_t
        for all $n \in P$ **do**
            Weaken weight: $w_n \leftarrow \rho \cdot w_n$
        end for
end for
for all $\ell \in L$ **do**
    if $\phi_\ell[c] > \epsilon$ **then** ▷ if target class remains dominant
        Set $\phi_\ell[c] \leftarrow 0$, normalize, and update in session
    end if
end for

A. Experiment Setup

All experiments were conducted on a system running Windows 11, equipped with an Intel Core i7-12700K CPU, an NVIDIA GeForce RTX 4080 GPU (16GB), and 64 GB RAM. Python 3.7 was used as the development environment.

IV. Experiment

This section presents the empirical evaluation of the proposed framework. For each dataset, we present the classification performance when performing general knowledge distillation and the performance of the proposed framework.

We evaluated the proposed framework on two benchmark image classification datasets: MNIST and CIFAR-10. As described in Section 3, each dataset was preprocessed using standard procedures. Both ConvNet and SDT model were trained and evaluated based on accuracy, precision, recall, and F1-score. To ensure fair and stable comparisons, all training processes incorporated early stopping to prevent overfitting and reduce unnecessary training time. For SDT models after unlearning, we conducted a focused evaluation comparing the

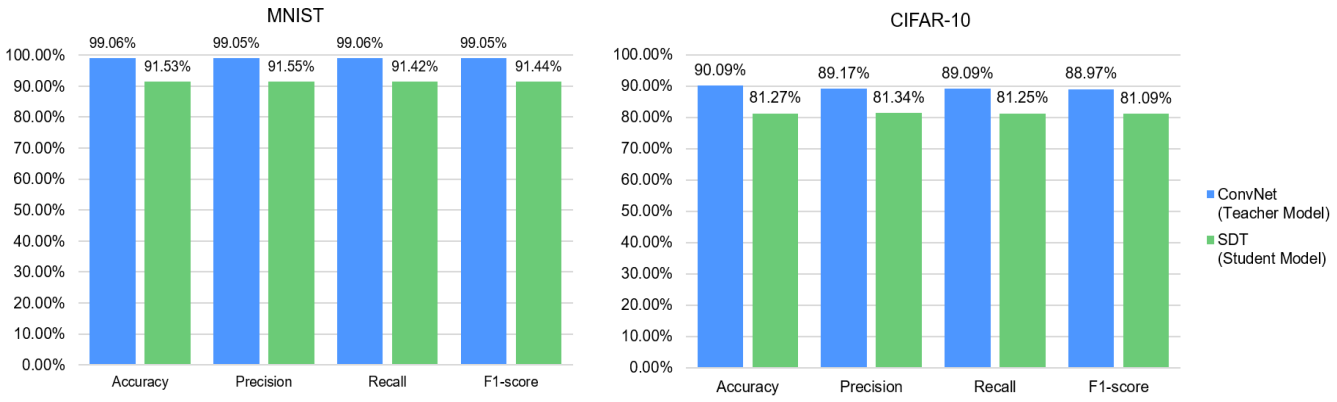

Fig 3. Performance of ConvNet and SDT.

TABLE II
PERFORMANCE AND TRAINING TIME COMPARISON BETWEEN PROPOSED FRAMEWORK AND RETRAINING

Datasets	Method	Test Accuracy	Target Accuracy	Non-Target Accuracy	Time
MNIST	Proposed Framework	80.74%	0%	89.81%	33m
	Retrain	80.62%		89.68%	51m 15s
CiFAR-10	Proposed Framework	71.12%	0%	79.02%	44m 23s
	Retrain	71.51%		79.46%	96m 20s

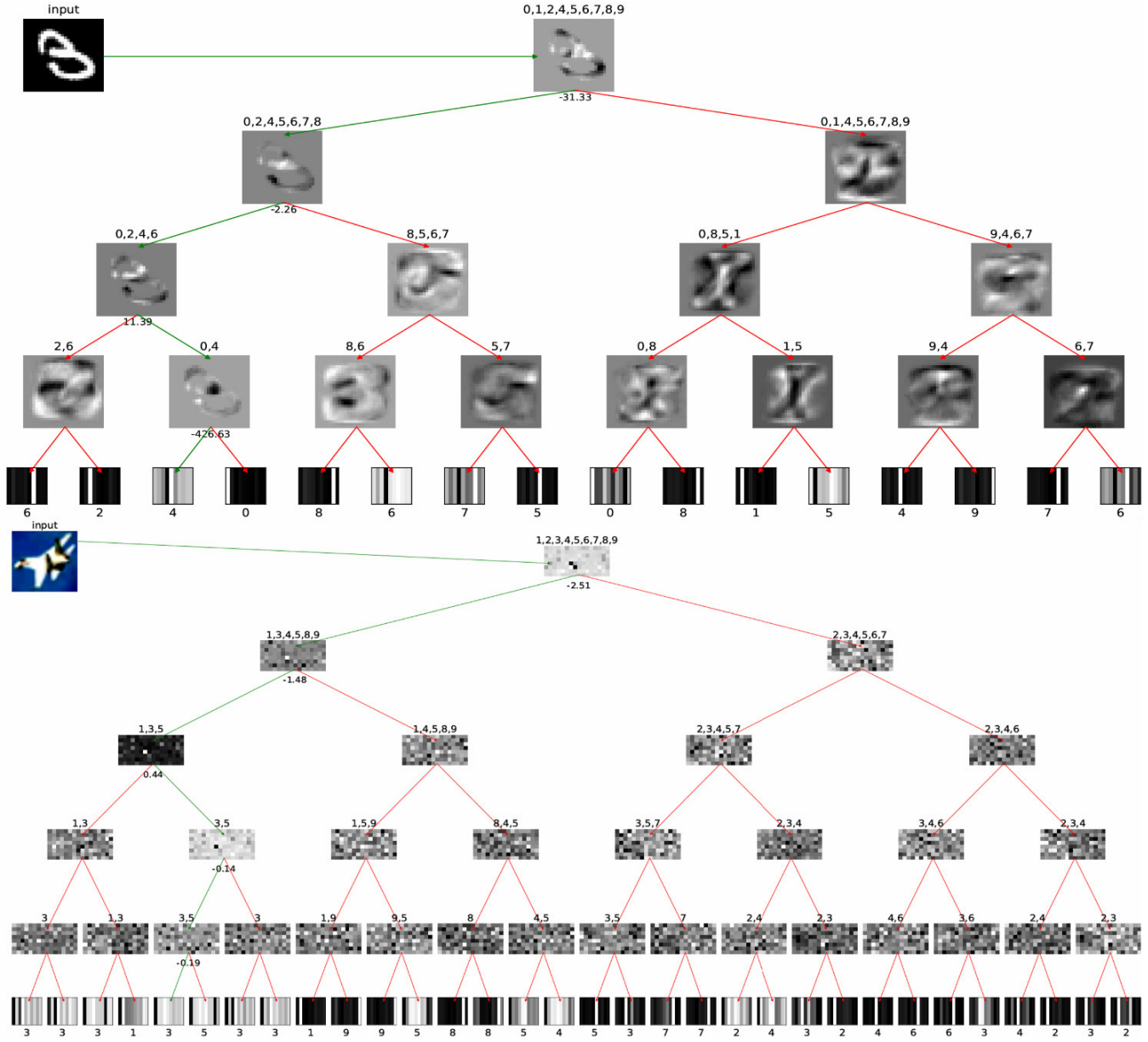

Fig 4. SDT after unlearning: (Top) MNIST dataset (target class 3), (Bottom) CIFAR-10 dataset (target class 0).

baseline approach of retraining after removing the target class with the proposed framework. In both cases, we analyzed the accuracy of target and non-target classes to assess the effectiveness of selective forgetting.

B. Performance Evaluation of Knowledge Distillation

To verify the suitability of ConvNet as the teacher model and to assess the effectiveness of knowledge distillation to the student model (SDT), we compared the classification performance of both models on MNIST and CIFAR-10 using four metrics: accuracy, precision, recall, and F1-score. Detailed metric comparisons for each dataset are visualized in Fig. 3. As shown in Fig. 3, the ConvNet achieved very high performance, with accuracy, precision, recall, and F1-score all exceeding 99% on MNIST and over 90% on CIFAR-10, confirming its appropriateness as a teacher model. In comparison, the SDT showed a consistent performance gap, with accuracy and other metrics approximately 8% lower than

those of ConvNet on both datasets. While this indicates some loss in predictive power due to the simpler and more interpretable structure of the SDT, the results still demonstrate that SDT can inherit a significant portion of the teacher model's knowledge through distillation.

C. Performance Evaluation of Proposed Framework

To evaluate the effectiveness of the proposed unlearning framework, we applied the algorithm described in Section 3 to SDT models distilled from teacher models trained on each dataset. For each dataset, a specific target class was selected for removal, and we assessed how effectively the SDT could forget the target class while preserving predictive performance on non-target classes. As a baseline for comparison, we adopted a conventional retraining approach where all training samples of the target class were removed from the dataset, and then both the teacher and student models were fully retrained from scratch. The evaluation was based on three key metrics: (1) overall classification accuracy on the test set, (2) accuracy on samples belonging to the target class, and (3) accuracy on samples from non-target classes. As summarized in Table II, both methods successfully removed the model's ability to predict the target class, achieving 0% accuracy in all cases. However, the proposed method consistently retained comparable or higher accuracy on non-target classes, demonstrating its effectiveness in preserving useful knowledge.

In addition to predictive performance, our framework offers a substantial advantage in training efficiency. Unlike the baseline, which involves retraining the entire pipeline, our method applies unlearning directly to the distilled student model without reinitializing the teacher. Although the total training time from teacher to student is comparable across some datasets, omitting the costly teacher retraining step significantly reduces overall computational overhead. This makes the proposed framework a practical and efficient solution, particularly in resource-constrained environments. Fig. 4 shows the structural changes in the SDT after unlearning the target classes. The top section presents an MNIST example where digit 3 was selected for removal. After unlearning, the SDT adjusts its internal decision paths, rerouting inputs that were previously associated with digit 3 toward alterative classes. In the visualization, the leaf nodes that initially had high confidence for class 3 now show redistributed probabilities favoring digits such as 4 and 6. Similarly, the bottom section shows a CIFAR-10 case where the airplane (class 0) class was removed. The SDT modifies its structure to suppress branches linked to the removed class and shifts decision confidence toward other classes. These structural adjustments both in internal routing and leaf-level distributions demonstrate how the SDT reflects the unlearning objective at the model level.

II. Conclusions

This study proposed an interpretable and selective machine unlearning framework based on knowledge distillation from a ConvNet to an SDT. The SDT is trained using soft labels and intermediate features extracted from the teacher model, providing critical advantages in interpretability and modularity essential for efficient unlearning. To facilitate class-level forgetting, we propose an unlearning algorithm that integrates soft redistribution and path pruning, enabling targeted suppression of class-specific information. Unlike conventional retraining-based methods, our framework applies unlearning directly to the distilled student model without retraining the teacher model, providing a more efficient and practical alternative.

Experimental results on two benchmark image classification datasets, MNIST and CIFAR-10, demonstrate that the proposed framework maintains high classification accuracy while effectively removing the influence of the designated target class. After unlearning, target class accuracy dropped as intended, while performance on non-target classes remained stable, validating the framework's selective forgetting capability. Furthermore, visualizations of the SDT before and after unlearning confirmed structural changes, including rerouted decision paths and updated leaf node distributions that reduced the model's reliance on the target class. By presenting an interpretable unlearning framework applicable to high-dimensional input data, this study contributes to the field of machine unlearning. Future research will focus on expanding the proposed framework in two main directions. First, we aim to extend the current class-level unlearning approach to support instance-level unlearning, enabling more granular control over the forgetting process. This will require developing fine-tuned strategies for identifying and suppressing the influence of individual training samples within the trees structure. Second, we plan to evaluate the scalability and robustness of the framework on more complex and high-resolution image datasets, such as STL-10 or ImageNet. These experiments will test whether the selective forgetting and interpretability properties of the SDT-based student model are preserved under more challenging data conditions. Consequently, these directions aim to enhance the generalizability and practicality of interpretable machine unlearning.

Acknowledgment

This research was partly supported by the MSIT (Ministry of Science and ICT), Korea, under the Convergence security core talent training business support program (IITP-2025-RS2023-00266605) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation), the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2025-00518960) and the "Regional Innovation System & Education (RISE)" through the Seoul RISE Center, funded by the Ministry of Education (MOE) and the Seoul Metropolitan Government.

References

[1] EU. Regulation (eu) 2016/679. https://eur-lex.europa.eu/eli/reg/2016/679/oj, 2016. [Online; accessed 16-April-2020].

[2] California. California consumer privacy act. https://leginfo.legislature.-ca.gov/ faces/billTextClient.xhtml?bill_id= 201720180AB375, 2018. [Online; accessed 16-April 2020].

[3] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, ... & N. Papernot, "Machine unlearning," In *2021 IEEE symposium on security and privacy (SP)*, pp.141-159, May 2021.

[4] L. Wang, T. Chen, W. Yuan, X. Zeng, K. F. Wong, & H. Yin, "KGA: A general machine unlearning framework based on knowledge gap alignment," arXiv preprint arXiv:2305.06545.

[5] A. Sekhari, J. Acharya, G. Kamath, & A. T. Suresh, "Remember what you want to forget: Algorithms for machine unlearning," Advances in Neural Information Processing Systems, vol. 34, pp. 18075-18086, 2021.

[6] J. Li, Y. Li, X. Xiang, S. T. Xia, S. Dong & Y. Cai, "TNT: An interpretable tree-network-tree learning framework using knowledge distillation," Entropy, vol. 22, pp. 1203, 2020.

[7] H. Lin, J. W. Chung, Y. Lao, & W. Zhao, "Machine unlearning in gradient boosting decision trees," In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp.1374-1383, August 2023.

[8] Z. Wu, J. Zhu, Q. Li, & B. He, "Deltaboost: Gradient boosting decision trees with efficient machine unlearning," Proceeding of the ACM on Management of Data, Vol. 1, pp. 1-26, 2023.

[9] J. Brophy, & D. Lowd, "Machine unlearning for random forests," In International Conference on Machine Learning, PMLR, pp.1092-1104, July 2021.

[10] T. Surve, & R. Pradhan, "Example-based Explanations for Random Forests using Machine Unlearning," arXiv preprint arXiv:2402.05007.

[11] S. Wang, Z. Shen, X. Qiao, T. Zhang, & M. Zhang, "DynFrs: An Efficient Framework for Machine Unlearning in Random Forest," arXiv preprint arXiv: 2410.01588.

[12] Z. Zuo, Z. Tang, B. Wang, K. Li, & A. Datta, "Ecil-mu: Embedding based class incremental learning and machine unlearning," IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6275-6279, 2024.

[13] J. Xu, Z. Wu, C. Wang, & X. Jia, "Machine unlearning: Solutions and challenges," IEEE Transactions on Emerging Topics in Computational Intelligence.

[14] J. Gou, B. Yu, S. J. Maybank & D. Tao, "Knowledge distillation: A survey," International Journal of Computer Vision, vol. 129, pp. 1789-1819, 2021.

[15] G. Hinton, O. Vinyals, & J. Dean, "Distilling the knowledge in a neural network," arXiv preprint arXiv:1503.02531, 2015.

[16] L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," IEEE signal processing magazine, vol. 29, pp.141-142, 2012.

[17] A. Krizhevsky, & G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[18] N. Frosst, & G. Hinton, "Distilling a neural network into a soft decision tree," arXiv preprint arXiv:1711.09784, 2017.

[19] T. Akiba, S. Sano, T. Yanase, T. Ohta, & M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pp.2623-2631, July 2019.