A Study on Training Data Influence for Identifying Inaccurate Instances

Saneyasu Yamaguchi
Department of Information and Communications Engineering
Kogakuin University
Tokyo, Japan
sane@cc.kogakuin.ac.jp

Shido Hosono
Electrical Engineering and Electronics Kogakuin University Graduate School
Tokyo, Japan
cm25071@g.kogakuin.jp

***Abstract*—Understanding how individual training instances influence model predictions is essential for improving data quality and enhancing the performance of deep learning models. Although prior studies have proposed influence-estimation methods such as TracIn and influence functions, the relationship between instance quality and influence scores remains insufficiently explored, despite its importance for detecting unsuitable training data. In this study, we investigate how incorrect labels affect influence estimates by intentionally generating mislabeled training instances and analyzing their TracIn score distributions. Our results reveal three characteristic distribution patterns—S-shaped, Z-shaped, and backslash-shaped—that determine where incorrectly labeled instances tend to appear within the score distribution. We further show that these patterns correspond to the degree of similarity between test instances and the training set, suggesting that the distribution shape reflects the underlying data geometry. In addition, we evaluate how the number of unsuitable instances impacts model performance and demonstrate that a small number of incorrectly labeled instances causes only limited degradation, implying a threshold effect. These findings provide insights into how influence estimation can reveal the behavior of unsuitable training instances, suggesting a data-cleaning strategies and guide the construction of higher-quality training datasets.**

Keywords—Machine learning, training data quality, influence estimation, TracIn, mislabeled data, interpretability, deep learning, data cleaning.

I. Introduction

Improving the quality of training data is a crucial factor in advancing the accuracy of modern AI systems. A number of studies have proposed methods that analyze the rationale behind an AI model's inference in order to identify unsuitable training instances [1][2][3]. These methods estimate the contribution of each training instance to a model's inference. For these methods to be effective, clarifying the relationship between the quality of individual instances and their influence scores is essential. However, this relationship has not yet been thoroughly investigated.

In this paper, we aim to clarify this relationship by focusing on TracIn [1], one of the most widely used methods for estimating the influence of each training instance on model inference. To explore how training-instance quality is reflected in influence scores, we generate inaccurate training instances by intentionally changing correct labels to incorrect ones and analyze their corresponding influence scores. We then identify trends in these scores that vary depending on the types of instances. Our analysis highlights consistent trends in how mislabeled instances appear within the score distribution, revealing structural behaviors dependent on the characteristics of the test instance.

The findings from this study not only deepen our understanding of the relationship between instance quality and influence estimation but also provide insights that can contribute to the development of more effective training-data cleaning methods and the construction of higher-quality datasets.

II. Related Work

A. *Providing Interpretability on AI's Inference*

Research on explaining the decision-making process of deep learning models has grown substantially in recent years. Early studies highlighted that deep neural networks often function as "black boxes," making it difficult to understand why a model reaches a particular output. Tulio et al. [4] and Montavon et al. [5] emphasized that the lack of interpretability raises concerns in domains where explanations are indispensable, such as judicial decision-making or policy decisions that affect the public.

Tulio et al. illustrated this issue using a model trained to distinguish between huskies and wolves [4]. Their analysis demonstrated that the classifier relied not on the animals themselves but on background cues, labeling images with snowy backgrounds as wolves. They referred to this phenomenon as a "bad model." To address such interpretability problems, Tulio et al. introduced LIME [4], a method that highlights which parts of the input most strongly influence the model's prediction.

Other approaches have also been proposed. Simonyan et al. developed a gradient-based visualization method to identify influential pixels in image-classification tasks [6]. Smilkov et al. proposed SmoothGrad [7], which improves gradient-based explanations by reducing noise. These methods estimated how perturbations in input features affect model outputs, enabling identification of regions that contribute most to a prediction.

Prior to the widespread use of deep learning, simpler interpretable methods had already been explored. For example, Shirataki et al. proposed a method for interpreting the decision boundaries of support vector machines [8].

In addition to these works on explaining predictions using test data, several studies have also examined methods for extracting decision bases from training data, which directly connect model behavior to the data that shaped it.

B. Influence Estimation from Training Data

Koh et al. [2] proposed a method for identifying the training points that are most responsible for a given prediction, without retraining the model. Their method uses influence functions, which is a classical technique from robust statistics. It traces a model's prediction through the learning algorithm and back to its training data, and estimates how infinitesimal upweighting or perturbation of a single training instance would affect the model parameters and the loss on a given test point. They also developed a simple and efficient implementation requiring only oracle access to gradients and Hessian-vector products to adapt the method to modern machine learning settings such as high-dimensional deep neural networks. They showed that influence functions can be used for explaining model behavior, detecting incorrectly labeled or harmful training samples, identifying domain mismatch, and even constructing adversarial training-set attacks.

Our work is based on these methods that detect training data with large impact on the inference. However, prior studies do not provide a thorough discussion on the relationship between training data quality and calculated influence scores.

Pruthi et al. proposed TracIn [1], a technique designed to estimate how individual training instances contribute to a model's predictions. Rather than relying on a single final model, TracIn leverages snapshots—or checkpoints—saved throughout the training process. By examining how the loss gradients of a training instance and a test instance align at each checkpoint, the method assigns an influence score that reflects the extent to which the training instance affects the final prediction.

During training with stochastic optimization methods such as stochastic gradient descent, model parameters evolve over many update steps. These updates lead to incremental changes in both loss values and prediction outputs. Although actual test data are not available during training, Pruthi et al. argued that checkpoint models can serve as a practical surrogate for tracing these effects [9].

Because modern training pipelines update parameters using batches of data, TracIn requires isolating the effect of each individual training instance. This is achieved by computing pointwise loss gradients, enabling the method to estimate per-instance influence even under mini-batch training conditions [9].

Formally, the TracIn score is computed by taking the inner product between the loss gradient of the test instance and that of each training instance across all checkpoints, weighting each contribution by the corresponding learning rate, and summing the results:

$$TracIn(z,z') = \sum_{i=i}^{k} \eta_i \nabla l(w_{t_i}, z) \cdot \nabla l(w_{t_i}, z')$$

The authors further demonstrated that training instances with large self-influence—such as mislabeled or otherwise corrupted data—can be effectively detected using TracIn [9].

C. Extraction of Inaccurate Instances in Training Data

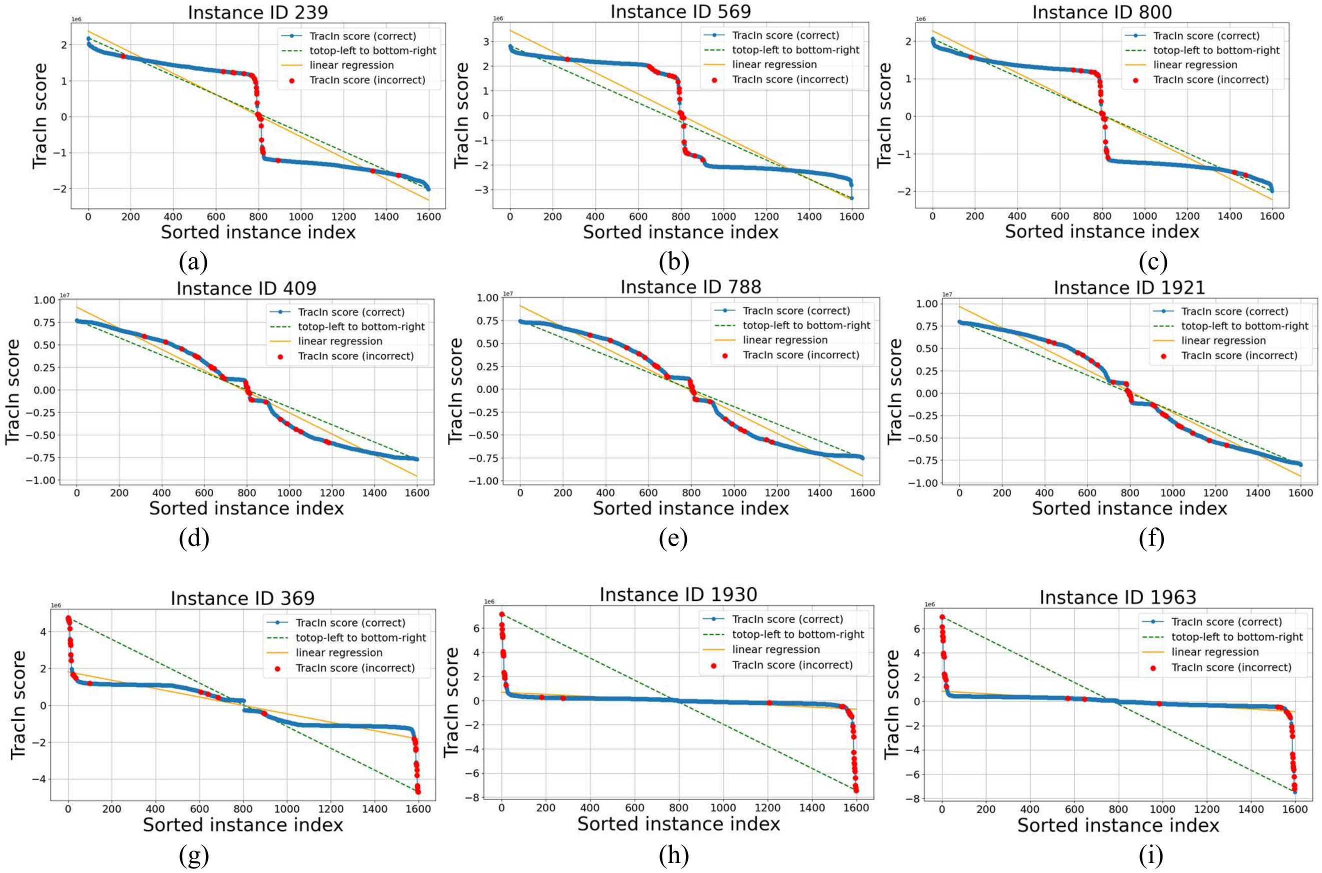
Several studies have examined methods for identifying incorrectly labeled instances in training data by scoring the influence of each instance. Garima et al. reported that incorrectly labeled instances tend to have large positive influence scores for themselves because such instances behave as outliers and tend to reduce the loss with respect to their incorrect labels. Their work is essential and forms a foundation for many subsequent studies. Our work is also based on their findings. However, unlike our study, they did not thoroughly investigate the relationship between influence scores and the accuracy of individual instances.

Hirabayashi et al. [3] proposed a method for improving the accuracy of deep learning models by leveraging decision interpretability to identify unsuitable training data. Their method first splits the overall training dataset into three subsets: the training data within the training data, the validation data within the training data, and the testing data within the training data. It then trains a model using the training data within the training data, validates it on the validation data within the training data, and tests it on the testing data within the training data. If any instance in the testing data within the training data is misclassified, the method selects a misclassified instance and calculates the influence score of each training instance on that specific misclassification using TracIn [1]. A training instance with a strong influence on a misclassification is regarded as an unsuitable instance, and is excluded from the training set. They evaluated the method on a news corpus and a tweet dataset, demonstrating that excluding unsuitable instances improved classification accuracy. However, they did not thoroughly discuss criteria for selecting the instances to be excluded.

III. Relationship Between Training Instance Quality and Influence Scores

In this section, we investigate the relationship between the quality of individual instances and their influence scores. To explore this relationship, we intentionally changed the labels of some randomly selected instances in the training dataset from their original correct labels to incorrect ones. These modified instances are treated as unsuitable (i.e., inaccurate) training instances. The proportion of label-flipped data was 2.5%; specifically, 40 out of 1,600 training instances were assigned incorrect labels.

We used the Rakuten dataset [10], which consists of review texts for various products. For our experiments, we split the dataset into 80% training, 10% validation, and 10% testing, and performed binary sentiment classification. For TracIn score computation, we used 10 checkpoints, which were saved at the end of each training epoch. The learning rate was set to 1e-5, and no learning-rate scheduler was applied.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 1. Training instances and TracIn scores of nine inaccurately infered testing instances

Fig. 1 shows the distribution of TracIn scores for all training instances with respect to nine misclassified test instances. The horizontal axes represent the indices of training instances sorted by their TracIn scores, and the vertical axes represent the TracIn scores themselves. Red plots indicate incorrectly labeled instances, and blue plots indicate correctly labeled ones.

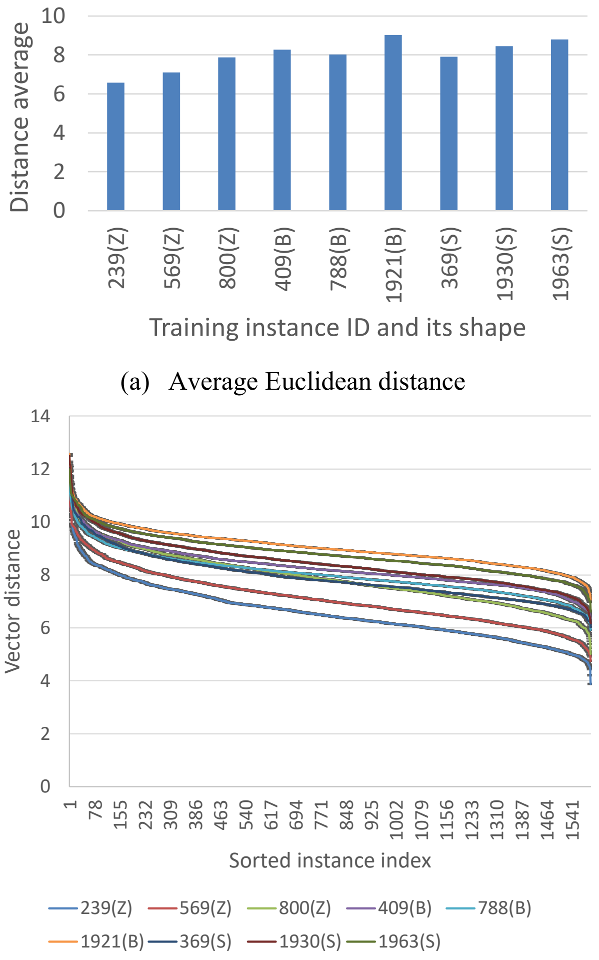
In the S-shaped cases, such as (a), (b), and (c), incorrectly labeled instances appear near both ends of the distribution, i.e., they receive large positive or large negative scores. In the Z-shaped cases, such as (g), (h), and (i), incorrectly labeled instances appear mainly near the center, i.e., they receive small absolute scores. In the backslash-shaped cases, such as (d), (e), and (f), incorrectly labeled instances are located around the center but are more widely spread than in the Z-shaped cases. This case can be considered intermediate between S-shaped and Z-shaped. The differences among S-, Z-, and backslash-shaped cases were consistent across all nine misclassified test instances, indicating that these patterns are not instance-specific noise but reflect underlying structural properties.

To quantify the distribution shape, we define an index of *S-shapedness*. Specifically, we compute the difference between slopes of (i) the linear regression line fitted to the sorted TracIn scores, yellow line in the figure, and (ii) the line connecting the top-left and bottom-right points of the plot, the green line. A larger difference corresponds to a more S-shaped distribution, and a smaller difference corresponds to a more Z-shaped or backslash-shaped distribution.

Fig. 3 shows the relationship between this S-shapedness measure and the ratio of incorrectly labeled instances located at the distribution ends. The ratio is defined by converting each incorrectly labeled instance's ranked position into a percentage distance from the center (0% at the center, 100% at either end) and averaging these values. Strongly S-shaped cases tend to have incorrectly labeled instances near the ends, whereas strongly Z-shaped cases tend to have incorrectly labeled instances near the center. This supports the qualitative trends described above.

Next, we investigate the relationship between instance characteristics and distribution shapes. In S-shaped cases, most training instances have small absolute TracIn scores. We hypothesize that this occurs when the test instance is relatively unique and only a small number of training instances are highly similar to it. In contrast, in Z-shaped cases, many training instances have large absolute scores. We hypothesize that this corresponds to test instances that have many similar training instances. Backslash-shaped cases exhibit intermediate characteristics.

Hypothesis: The shape of the TracIn score distribution is influenced by the degree of similarity between the test instance and the training instances. Higher similarity produces more Z-



(a) Average Euclidean distance

(b) Euclidean distance of each instance

Fig. 2. Disrtances of the testing instance and every training instances

shaped distributions, while lower similarity produces more S-shaped distributions.

To verify this hypothesis, we measured similarity between each test instance and all training instances using BERT embeddings. We used a BERT pre-trained model [11] fine-tuned for sentiment analysis using the Rakuten Ichiba dataset [10] with a learning rate of 1e-5 and a batch size of 32. Model checkpoints for TracIn were saved at regular intervals during training. The embedding vector of each instance was obtained from the final hidden-state vector of the [CLS] token (768 dimensions), and Euclidean distance between vectors was used as the similarity measure. Fig. 2(a) shows the average Euclidean distance, and Fig. 2(b) shows distances sorted by training-instance rank. Each label represents the instance ID that was misclassified by the model and its distribution shape. For example, "239 (Z)" indicates that test instance ID 239 produced a Z-shaped distribution. As shown, S-shaped cases tend to have larger distances, Z-shaped cases have smaller distances, and backslash-shaped cases exhibit intermediate values.

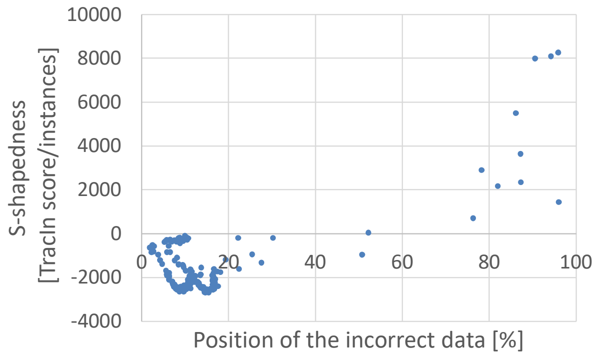


Fig. 3. Degree of S-shapeness and position of the incorrect data

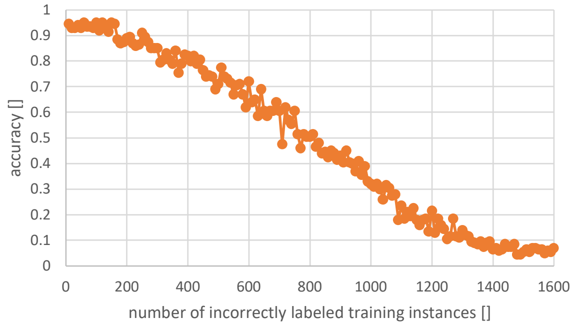


Fig. 4. Num. of incorrectly labeled data and accuracy

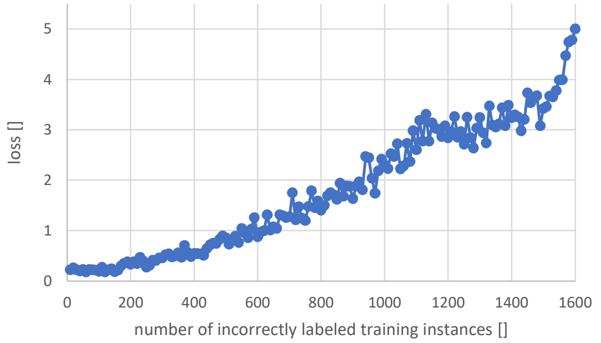


Fig. 5. Num. of incorrectly labeled data and loss

IV. Number of Unsuitable Training Instances and Inference Accuracy

In this section, we study the relationship between number of unsuitable instances in training data and the accuracy of a model trained with this training data.

As in Section III, we reversed the labels of some instances in the training data to investigate the impact of incorrectly labeled instances on inference accuracy. We used the Rakuten dataset [10]. We used 80% of the dataset for training, 10% for validation, and 10% for testing.

Fig. 4 illustrates the relationship between the proportion of incorrect labels and inference accuracy. The results show a trend of decreasing accuracy as the ratio of incorrect instances increases. Based on these results, excluding unsuitable instances from a training dataset is likely to improve performance when the number of such instances is sufficiently large. Focusing on cases with small ratios, we see that accuracy does not decrease significantly. Thus, a small number of incorrect instances have only a minimal negative impact on performance. Therefore, excluding only a small number of incorrect instances does not greatly improve performance, and a substantially larger number of unsuitable instances must be excluded to achieve noticeable improvement.

Fig. 5 shows the relationship between the proportion of incorrectly labeled data and the training loss. Fig. 5 shows a trend similar to that in Fig. 4. Namely, the loss is affected only slightly by the increase in incorrectly labeled data when the number of incorrect instances is small. On the other hand, if the number of incorrect instances is not small, the loss increases largely as the number of incorrect data increases.

V. Discussion

A. *Identification of Unsuitable Training Instances*

We discuss how model performance can be improved by excluding unsuitable training instances. The results in Section IV indicate that excluding only a small number of instances has little impact on model accuracy. Therefore, excluding a larger number of instances may be necessary to achieve meaningful improvement. If the accuracy on the "testing data within the training dataset" (as defined in Section II.C) is low, it suggests that the quality of the training data is poor and that many instances may be incorrectly labeled. In such cases, excluding a substantial number of unsuitable instances may improve the model's accuracy. Even if exclusion does not immediately affect accuracy, removing incorrectly labeled instances can still be beneficial. Such exclusion may have a positive long-term impact on inference, especially when additional training data are incorporated into the dataset in the future.

B. *Additional Perspectives on Influence-Score Distributions*

Our findings imply that the shape of the influence-score distribution can serve as an indicator of underlying characteristics of the training dataset or a specific test instance. For example, S-shaped distributions indicate that only a small subset of training instances substantially interacts with the test instance, suggesting that the test instance is relatively unique or potentially problematic. Evaluating whether such test instances themselves are suitable may therefore be valuable. If many test instances exhibit S-shaped distributions, the dataset may be sparse or highly diverse, indicating a need for targeted data augmentation or additional data collection.

In contrast, if many test instances exhibit Z-shaped distributions, the model relies on a more homogeneous set of neighbors. This provides a new perspective on influence estimation beyond identifying mislabeled instances: the global geometry of influence scores reflects dataset density, redundancy, and locality. Such patterns may reveal highly dense regions where redundancy is high, in which active data pruning may be beneficial.

C. *Implications for Model Robustness and Generalization*

The results in Section IV indicate that the effect of unsuitable training data on inference accuracy is nonlinear. A small number of mislabeled instances does not substantially harm accuracy, but once the number exceeds a certain threshold, the negative impact increases rapidly. This behavior is consistent with findings in robust statistics and adversarial training, where models exhibit phase-transition-like sensitivity to data corruption. Understanding how the shapes of influence-score distributions relate to this threshold could lead to new theoretical insights into the robustness and generalization properties of deep neural networks.

D. *Further Improvement Beyond Exclusion*

While this study focuses on excluding unsuitable training instances, additional responses become possible once such instances are identified. For example, when an instance is highly likely to have an incorrect label, relabeling rather than excluding may be a more effective approach. When the confidence is lower, adjusting the instance's weight based on its estimated influence may be appropriate. Because S-shapedness provides a quantitative indicator of distribution structure, this index may help determine which corrective action is most suitable for each instance. By integrating influence estimation into the training process itself, it may be possible to achieve more robust learning under noisy or heterogeneous datasets.

E. *Limitations*

This study has several limitations. First, we intentionally introduced synthetic label noise, which may not fully reflect naturally occurring annotation errors. Second, all experiments were conducted on a single text dataset, and further validation across other domains and modalities is needed. Third, our analysis focused solely on TracIn, and other influence-estimation methods may exhibit different distribution behaviors.

VI. Conclusion

In this paper, we investigated the relationship between the quality of training instances and their influence scores, focusing on TracIn as a representative method for estimating per-instance influence. By intentionally generating incorrectly labeled instances and analyzing their TracIn score distributions, we found that the distributions of incorrectly labeled data follow three characteristic distribution patterns. Specifically, we identified three characteristic distribution patterns, which are S-shaped, Z-shaped, and backslash-shaped. In the case of S-shaped distribution, the incorrectly labeled data tend to be located near both ends of distribution, i.e., with large absolute TracIn scores. In the Z-shaped cases, the incorrectly labeled data tend to appear near the center, with small absolute TracIn scores. Backslash-shaped cases exhibit intermediate behavior between the S-shaped and Z-shaped cases. In addition, we showed that these patterns correlate with the degree of similarity between test and training instances. Our findings suggest that the shape of the influence score distribution reflects the structural relationship between data points in the embedding space. We also examined how the number of unsuitable instances affects model performance. The results indicate that a small number of incorrect instances has only a limited negative impact on accuracy, implying that exclusion of only a few such instances does not substantially improve model performance. However, when many incorrectly labeled instances exist in the training data, their exclusion becomes essential for improving the model. We expect that this study provides empirical insights into how influence estimation can reveal the behavior of unsuitable training instances and guide effective data cleaning strategies. These findings contribute to a deeper understanding of training-data-based interpretability and provide a foundation for future work on automated identification and exclusion of low-quality data.

For future work, we plan to improve methods for enhancing model performance by excluding unsuitable training data, building on the findings of this study. We also plan to examine whether the observed trends hold across different datasets and other influence-estimation methods such as influence functions.

ACKNOWLEDGMENT

In this paper, we used "Rakuten Dataset" (https://rit.rakuten.com/data_release/) provided by Rakuten Group, Inc. via IDR Dataset Service of National Institute of Informatics.

This work was supported by JSPS KAKENHI Grant Number 25K15100.

REFERENCES

[1] Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. 2020. Estimating training data influence by tracing gradient descent. In Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS '20). Curran Associates Inc., Red Hook, NY, USA, Article 1672, 19920–19930.

[2] Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In Proceedings of the 34th International Conference on Machine Learning - Volume 70 (ICML'17). JMLR.org, 1885–1894.

[3] S. Yamaguchi, F. Hirabayashi and A. Tamekuri, "Improvement of Deep Learning Models by Excluding Inappropriate Data Based on Interpretability," 2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC), Osaka, Japan, 2024, pp. 291-296, doi: 10.1109/COMPSAC61105.2024.00048.

[4] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). ACM, New York, NY, USA, 1135-1144. DOI: https://doi.org/10.1145/2939672.2939778

[5] Montavon, G., Samek,W. andM¨uller, K.-R.: Methods for Interpreting and Understanding Deep Neural Networks, Digital Signal Processing, Vol.73, pp.1–15 (Feb. 2018).

[6] Karen Simonyan, Andrea Vedaldi and Andrew Zisserman, Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, Workshop on ICLR, 2014.

[7] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas and Martin Wattenberg, "SmoothGrad: removing noise by adding noise," Workshop on Visualization for Deep Learning in ICML, 2017

[8] S. Shirataki and S. Yamaguchi, "A study on interpretability of decision of machine learning," 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 2017, pp. 4830-4831, doi: 10.1109/BigData.2017.8258557.

[9] Frederick Liu and Garima Pruthi, "TracIn — A Simple Method to Estimate Training Data Influence," https://research.google/blog/tracin-a-simple-method-to-estimate-training-data-influence/ (2025/12/12 access)

[10] Rakuten Group, Inc. (2025): Rakuten Ichiba data. Informatics Research Data Repository, National Institute of Informatics. (dataset). https://doi.org/10.32130/idr.2.1

[11] Tohoku NLP Group, "BERT models for Japanese text," https://huggingface.co/tohoku-nlp