Toward Model Selection Through Measuring Dataset Similarity on TensorFlow Hub

SeungYoung Oh[1] Hyunmin Lee[1] JinHyun Han[1 2] Hyunggu Jung[1]

Abstract

For novice developers, it is a challenge to select the most appropriate model without prior knowledge of artificial intelligence (AI) development. The main goal of our system is to provide an automated approach to presenting models through dataset similarity. We present a system that allows novel developers to select the best model among existing models from a TensorFlow Hub (TF Hub) online community. Our strategy was to use the similarity of two datasets as a measure to determine the best model. Through a systematic review, we identified several limitations, each of which corresponds to a function to be implemented in our proposed system. We then created a model selection system that enables novice developers to select the most appropriate ML model without prior knowledge of AI by implementing the identified functions. The analysis of this study reveals that our proposed system performed better by successfully addressing three out of six identified limitations.

1. Introduction

As the development of machine learning (ML) models involves increasing time and resources, computational requirements, such as computational parameters, continue to grow[1]. Consequently, the computational resource of ML development is continuously increased resources (Wu et al., 2022). As the number of parameters increases to minimize the loss, the training time also escalates (Kaplan et al., 2020). The increasing cost of computing resources and time-consuming becomes a critical problem.

One of the existing methods to reduce the cost of computational resources and time is to select a suitable model to be trained on the dataset by the ML model selection system. However, the existing ML model selection systems have two limitations. First, previous ML model selection systems focus on selecting models from a predefined set without addressing how to determine the appropriate set of models. For example, researchers arbitrarily select relevant tasks to obtain state-of-the-art (SOTA) models for performance evaluation (Abdallah et al., 2022). Second, these methods tend to apply only to specific inputs or model architectures (Yoon et al., 2021; Xiang & Gong, 2008; Yang et al., 2023; Abdallah et al., 2022). While researchers created the ML model selection system, No studies proposed an ML model selection system that automatically creates an ML model set from existing online AI communities.

To reduce this gap, we created the ML model selection system that automatically creates an ML model set from TensorFlow Hub (TF Hub)[2]. Inspired by Yoo et al., we used TF Hub which is an online AI community containing models and datasets to enable novice developers to select models without prior knowledge of AI (Yoo et al., 2022). To achieve our goal, we conducted the following process (see Figure 1): (1) find a list of functions that address prior limitations of the existing model selection system, (2) determine and implement functions of our system, and (3) evaluate the performance of our proposed system. By conducting the process above, we created the ML model selection system and evaluate the performance of our proposed system by identifying the number of addressed limitations we found in a systematic review. Our system addressed three out of six identified limitations. Our study makes the following contributions to the AI and human-computer interaction (HCI) communities: (1) We propose a method that utilizes existing models and datasets on TF Hub; (2) We propose a model selection system that enables novice developers to select ML models without prior knowledge.

2. Related Work

This section summarizes prior studies that propose ML model selection systems.

[1]University of Seoul, Republic of Korea [2]UST21, Republic of Korea. Correspondence to: Hyunggu Jung <hjung@uos.ac.kr>.

AI & HCI Workshop at the 40^{th} International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA. 2023.

[1] https://openai.com/research/ai-and-compute

[2] https://tfhub.dev/s

2.1. ML Model Selection in Several Dataset Domains

Researchers have developed a variety of systems that select ML models with multiple dataset domains with two types of data: tabular and non-tabular format input data. While some studies select models with tabular format input data such as time series (Abdallah et al., 2022) and non-time series that each row of the dataset is independent with other rows (Pevec & Kononenko, 2012; Ghorai et al., 2010; Vosseler, 2022), The other studies proposed the ML model selection system with non-tabular format input data such as image (Kutukcu et al., 2022; Liu et al., 2021), text (Le & Lo, 2018), audio (Yoon et al., 2021), and video domain (Xiang & Gong, 2008).

2.2. ML Model Selection Through Dataset Similarity

Likewise, researchers have developed systems for selecting ML models based on by similarity of data sets. While one study specified where the model was taken from WEKA[3] (Makhtar et al., 2011), three studies did not specify how they collected the models in the model set for performance evaluation (Noguchi et al., 2021; Yang et al., 2023; Liu et al., 2002).

2.3. Limitations of Prior Studies

We found two limitations of prior studies that propose a system to select the ML model. First, the lack of objective criteria for including models in the model set is observed in all related studies. While they exclusively used models from WEKA, the researchers arbitrarily selected the models to be used (Makhtar et al., 2011). Second, we found that some systems were only applied to a specific type of dataset (e.g., time-series (Yang et al., 2023; Abdallah et al., 2022)) or model architecture (e.g., teacher-student architecture (Yoon et al., 2021)). According to these limitations, we found that no studies have proposed an ML model selection system that automatically creates an ML model set from the existing AI community (e.g., TF Hub, Hugging Face [4]). Therefore, this study aims to propose an ML model selection system that addresses these limitations.

3. Preliminary Study

We created a list of functions that address the limitations of the model selection systems proposed by prior studies. First, we conducted a systematic review to find the studies that proposed ML model selection systems. Second, we identified the limitations of the system proposed by prior studies and created a list of functions that address the limitations.

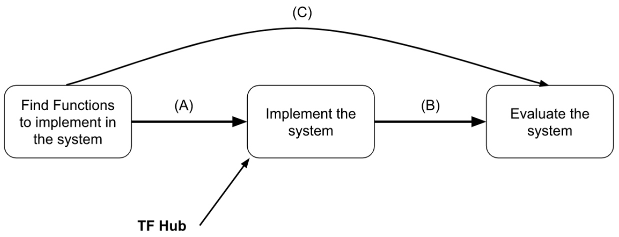

Figure 1. This diagram illustrates the study procedure. First, we created a potential functions list (A) to implement in the system and a limitations list (C) through a systematic review. Second, we implemented system (B) using the data on models and datasets from TF Hub. Third, we evaluated the performance of the system by identifying the number of limitations addressed in the limitations list (C).

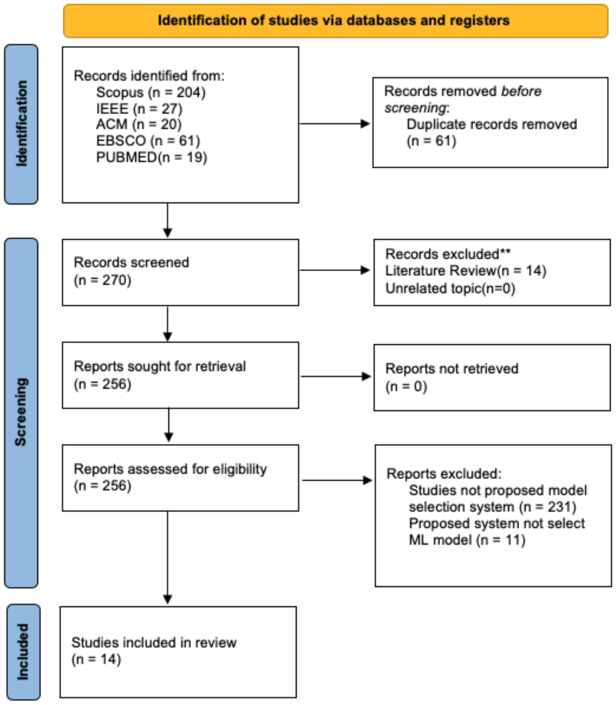
Identification of studies via databases and registers
Identification
Records identified from:
Scopus (n = 204)
IEEE (n = 27)
ACM (n = 20)
EBSCO (n = 61)
PUBMED(n = 19)
Records removed before screening:
Duplicate records removed
(n = 61)
Screening
Records screened
(n = 270)
Records excluded**
Literature Review(n = 14)
Unrelated topic(n=0)
Reports sought for retrieval
(n = 256)
Reports not retrieved
(n = 0)
Reports assessed for eligibility
(n = 256)
Reports excluded:
Studies not proposed model selection system (n = 231)
Proposed system not select ML model (n = 11)
Included
Studies included in review
(n = 14)

Figure 2. Systematic review strategy based on PRISMA (2020) Guidelines (Page et al., 2021).

3.1. Data Collection Through Systematic Review

We conducted a systematic review. The goal of the systematic review was to identify prior studies that proposed an ML model selection system. To achieve the goal, we took the following steps: (1) conducting the literature search process, (2) determining eligibility criteria, and (3) data collection and synthesis. As a result, we obtained a list of eligible papers that proposed ML model selection systems.

To identify eligible studies, we used the following five databases ACM, EBSCO, IEEE, PubMed, and Scopus. The

[3] https://weka.sourceforge.io/doc.dev/overview-summary.html

[4] https://huggingface.co/

same search string was utilized for each database and applied to at least one of the following items: title, abstract, or keywords (see Table 1). The initial search yielded 331 articles that matched the search keywords. In the second phase, we refined the search results by excluding 61 duplicate articles. In the third phase, we performed a title and abstract screening, eliminating 14 articles that were either literature reviews or unrelated to the topic. As a result, we identified a total of 256 relevant articles.

We determined the eligibility criteria to identify existing model selection systems. The eligibility criteria for filtering the papers were as follows: (1) The study proposes a model selection system or tool; (2) The study involves the selection of ML models. In this study, we define an ML model as a model that meets with the following criteria[5][6][7]: (1) The model explains how something works and calculates what might happen; (2) The model is a file or a program; (3) The model recognizes or finds a pattern in data; (4) The model is completed through training; (5) The model is being trained to improve its accuracy. Using the eligibility criteria, we determine ML models. We thoroughly reviewed 256 screened articles and filtered them based on our eligibility criteria. For collecting data and synthesis, we developed a data-charting form to organize and identify limitations in the studies (see Figure 2). After applying the eligibility criteria, only fourteen studies remained, which contributed to identifying limitations from previous studies in our systematic review.

3.2. Identification of Limitations and Functions for Model Selection

We identified limitations and created a list of potential functions of our proposed system through the following process. First, we found the functions of the system proposed by prior studies. Second, we found the limitations of the existing functions (see Table 2). Third, we created a list of potential functions of our proposed system through the following process (see Table 3).

4. Implementation

The goal of this section is to describe a method of how we implemented the system that allows novice developers to select ML models without prior knowledge of AI. To achieve this goal, we conducted the following process: (1) determining the functions we implement in our system, (2) implementing the determined functions, and (3) creating the system containing the functions we implemented. Following these steps, we created the ML model selection system, which enables novice developers to select an ML model without prior knowledge of AI.

4.1. Determining the Implemented Functions in the System

From the list of functions to address limitations in subsection 3.2, we determined the functions that meet the following eligibility criterion: The function to help developers without prior knowledge of AI. By determining the functions by the eligibility criterion, we created the system by implementing three functions (see Table 3).

4.2. Implementing the Determined Functions in the System

We implemented the identified functions as follows. First, we crawled TF Hub to collect data containing the model and the dataset that the model had previously learned. Then, we measured the similarity between the datasets using the crawled data and the implemented functions. Using the Python (Van Rossum & Drake, 2009) libraries including Selenium[8] and BeautifulSoup[9], we crawled TF Hub to collect data containing the model and the dataset that the model previously learned. We limited our crawling to the text domain out of four dataset domains(text, audio, video, and image) available in TF Hub, because the text domain model rapidly increases training time (Wu et al., 2022). In the crawled data, we removed duplicate models. If the model_name, architecture, description, publisher, and dataset were all the same, we considered them duplicates. As a result, we created a spreadsheet with the publisher, model_name, description, architecture, dataset, and link as columns.

We implemented each function that measures the similarity between two datasets. We determined the following seven datasets that meet the following eligibility criteria: (1) The dataset has a predefined size; (2) The dataset is obtainable; (3) The dataset has labeled ground truth. To measure the similarity between the two datasets, we followed two steps. First, we used the Sent2Vec[10] model to obtain embeddings of the combined dataset representation (see Table 4). Second, we used the Dataset2Vec (Jomaa et al., 2021) model to calculate the embedding value of each dataset. We sample 1000 instances of dataset randomly, due to memory constraints. By following the two steps above, we measure the similarity between each pair of datasets and obtain their

[5] https://dictionary.cambridge.org/dictionary/english/model

[6] https://www.tensorflow.org/guide/intro_to_modules#defining_models_and_layers_in_tensorflow

[7] https://learn.microsoft.com/en-us/windows/ai/windows-ml/what-is-a-machine-learning-model

[8] https://www.selenium.dev/

[9] https://beautiful-soup-4.readthedocs.io/en/latest/

[10] https://github.com/epfml/sent2vec

Table 1. Search string. Abstract abbreviated as either “ABS” or “AB”. Keywords abbreviated as “KEY”.

Database	Search String
ACM Digital Library	[[Abstract: model] OR [Abstract: architecture] OR [Abstract: framework]] AND [Abstract: ”model selection”] AND [Abstract: similarity] AND [[Abstract: tool] OR [Abstract: app] OR [Abstract: platform] OR [Abstract: application] OR [Abstract: system]]
EBSCO	AB (model OR architecture OR framework) AND AB (tool OR app OR platform OR application OR system) AND AB ”model selection” AND AB similarity
IEEE Xplore	(”Abstract”: model OR ”Abstract”: architecture OR ”Abstract”: framework) AND (”Abstract”: tool OR ”Abstract”: app OR ”Abstract”: platform OR ”Abstract”: application OR ”Abstract”: system) AND (”Abstract”: ”model selection”) AND (”Abstract”: similarity)
PubMed	(model[Title/Abstract] OR architecture[Title/Abstract] OR framework [Title/Abstract]) AND (”model selection” [Title/Abstract]) AND similarity[Title/Abstract] AND (tool[Title/Abstract] OR app[Title/Abstract] OR platform[Title/Abstract] OR application[Title/Abstract] OR system [Title/Abstract])
Scopus	TITLE-ABS-KEY ((model OR architecture OR framework) AND (”model selection”) AND similarity AND (tool OR app OR platform OR application OR system))

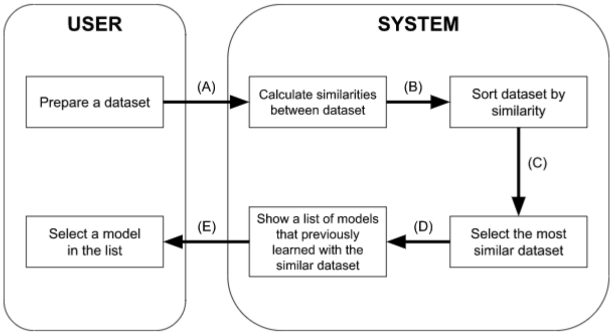
USER
SYSTEM
Prepare a dataset
(A)
Calculate similarities between dataset
(B)
Sort dataset by similarity
(C)
Select the most similar dataset
(D)
Show a list of models that previously learned with the similar dataset
(E)
Select a model in the list

Figure 3. The flowchart of the system. The user input the dataset (A) that user wants to train. The system calculates the similarity between datasets and creates the list (B) of similarity. After sorting the list (C) by the value of similarity, the most similar dataset name (D) is determined. Finally, the user selects the model in the list (E) containing models which previously learned with (A).

corresponding embedding values (see Figure 4). The similarity value between datasets was calculated by applying the embedding values of the datasets to the cosine similarity formula. A and B are the data sets to measure the similarity. If A and B are identical, the similarity value is 0.

$$cosine\ similarity = \frac{A \cdot B}{\|A\|\|B\|}$$

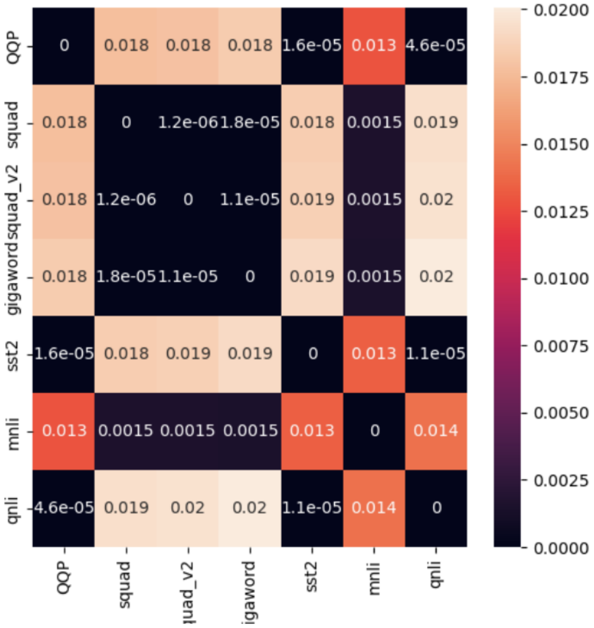

Figure 4. The heatmap of the dataset similarity.

4.3. Proposed System

The interaction between the user and our proposed system operates as follows (see Figure 3): (1) The user prepares the dataset they want to train, (2) The system measures the embedding values of the dataset and computes the similarity

Table 2. Limitations of prior studies.

Limitations of Prior Studies	Description of Limitations	Studies with Limitations
No criterion of the model set	Because the model selection system doesn't address how to define a model selection set, the developer adds models to the model set one by one	(Yang et al., 2023; Liu et al., 2002; Makhtar et al., 2011; Noguchi et al., 2021; Pevec & Kononenko, 2012; He & Shaposhnik, 2023; Kutukcu et al., 2022; Ghorai et al., 2010; Le & Lo, 2018; Yoon et al., 2021; Xiang & Gong, 2008; Vosseler, 2022; Abdallah et al., 2022; Liu et al., 2021)
Dependent on model structure	Because the model selection system is dependent on model structure, the developer only chooses the typical structure of model	(Yoon et al., 2021; Xiang & Gong, 2008)
Dependent on dataset type	Because the model selection system is dependent on dataset type, the developer only chooses the typical model that is trained on the specific type of dataset	(Yang et al., 2023; Abdallah et al., 2022)
No numerical indication of the best model	Because the model selection system only gives you the optimal number of models, the developer is unable to recognize the best model	(Liu et al., 2002; Xiang & Gong, 2008; Liu et al., 2021)
Need to maintain model pool	Because the model selection system needs to maintain the model pool, the developer is unable to recognize the best model	(Noguchi et al., 2021)
No criterion of model selection	Because the model selection system determines the best model heuristically, the developer is unable to recognize the criteria	(Le & Lo, 2018)

between datasets, (3) The datasets are sorted based on their similarity, (4) The system selects the dataset with the highest similarity value, (5) A list of models trained on the most similar datasets is generated, and (6) The system presents the generated list to the user.

The usage scenario of our system is as follows: Grek is a researcher in the field of bioinformatics. Grek has no previous knowledge of AI. Grek needs to train the ML model on a dataset for his research. Grek inputs the dataset *qnli* to our proposed system (Wang et al., 2018). As a result of measuring the similarity of the datasets, *sst2*[11] is the most similar dataset to *qnli* that the user inputs to the system. The system returns a list of models previously trained with *sst2*. Grek selects a model called *experts/bert/wiki_books/sst2*[12] that was previously trained with *sst2*.

5. Evaluation

We evaluated the performance of our proposed system by identifying the number of addressed limitations found in subsection 3.2. Ultimately, our proposed system addressed three out of the six identified limitations. To evaluate how many limitations our system has addressed, we created a spreadsheet with the limitations identified in subsection 3.2 as evaluation items (see Table 5). We have indicated in the corresponding rows whether our proposed system addresses each limitation and described how it has been addressed in the case of a "YES" answer.

6. Discussion

In this study, we found several key findings. First, the value of similarity between two samples from two different datasets was low, but the value of similarity between two samples from the same dataset was high. Second, no studies provided objective criteria for selecting models in

[11] https://nlp.stanford.edu/sentiment/treebank.html

[12] https://tfhub.dev/google/experts/bert/wiki_books/sst2/2

Table 3. Function list to address the limitations.

Limitations of Prior Studies	Functions to Address Limitations	Implemented
No criterion of the model set	Function to pull in relevant models listed in communities like TF Hub	YES
Dependent on model structure	Function to recommend models based on the similarity between datasets that have been trained on	YES
Dependent on dataset type	Embedding functions to measure similarity across any type of dataset	YES
No numerical indication of the best model	Numerical indication of the best model	NO
Need to maintain model pool	Numerical indication of the best model	NO
No criterion of model selection	Determine models based on specific criteria, not heuristics	NO

models set for performance evaluation. Third, the models that were trained on undefined datasets, such as Twitter[13] and Wikipedia[14]. Fourth, the models recommended by our system were unable to perform the desired task.

6.1. Factors Affecting Dataset Similarity

In this study, we observed that the embedding similarity values were low between different datasets, but high when sampling embeddings from the same dataset. This finding led us to investigate what factors significantly influence the measurement of similarity using embedding scores. Previous studies on embedding vector spaces have shown that the architecture used to measure embedding values has a significant impact on the performance of a given task (Mikolov et al., 2013). In addition, Mikolov et al. reported that higher-dimensional embeddings better capture the semantic relationships between words in meaning-related tasks (Mikolov et al., 2013). We conclude that the dimensionality of the embeddings and the model architecture may have a critical role in determining the similarity measured by embeddings.

6.2. Criteria of Model Set for Evaluating Performance

In reviewing the relevant literature, we found that no papers provided objective criteria for selecting models to be included in the model set for performance evaluation. For example, Makhtar et al. obtained models from the Weka for performance evaluation (Makhtar et al., 2011). In some cases, researchers arbitrarily selected SOTA models from tasks similar to the one being evaluated (Abdallah et al., 2022). Thus, we conclude that researchers tended to rely on their personal experience to determine the model set for evaluating AI model performance.

6.3. Undefined Dataset

We obtained model and dataset information from TF Hub, which included models trained on datasets such as Wikipedia and Twitter, where the sizes of the datasets obtained were not specified. This led us to question whether the models trained on the unspecified size datasets had intentionally limited the dataset sizes during training. We investigated whether there are any considerations when using unspecified size datasets compared to fixed size datasets. In recent studies using Twitter as a dataset, researchers randomly selected dates and used them as the dataset, taking into account the continuously generated data (Agarwal et al., 2011). Furthermore, the mention of an unspecified size could also mean that the dataset was not preprocessed for training. Therefore, we assumed that the researchers in these studies applied their own criteria to truncate and refine the dataset based on an algorithm, excluding data they deemed useless by attaching a “junk” label.

In brief, our results raise important questions about the factors influencing similarity measurements using embeddings, the similarity of the data sets, the criteria for selecting models in model sets, and considerations when using data sets of unspecified size. These aspects need to be further explored and addressed to improve the efficiency and effectiveness of AI model selection and evaluation.

[13] https://twitter.com/

[14] https://www.wikipedia.org/

Table 4. Columns of the dataset and converting the format of the input. The transformation of text to vector is processed using Sent2Vec.

Dataset Name	Columns of Dataset	Input Transform Format
qqp (Wang et al., 2017)	[‘text1’, ‘text2’, ‘label’, ‘idx’, ‘label_text’]	input: (‘text1’+‘text2’)→vector label: ‘label’
squad (Rajpurkar et al., 2016)	[‘id’, ‘title’, ‘context’, ‘question’, ‘answers’]	input: ‘question’ →vector label: ‘answer’ → vector
squad_v2 (Katyayan & Joshi, 2022)	[‘id’, ‘title’, ‘context’, ‘question’, ‘answers’]	input: ‘question’ →vector label: ‘answer’ → vector
gigaword (Napoles et al., 2012)	[‘document’, ‘summary’]	input: ‘document’ → vector label: ‘answer’ → vector
sst2 (Socher et al., 2013)	[‘idx’, ‘sentence’, ‘label’]	input: ‘sentence’ → vector label: ‘label’
mnli (Williams et al., 2018)	[‘text1’, ‘text2’, ‘label’, ‘idx’, ‘label_text’]	input: (‘text1’ + ‘text2’) → vector label: ‘label’
qnli (Wang et al., 2018)	[‘text1’, ‘text2’, ‘label’, ‘idx’, ‘label_text’]	input: (‘text1’ + ‘text2’) → vector label: ‘label’

6.4. Models with Different Tasks

In our study, we found that some of the models recommended by our system were unable to effectively perform the desired task with the given dataset. This raised the question of whether these models are truly unusable, or whether there are alternative approaches to using them effectively. To explore this further, we examined the success of the Transformer architecture, which is widely recognized as the standard for natural language processing (NLP) tasks, in achieving state-of-the-art (SOTA) performance in image recognition on one of the most popular datasets *CIFAR-10* [15] [16] (Dosovitskiy et al., 2021). We have found cases where the Transformer architecture has been successfully applied to image recognition, demonstrating its versatility across domains. Although not directly applicable to the task at hand, these models could provide valuable insights and alternative perspectives. By considering the successes of the Transformer architecture in image recognition, we could gain similar insights and potentially improve the development of models recommended by our system. Overall, our finding of models that may not be directly applicable to the desired task encourages exploration of the successes of the Transformer architecture in image recognition, providing valuable insights and paving the way for improved model recommendation systems in image-related tasks.

7. Limitations and Future Work

We created and evaluated the model selection system that enables novice developers to select ML models without prior knowledge of AI. However, this study has several limitations. First, while we evaluate the performance of our proposed system by identifying the number of limitations, we did not evaluate the performance of our proposed system that the models selected by our system perform well on the datasets that users intend to train. Second, while we measure the similarity between datasets, we did not evaluate the performance of the dataset similarity measured by our system accurately reflects the similarity between actual datasets. Third, although, we output the embeddings for each dataset, we converted the 1000 sampled datasets into a vector representation due to memory constraints. Fourth, we used five different electronic databases for this study. However, by using a search strategy across limited databases, we have the risk of having missed relevant studies.

Future work remains to create a tool that enables ML model developers to interact with the system through a visual interface. The visualized tool would enable users to better understand the recommendations and make informed decisions regarding model selection. We may evaluate the performance of models in the list that the system shows to

[15] https://paperswithcode.com/task/image-classification

[16] https://www.cs.toronto.edu/~kriz/cifar.html

Table 5. The result of the evaluation.

Limitations of Prior Studies	Addressed	Description of the Functions Implemented in Proposed System
No criterion of the model set	YES	Our system automatically collects models from TF Hub
Dependent on model structure	YES	Our system is not dependent on model structure as it recommends models based on the similarity of the dataset and the records the model has been trained on
Dependent on dataset type	YES	Our system take any type of data if it turns it into a tabular format of numbers.
No numerical indication of the best model	NO	
Need to maintain model pool	NO	
No criterion of model selection	NO	

the user. Performance evaluation allows us to verify that our approach to model recommendation via the similarity of datasets actually works well. Also, we may improve dataset embedding models to represent the similarity of the dataset well. Improved embedding leads the proposed system to suggest a more valuable model list. By addressing these limitations and conducting the suggested future work, we enhance the practicality and reliability of our system, making it a more valuable tool for researchers and practitioners in selecting and training AI models.

8. Conclusion

The ultimate goal of this study was to reduce the time and resource requirements involved in selecting models for AI training. The key contribution of this paper is proposing an ML model selection system that enables novice developers to identify ML models without prior knowledge of AI. To create such a system, we identified the limitations of prior studies and implemented functions to address the identified limitations. Future work remains to improve the usability and feasibility of the system by addressing the needs of stakeholders (i.e., novice developers) of the system. We hope that our proposed system would be applicable to the practices of novice developers without prior background knowledge in the AI and HCI communities.

Acknowledgements

We appreciate HCAIL members for their constructive feedback on our initial manuscript.

References

Abdallah, M., Rossi, R., Mahadik, K., Kim, S., Zhao, H., and Bagchi, S. Autoforecast: Automatic time-series forecasting model selection. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 5–14, 2022.

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., and Passonneau, R. J. Sentiment analysis of twitter data. In *Proceedings of the workshop on language in social media (LSM 2011)*, pp. 30–38, 2011.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

Ghorai, S., Mukherjee, A., Sengupta, S., and Dutta, P. K. Cancer classification from gene expression data by nppc ensemble. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(3):659–671, 2010.

He, Z. and Shaposhnik, Y. Visualizing the implicit model selection tradeoff. *Journal of Artificial Intelligence Research*, 76:829–881, 2023.

Jomaa, H. S., Schmidt-Thieme, L., and Grabocka, J. Dataset2vec: Learning dataset meta-features. *Data Mining and Knowledge Discovery*, 35:964–985, 2021.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Katyayan, P. and Joshi, N. Design and development of rule-based open-domain question-answering system on squad v2. 0 dataset. *arXiv preprint arXiv:2204.09659*, 2022.

Kutukcu, B., Baidya, S., Raghunathan, A., and Dey, S. Contention grading and adaptive model selection for machine vision in embedded systems. *ACM Transactions on Embedded Computing Systems (TECS)*, 21(5):1–29, 2022.

Le, T.-D. B. and Lo, D. Deep specification mining. In *Proceedings of the 27th ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA 2018, pp. 106–117, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356992. doi: 10.1145/3213846.3213876. URL https://doi.org/10.1145/3213846.3213876.

Liu, B., Zhao, Y., Jiang, X., Wang, S., and Wei, J. 4d epanechnikov mixture regression in lf image compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(6):3906–3922, 2021.

Liu, X., Gong, Y., Xu, W., and Zhu, S. Document clustering with cluster refinement and model selection capabilities. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 191–198, 2002.

Makhtar, M., Neagu, D. C., and Ridley, M. J. Binary classification models comparison: On the similarity of datasets and confusion matrix for predictive toxicology applications. In *Information Technology in Bio-and Medical Informatics: Second International Conference, ITBAM 2011, Toulouse, France, August 31-September 1, 2011. Proceedings 2*, pp. 108–122. Springer, 2011.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Napoles, C., Gormley, M. R., and Van Durme, B. Annotated gigaword. In *Proceedings of the joint workshop on automatic knowledge base construction and web-scale knowledge extraction (AKBC-WEKEX)*, pp. 95–100, 2012.

Noguchi, H., Isoda, T., and Arai, S. Shared trained models selection and management for transfer reinforcement learning in open iot. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2170–2176. IEEE, 2021.

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., McGuinness, L. A., Stewart, L. A., Thomas, J., Tricco, A. C., Welch, V. A., Whiting, P., and Moher, D. The prisma 2020 statement: an updated guideline for reporting systematic reviews. *BMJ*, 372, 2021. doi: 10.1136/bmj.n71. URL https://www.bmj.com/content/372/bmj.n71.

Pevec, D. and Kononenko, I. Model selection with combining valid and optimal prediction intervals. In *2012 IEEE 12th International Conference on Data Mining Workshops*, pp. 653–658. IEEE, 2012.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Van Rossum, G. and Drake, F. L. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009. ISBN 1441412697.

Vosseler, A. Unsupervised insurance fraud prediction based on anomaly detector ensembles. *Risks*, 10(7):132, 2022.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

Wang, Z., Hamza, W., and Florian, R. Bilateral multi-perspective matching for natural language sentences. *arXiv preprint arXiv:1702.03814*, 2017.

Williams, A., Nangia, N., and Bowman, S. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/N18-1101.

Wu, C.-J., Raghavendra, R., Gupta, U., Acun, B., Ardalani, N., Maeng, K., Chang, G., Aga, F., Huang, J., Bai, C., et al. Sustainable ai: Environmental implications, challenges and opportunities. *Proceedings of Machine Learning and Systems*, 4:795–813, 2022.

Xiang, T. and Gong, S. Video behavior profiling for anomaly detection. *IEEE transactions on pattern analysis and machine intelligence*, 30(5):893–908, 2008.

Yang, Z., Peng, X., Song, J., Duan, R., Jiang, Y., and Liu, S. Short-term wind power prediction based on multi-parameters similarity wind process matching and weighed-voting-based deep learning model selection. *IEEE Transactions on Power Systems*, 2023.

Yoo, T., Chun, M., Bae, Y., Kwon, S., and Jung, H. Exploring the community of model publishers on tensorflow hub. In *Companion Publication of the 2022 Conference on Computer Supported Cooperative Work and Social Computing*, pp. 63–67, 2022.

Yoon, J. W., Lee, H., Kim, H. Y., Cho, W. I., and Kim, N. S. Tutornet: Towards flexible knowledge distillation for end-to-end speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:1626–1638, 2021.