Reusability in Artificial Neural Networks: An Empirical Study

Javad Ghofrani
Faculty of Informatics/Mathematics
HTW University of Applied Sciences
Dresden, Germany
javad.ghofrani@gmail.com

Ehsan Kozegar
Faculty of Engineering (Eastern Guilan)
University of Guilan
Guilan, Iran
kozegar@guilan.ac.ir

Arezoo Bozorgmehr
Uni Klinikum Bonn
Bonn, Germany
arezoo.bozorgmehr16@gmail.com

Mohammad Divband Soorati
University of Lübeck
Lübeck, Germany
divband@iti.uni-luebeck.de

ABSTRACT

Machine learning, especially deep learning has aroused interests of researchers and practitioners for the last few years in development of intelligent systems such as speech, natural language, and image processing. Software solutions based on machine learning techniques attract more attention as alternatives to conventional software systems. In this paper, we investigate how reusability techniques are applied in implementation of artificial neural networks (ANNs). We conducted an empirical study with an online survey among experts with experience in developing solutions with ANNs. We analyze the feedback of more than 100 experts to our survey. The results show existing challenges and some of the applied solutions in an intersection between reusability and ANNs.

CCS CONCEPTS

• **Computing methodologies → Artificial intelligence**; • **Software and its engineering → Reusability**; *Software product lines*; • **General and reference** → *Empirical studies.*

KEYWORDS

Systematic reuse, artificial neural networks, reusability, survey, empirical study

ACM Reference Format:

Javad Ghofrani, Ehsan Kozegar, Arezoo Bozorgmehr, and Mohammad Divband Soorati. 2019. Reusability in Artificial Neural Networks: An Empirical Study. In *23rd International Systems and Software Product Line Conference - Volume B (SPLC '19), September 9–13, 2019, Paris, France.* ACM, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.1145/3307630.3342419

1 INTRODUCTION

Artificial neural networks (ANNs), especially deep neural networks (DNNs), have shown fast raising success in last few years. Contribution of DNNs in various fields such as speech processing, natural language processing, and computer vision was significant. Deep learning has been a powerful technique in the edge of technologies including smart production systems [22], smart cities [9], and self-driving cars. Recently, the world's biggest companies (e.g., Google, Amazon, and Microsoft) started to invest in DNN based solutions. Tensorflow [1] and CNTK [36] are two of many frameworks offered by companies to facilitate and standardize the development of ANN-based software systems.

Big data analytics and Convolutional Neural Networks (CNNs) drew the attention of researchers and practitioners to ANN techniques. Instead of writing complicated algorithms, software systems are developed based on training the neural networks on large datasets [25]. Despite outstanding performance and satisfying functionality in special contexts (e.g., image classification or prediction of future events in industrial systems), ANNs is not able to solve every challenge in the domain of software engineering (SE) [31].

Development of software systems with ANNs is not exploited enough by the community of (SE) [6]. Most of the concepts in SE are applied in the development of software products based on algorithms, codes, and logical test cases. One of the most important concepts in every stage of software development is reusing software artifacts [33]. Libraries of codes [30], models [13, 19], and meta-models are some of the reusable components that software engineers build. Reusability is a well-established method in the development of software systems in various fields and ecosystems [14, 18]. Systematic reuse as the main concept of software product lines (SPLs) leads to higher quality, less costs, and faster time-to-market [5]. In such software intensive systems, general-purpose building blocks (i.e., core assets) are created to be reused in development of a family of similar software systems. Nevertheless, utilization of reusability concepts is neglected in most sub-domains of developing ANNs. Efforts were made to develop libraries such as Keras [11] and PyTorch [32] to foster the development of ANN-based solutions. However, many aspects of SE are not considered in such solutions. For example, there are several concerns including stability, integrity issues, and conflicts between the dependencies of these libraries. These frequently occurring issues show a lack of maturity in using SE methods for developing ANN tools and frameworks. We assume that the reasons behind the existing gap are: (i) low quality codes written and maintained during the development of ANN-based solutions compared to conventional software systems; (ii) dependency of ANN-based solutions to their application

SPLC '19, September 9–13, 2019, Paris, France

ACM ISBN 978-1-4503-6668-7/19/09…$15.00
https://doi.org/10.1145/3307630.3342419

domains; and (iii) lack of awareness among ANN experts about the SE community and vice versa.

In this paper, we aim for bridging the gap between the communities of SE and ANN. We investigate the state of practice and existing challenges of applying reusability methods in development of ANN-based systems. We study how practitioners and researchers reuse the tools, solutions, and assets and how often reusability was considered while developing ANNs. We formulated our research questions as following:

- **RQ1: How are neural networks already reused?** The motivation behind this question is to estimate the state of reuse among experts of ANN. This helps us to find shortages, weaknesses, and strengths of reuse in this field.
- **RQ2: What types of neural networks are commonly reused and why?** Some of the neural networks are reused more often than the others. Understanding the reasons and factors behind it helps us to realize the features that leads to this preference.
- **RQ3: What are the main challenges related to the reuse of ANNs?** This question specifies the issues with reusing ANNs. A solution to these challenges can pave the way for a significant improvement of reusability in ANN-based projects.

In order to answer these questions, we conducted an empirical study using an online survey following the guideline proposed by Fink [12]. We collected and analyzed the opinion of 112 practitioner who are involved in the development of ANNs. Our contribution is to point out several issues that researchers can resolve to improve the quality of applying reusability techniques in the development of software systems based on ANNs.

In Section 2 we review existing approaches on supporting reuse in development of ANN-based solutions. Section 3 presents the details of our survey. Research questions were answered based on the collected feedback in Section 4. Discussions and conclusions are presented in Section 5 and 6.

2 RELATED WORK

Several studies proposed the utilization of ANNs to support SPLE activities such as Requirements Engineering [28] and management of variability and configurations [7]. Ghamizi [15] also introduced an end-to-end framework for systematic reuse of DNN architectures to create a product line of ANN-based software solutions. To the best of our knowledge, there are no notable empirical studies in the context of combining the reusability concepts and developing ANN-based solutions. However, efforts have been made to foster the development of ANN-based systems [4, 29, 34]. In this section, we address the existing approaches on supporting reusability in the development of ANN-based solutions.

Transfer learning is a common technique in reusing models among DNNs [16, 39] where a model that is trained on a data set will be trained further on another data set as well. This method saves time, costs, and resources for training a DNN. However, this type of reuse is only feasible for models that were trained on a more general data-set [40]. Transfer learning will be utilized in two common situations: lack of enough data or limited resources required to train a model of DNN. ImageNet [26] is one of the most used examples of transfer learning in the domain of image classification. This model is trained to classify 1000 classes of objects in the images and can be extended for other purposes in the domain of image classification. Similar models are also provided by Oxford [1] [37], Google [2] and Microsoft [3] [20]. Transfer learning is not limited to the models of image processing. Reuse of Models for language processing are also supported through approaches such as word2vec Model[4] from Google and GloVe Model [5] provided by Stanford.

A slew of machine learning frameworks are created to handle the complexity in development of ANNs based systems. A handful and easy to use set of APIs are available that support a fast development of ANNs. For instance, PyTorch [6] library provides flexibility in the development of DNNs in Python by facilitating the building of computational graphs. Another tool is TensorFlow [2] which is a framework to handle the complexity of development tasks on ANN-based solutions. In comparison to the other frameworks, it supports wider range of programming languages and platforms. CNTK [36] also is an open-source toolkit provided by Microsoft for handling the tasks related to development of deep-learning based software. Amazon adapted Apache MXNet [10] on its web services due to the scalability of the tool and the number of languages that it supports in the development of deep learning tasks.

3 SURVEY

3.1 Preparation

We had a brainstorming session to prepare the questions about reusability. The questions were:

(1) What is your understanding about reusability?
(2) How does your team apply reuse? on which components? and in which situations? any details?
(3) Which components were reused in your development processes? and why?
(4) What are the positive effects that you try to achieve with reuse? how important are those effects?
(5) Do you or your team apply any tools to support reuse and reusability? Why did you choose this/these tool(s)?

We asked seven experts in SE and three in ANN domains to answer these questions. We collected their answers and categorized them using a few tags. Based on these answers, we extracted questions with ready-to-select options to make the answering process as easy as possible. In the next step, we performed pilot tests with five participants who took part in our survey. The feedbacks and recommendations were then considered to finalize the survey. The survey is available online in PDF format [7]. Furthermore, the anonymized and filtered version of results for our survey is publicly available [8].

[1] http://www.robots.ox.ac.uk/~vgg/research/very_deep/
[2] Google Inception Model: https://github.com/tensorflow/models/tree/master/research/inception
[3] https://github.com/KaimingHe/deep-residual-networks
[4] https://code.google.com/archive/p/word2vec/
[5] https://nlp.stanford.edu/projects/glove/
[6] https://pytorch.org/
[7] https://doi.org/10.6084/m9.figshare.8152664.v3
[8] https://doi.org/10.6084/m9.figshare.8230178.v1

We used LimeSurvey software provided by the education portal of Saxony in Germany[9] to publish our survey and collect responses. The survey was active for seven weeks (from April 8th, 2019 to May 15th, 2019)[10]. We investigated scientific databases, including IEEE Explore, Springer, and ACM to create a list of papers related to ANNs. We sent out more than 1000 emails to the authors of these papers. We also distributed calls for participation within groups and forums of machine learning in ResearchGate, Linkedin, Twitter, Xing and Google.

3.2 Structure

The first page of the survey is filled with a short introduction similar to the content of the email which explains the goal of the survey. The questions with multiple choice answers have a free text field which enables the participants to write their answers the desired answer is not provided already as an option.

The survey consists of five steps: First step (Section A) includes a question to filter the non-relevant participants from the survey:

- A1. Have you ever used or implemented any artificial neural networks in any form?

In the second step (Section B) of the survey, four questions were proposed to collect information about "Most significant experience with ANNs". These questions are listed here:

- B1. When was your most significant project with ANNs?
- B2. How big is the team that you work with on this project (including yourself)?
- B3. Did you use any systematic development process(es) in this project?
- B4. In what way did you work at most with ANNs?

Third section of the survey (Section C) extracts details from participants about their opinions and experiences of reuse in ANNs which consists of seven questions:

- C1. How do you define reusability?
- C2. How often do you reuse the following parts related to artificial neural networks? (code, data-set, and parameters)
- C3. What is the most significant source related to artificial neural networks that you reuse for your projects?
- C4. In your projects, which of the following conditions trigger reuse?
  - First phase of each project
  - When I don't have enough knowledge about the domain
  - I know the domain and I am sure that similar work exists
  - If stakeholder(s) asks for reusing
- C5. How often do you reuse the following kinds of artificial neural networks? (Shallow Artificial Neural Networks (such as MLP, RBM, SOM, RBF, etc.), Deep Convolutional Neural Networks (CNNs), Deep Recurrent Neural Networks (RNNs), Deep Generative Adversarial Networks (GANs), Auto-encoders, Deep Belief Networks, Capsule Neural Networks, Other types of Deep Neural Networks)
- C6. How much do you agree with the following statements about reusing ANNs?
  - Searching for useful reusable modules is complicated/time consuming
  - Making modules reusable is complicated/time consuming
  - Modules are poorly documented, making it difficult to use
  - It is difficult to find out whether existing modules fulfills the task
- C7. How important is reuse in your process predominantly?

In step 4 (Section D), we collected the demographic data about participants and their working domain. Furthermore, we asked them about their background and experience in the development of ANNs. Final step (Section E) was to gather personal feedback about the survey to improve our future work. This step also contains an additional question about contact details to send the results of the survey and get back to the participants for more detailed questions if needed.

4 RESULTS

We received a total number of 229 responses including 114 complete and 115 incomplete surveys. We applied two filters on responses of the survey to increase the quality of our analysis. First filter removed the 115 incomplete surveys since they did not offer a significant information. The second filter discarded two responses from the remaining 114 surveys in which the participants did not select *Yes* as an answer to the filter question *A1.* Applying the previous filters led to 112 surveys in total which are considered to answer the proposed research question in Section 1.

4.1 Demographics

According to the results of the profiling part, we have collected the following answers.

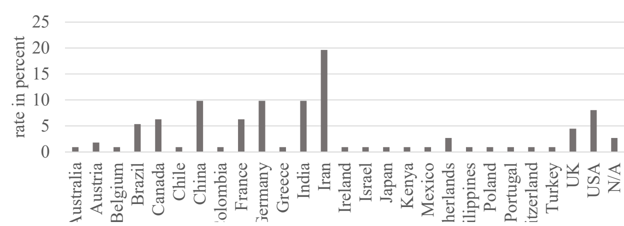

Figure 1: Rate of participants from different countries

As Figure 1 shows, most participants are from Iran (22 participants, 19.64%). China, Germany, India have 11 participants (9.83%) . USA has the third highest participation in our list with 9 participants (8.04%). France and Canada with 7 participants (each 6.25%) are fourth. Brazil with 6 participants (5.36%), UK with 5 participants (4.46%), Netherlands with 3 participants (2.68%) sit at the lower part of our list. We had also 17 (15.13%) participants from other countries: Japan, Pakistan, Australia, Switzerland, Kenya, Poland, Austria, Austria, Mexico, Greece, Chile, Turkey, Belgium, Ireland, Israel, Colombia, Philippines.

Most participants (64 participants, 57.14%) are computer scientists. 37 participants (33.04%) chose "engineering" as their field of

[9] https://bildungsportal.sachsen.de/portal/
[10] https://bildungsportal.sachsen.de/umfragen/limesurvey/index.php/782756?lang=en

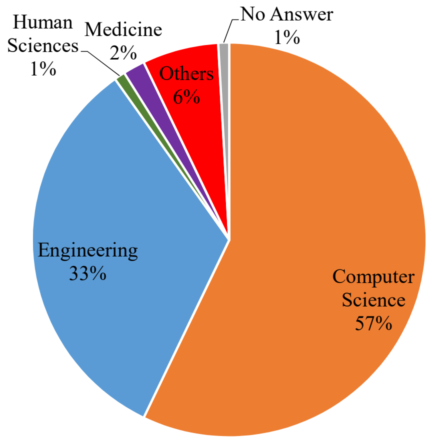

Figure 2: Participants towards their field of study

study. We have 11 participants from other fields such as bio-medical engineering, physics, mathematics and medicine (see Figure 2).

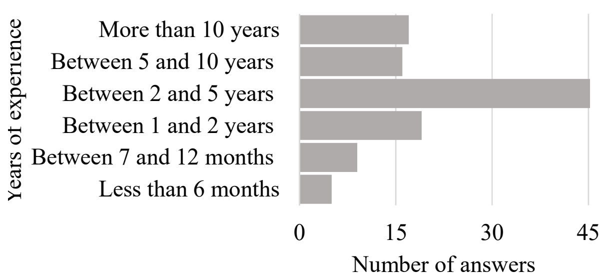

Figure 3: Years of experience with ANN

Almost half of the Participants (46 participants, 41.07%) have between 2 and 5 years of experience/background with ANNs (See Figure 3). Based on the result in Figure 4, 109 participants used ANNs in the context of research, 37 in teaching, 42 in industrial context, and 24 for hobby, There are few answers about working with ANNs in art (2 participants), 15 in private, and 2 in other contexts. Selecting more than one answer to this question was allowed.

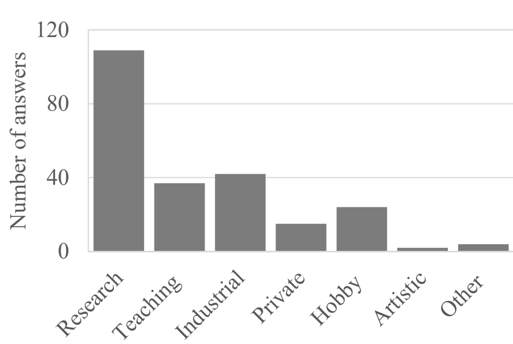

Figure 4: Field of experience with ANN

Question B3 asked participants about development processes that are used in most significant project that they had with ANN. As Figure 5 illustrates, ad-hoc process are used most by the teams that worked with ANNs (63 answers), while some of them used systematic development processes (33 answers).

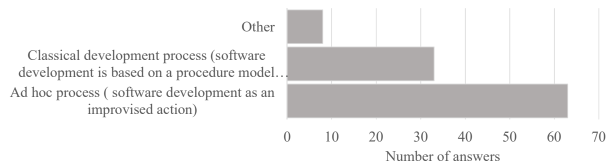

Figure 5: Utilized Development Process in the most significant ANN-based Project

Demographic data shows that many participants have between 2 and 5 years of experience. Most participants come from the fields of computer science and engineering, and they are active in research, teaching, and industry. Furthermore, applying ad-hoc processes for managing the projects is more common among participants in comparison to classical development processes.

4.2 RQ1: How are neural networks already reused?

According to the answers to C1 (Shown in Figure 6), 79 of 112 participants define reusability as "Modify an existing artificial neural network to adopt it to a specific context", while 53 of 112 participants define reusability as "Create a general neural network with the aim of using in various contexts (e.g. convolutional neural networks that have been trained on data-sets like ImageNet, without any changes)". Note that the participants had also the option to select none or both definitions.

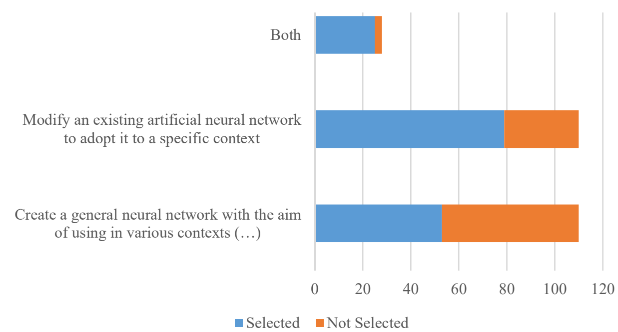

Figure 6: Selected Answers for the question "How do you define Reuse"

Reuse can occur in three ways through ANNs: data-set, network structures (i.e., code), and parameters (i.e., weights). In the first way, the data-set from third party will be reused to train an ANN that the user have developed on his own. Based on the answers to C2, reuse occurs "Often" in "data-set" level among participants. This is the most common abstraction level of reuse in ANNs. Second stage of reuse in ANNs is reuse of specific architectures (such as Alexnet[26], VGG[37], GoogleNet[38], ResNet[20], and DensNet[23]) that can be utilized as network structures (i.e., code) . In this level, an ANN is trained using an arbitrary training set. The weights of the network (parameters) should be tuned to achieve a high degree of accuracy. Participants "Often" reused the structure. Third level of reuse is increased to the values stored in the structure of the network (i.e., parameters). In this approach, a pre-trained network including its layers and weights (i.e., parameters) will be used completely or partially (i.e., transfer learning). Based on the results of question C2, participants "Sometimes" reuse parameters of ANN in their projects.

Figure 7 illustrates that all three parts related to ANNs (code, data-set, and parameters) are somehow reused. Code and data-set are reused more often compared to parameters. Answers to the question C2 confirm that all participants perform reuse in at least one of the three abstraction levels of ANNs.

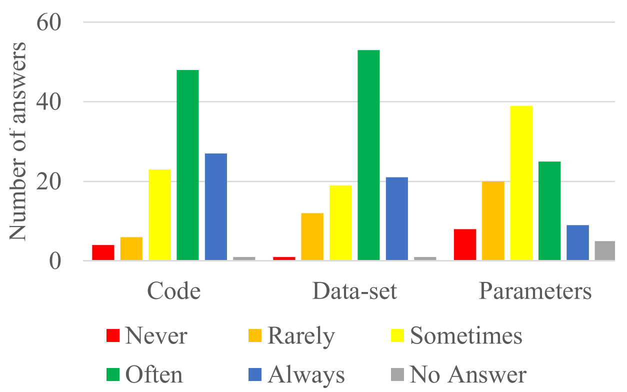

Figure 7: How often different parts related to ANN are reused

It should be noted that most of the participants (79%) use Public repositories, 34% use Private own collections. 16% received it from colleges, and 11% through Other developers via private communications.

Even though the benefits of reuse has been well established in software development, understanding the reasons behind the utilization the reuse in ANN-based projects provide a comprehensive explanation about situations where reuse can be triggered. Therefore, question C4 then asks the participants which conditions trigger reuse in their projects. Figure 8 shows that 76 participants selected the option "I know the domain and I am sure that similar work exists" as the trigger for reuse. 42 participants chose "First phase of each project" and 32 participants mentioned "When I don't have enough knowledge about the domain". 8 participants mentioned that the demand from the stakeholders was the motivation for reuse by selecting "If stakeholder(s) asks for reusing". It should be noted that participants were free to choose more than one of the listed answers.

Question C7 asks how valuable reuse is for participants. It is "Not at all important" for 4.14% of our participants. 23.21% of them chose "Slightly important", 36.61% of participants found it "Important". Finally, it is "Fairly Important" for 18.75% as well as "Very Important" for 11.61% of participants. In total, 66,96% of participants find reuse "important", "fairly important", or "very important".

In summary, reuse is important in the development of ANNs and it happens in all main levels of ANNs' structure. However, the network structures and data-sets will be reused more than the parameters. Furthermore, there are more tendency to modify an ANN and reuse it instead of providing reusable networks.

4.3 RQ2: What types of neural networks are commonly reused and why?

Results of initial interviews and the investigation in literature guided us to eight common types of ANNs that experts reuse in their works. The common types of the NNs and their frequency of reuse is shown in Figure 9. Among proposed ANNs, Deep Belief Networks (DBNs) [21] and Capsule NNs [35] have the most answers on getting "never" reused. However, Deep Convolutional Neural Networks (CNNs) [27] are reused ANNs among participants the most.

We calculated weighted mean values for the Likert scale [3] for each ANN category. Based on the resulting mean values, CNNs get "sometimes" reused. While Deep Recurrent Neural Networks (RNNs), Shallow artificial NNs (such as multilayer perceptron, restricted Boltzmann machine, self-organizing map, radial basis function) [24], auto-encoders [8], and Deep Generative Adversarial Networks (GANs) [17] are reused "rarely".

4.4 RQ3: What are the main challenges related to the reuse of ANNs?

Question C6 measures the degree of agreement of participants with following statements:

- (SQ1) Searching for useful reusable modules is complicated/-time consuming
- (SQ2) Making modules reusable is complicated/time consuming
- (SQ3) Modules are poorly documented, making it difficult to use
- (SQ4) It is difficult to find out whether existing modules fulfills the task

Figure 10 illustrates the overview of the answers collected for this question. A small group of participants is more than "Agree" with this statement. Calculating the average values for given scales, reveals that all four statements are "Neutral" among participants. There are considerable "Disagree" and "Neutral" answers to the statements. Most answers agree with both statements in SQ2 and SQ4. These statements consider the difficulties in creating reusable components and finding whether the existing modules fulfills the task. Despite the agreement of 33% of participants with SQ1, 27% of participants disagree that searching for reusable modules is complicated or time consuming. Level of agreement and disagreement is decreased for SQ3. It shows that 16,96% "Disagree" with the problems around documentations of reusable modules, while 33,04% are "Neutral" and 27,68% "Agree".

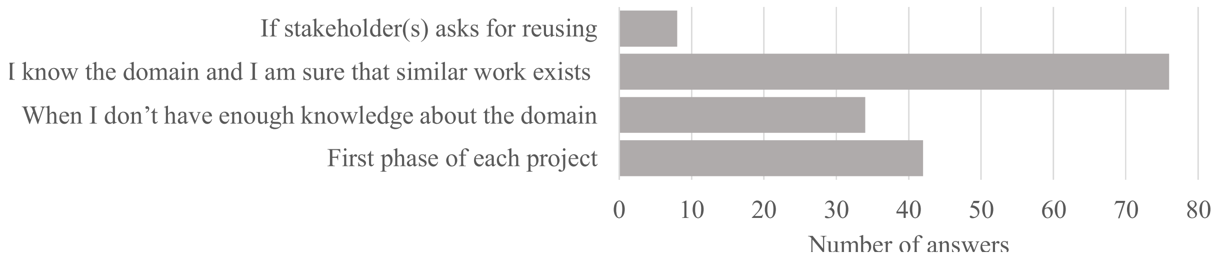

Figure 8: Number of answers to C4 about factors which trigger reuse

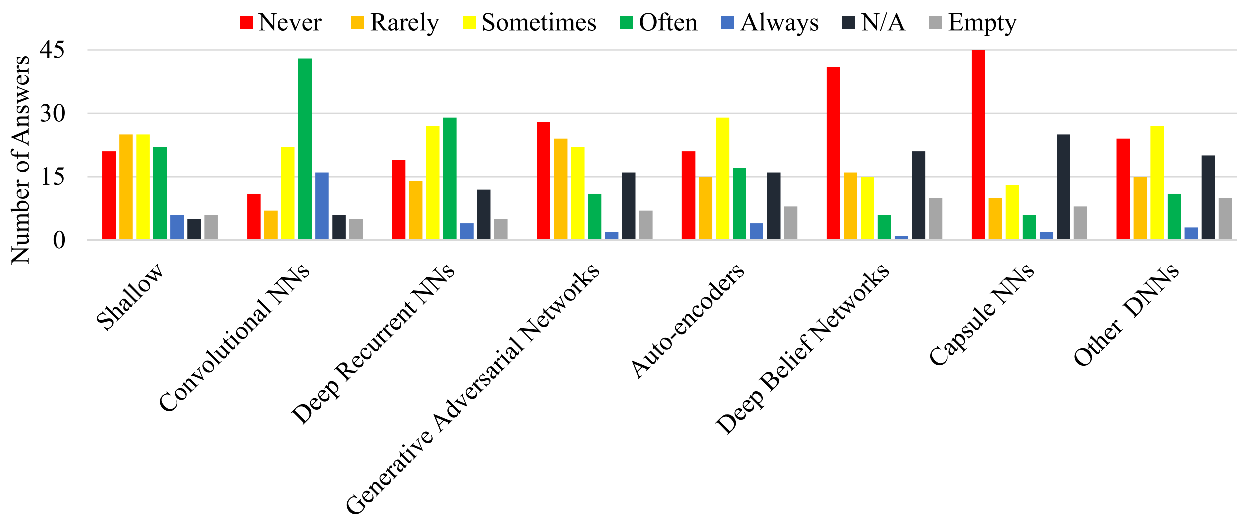

Figure 9: Frequency of reuse for common ANNs

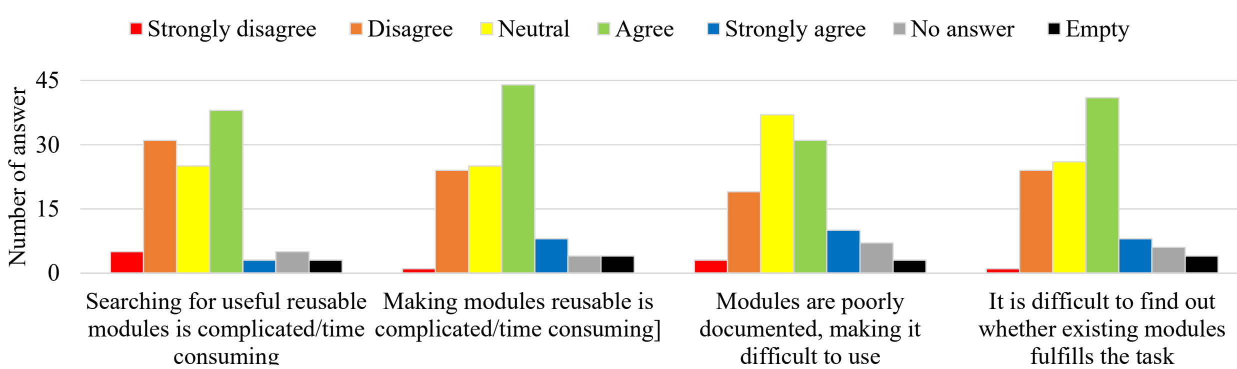

Figure 10: Agreement level of the participants (in percent) towards four statements on challenges in applying reuse to ANNs

5 DISCUSSION

The period of data gathering with our survey was arguably enough as we tried to motivate experts as much as possible. We carefully designed the survey questions to be short but unequivocal so that the participants would not be bored or disappointed.

The number of participants to our survey is 112 which is not a huge set of experts. One reason was that we filtered out many and tried to reach only those that were qualified enough through published scientific works and expert forums. We discarded uncompleted submissions from our results to improve the quality of the survey.

A significant proportion of experts in our survey are from a research community which was predicted as many industrial partners do not publish the contact details of their employees. We tried to reach as many as we could via LinkedIn and online forums. We do not consider the lack of participant from industrial sector to have negative impact on our study since the main objective of our research was to bridge the gap between SE and ANN communities.

Several questions could improve the study that we left out to keep the survey short. Asking these questions could improve the quality of answers. However, there is a risk of reducing the number of participants since the survey takes longer to fill. Furthermore, we collected the email addresses of the interested participants for further questions. We will contact them for detailed questions and create an extension of our study for future work. It may be useful for further studies to know (i) with which kind of deep neural network do they often work, (ii) in their opinion which challenges exists, and (iii) suggestions to improve the reusability of ANNs.

Answering all questions in the field of ANNs and reusability is beyond the scope of our research. Our main objective is put a spotlight on the state of reusability in development of ANN-based projects. Considering the results of our survey can motivate other researchers to create further research questions.

We have considered central tendency bias in addressing the multiple-choice questions by providing examples about proposed answers. However, some questions do not have enough examples or details that could be a bias for choosing an answer. The problem in this case is that there is no proper examples. We avoid using odd-numbered scales with a middle value which could confuse the participants.

Regarding to C5, we categorized the common ANNs based on their conceptual differences in existing implementations of ANNs. That is why some of the well-known ANNs such as Long Short Term Memories (LSTMs) are not included since it is a subset of deep recurrent NNs. Furthermore, we should mention that some of the results were unexpected for this question. For example, implementing a Capsule Net or GANs from scratch is very complicated. One reason that significant portion of answers to the frequency of reuse of such ANNs is "Never" or "Rarely" is the lack of reusability support for these types of ANNs in official distribution of common libraries and tools such as Keras. Reuse in deep convolutional NNs (CNNs) is dominant compared to other architectures listed in C5. This may be because CNNs are the first group of deep ANNs [27] that addressed the issue of computational complexity by utilizing GPUs [26]. Wide range of application domains for these types of ANNs resulted in various tools and frameworks that facilitated their applications.

Mentioned challenges in C6 (i.e., RQ3) are based on the comments collected from preliminary interviews and the pilot test. We were aware of the need for further research to provide a systematic overview on existing challenges, but none of participants left comments about potential challenges which may be ignored in the survey. As shown in Figure 10, the number of participants who "Agree" or "Disagree" in SQ2, SQ3, and SQ4 is considerably different. For example, regarding "It is difficult to find out whether existing modules fulfill the task" 49 participants selected "Agree" or "Strongly agree". 25 participants selected (strongly) disagreed. Based on answers to C1 (see Figure 6), we assume that their disagreement is because of their different perceptions or insufficient understanding of reusability. Both options in C1 are valid definitions of reusability concept, yet a large number of participants did not select both (see Figure 6). Different opinions show that the ANNs experts not only have fundamentally different views, which could be a signal of the lack of knowledge in understanding important concepts—reusability—which exists in the SE domain.

Answers to C7 reveals that reuse is important among experts of ANN. However, the results illustrated in Figure 7 show the low rate of reuse for different types of ANNs (except CNNs) among participants.

Although we asked about systematic development process(es) in B3, further investigations about the knowledge of participants in software development methods is required. Regarding close and fast-growing collaboration between SE and ML, a systematic approach for assessment, validation, and standardization of the level of knowledge among experts from both fields seem necessary.

Besides the important factors mentioned in RQ1, biases and hyperparameters (e.g., number of epochs and learning rates) also play a crucial role in ANNs. We assume that reusing them is not common as none of our interviewees and participants mentioned them in their answers. Therefore, we did not include reusability of biases and hyperparameters in our survey.

We performed an empirical study on reusability applied to the domain of ANNs. However, introducing and investigating all aspects of SPLs (e.g., variability models, configuration, feature interactions) in ANNs and machine learning is not the focus of this paper.

6 CONCLUSIONS

In this paper, we investigated the state of reuse among experts of ANNs using empirical results collected from an online survey in 2019 with 112 participants. Three main research questions were answered: (i) how neural networks will be reused? (ii) what kind of neural networks are commonly reused? and (iii) what are the common challenges related to the reuse of ANNs? Although reuse is applied in the development of ANN-based solutions, the level of reuse and the maturity of knowledge among experts of ANN are arguably low. We found that codes and data-sets related to the development of ANNs are reused more often in comparison to parameters. Furthermore, the public repositories (e.g., GitHub) are the most attended references for reuse. A large number of participants find reuse essential for their projects although they have different opinions about reuse and reusability. Participants mainly apply reuse in their domains of expertise Otherwise, they prefer to develop new solutions from scratch. It seems to be obvious that deep CNNs have more reused compared to other common architectures of ANNs.

Creating reusable modules is one of the challenges that should be considered. Finding out whether existing modules fulfill the requirements of the task is another challenge.

Modules are often poorly documented that makes the utilization of reuse more complicated. The result of our empirical study points out an opportunity for two communities of SE and ANNs for knowledge transfer. We base our future work on bridging the gap especially by establishing the domain of reusability for ANNs.

ACKNOWLEDGMENTS

This project is co-financed with funds on the basis of the budget adopted by the deputies of the Saxon state parliament

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://www.tensorflow.org/ Software available from tensorflow.org.

[2] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*. 265–283.

[3] I Elaine Allen and Christopher A Seaman. 2007. Likert scales and data analyses. *Quality progress* 40, 7 (2007), 64–65.

[4] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 39–48.

[5] Sven Apel, Don Batory, Christian Kästner, and Gunter Saake. 2016. *Feature-oriented software product lines*. Springer.

[6] Anders Arpteg, Björn Brinne, Luka Crnkovic-Friis, and Jan Bosch. 2018. Software engineering challenges of deep learning. In *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, 50–59.

[7] Davide Bacciu, Stefania Gnesi, and Laura Semini. 2015. Using a machine learning approach to implement and evaluate product line features. *arXiv preprint arXiv:1508.03906* (2015).

[8] Pierre Baldi. 2012. Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML workshop on unsupervised and transfer learning*. 37–49.

[9] Michael Batty, Kay W Axhausen, Fosca Giannotti, Alexei Pozdnoukhov, Armando Bazzani, Monica Wachowicz, Georgios Ouzounis, and Yuval Portugali. 2012. Smart cities of the future. *The European Physical Journal Special Topics* 214, 1 (2012), 481–518.

[10] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. 2015. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv preprint arXiv:1512.01274* (2015).

[11] François Chollet et al. 2015. Keras. https://keras.io.

[12] Arlene Fink. 2015. *How to conduct surveys: A step-by-step guide*. Sage Publications.

[13] William Frakes and Carol Terry. 1996. Software reuse: metrics and models. *ACM Computing Surveys (CSUR)* 28, 2 (1996), 415–435.

[14] William B Frakes and Kyo Kang. 2005. Software reuse research: Status and future. *IEEE transactions on Software Engineering* 31, 7 (2005), 529–536.

[15] Salah Ghamizi, Maxime Cordy, Mike Papadakis, and Yves Le Traon. 2019. Automated Search for Configurations of Deep Neural Network Architectures. *arXiv preprint arXiv:1904.04612* (2019).

[16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[18] Martin L Griss, I Jacobson, and P Jonsson. 1998. Software reuse: Architecture, process and organization for business success.. In *TOOLS (26)*. 465.

[19] Haitham Hamza and Mohamed E Fayad. 2002. Model-based software reuse using stable analysis patterns. ECOOP.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *arXiv preprint arXiv:1512.03385* (2015).

[21] Geoffrey E Hinton. 2009. Deep belief networks. *Scholarpedia* 4, 5 (2009), 5947.

[22] Hartmut Hirsch-Kreinsen. 2014. Smart production systems. A new type of industrial process innovation. In *DRUID Society Conference*. 16–18.

[23] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708.

[24] Anil K Jain, Jianchang Mao, and KM Mohiuddin. 1996. Artificial neural networks: A tutorial. *Computer* 3 (1996), 31–44.

[25] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. 2007. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering* 160 (2007), 3–24.

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[27] Yann LeCun, LD Jackel, Leon Bottou, A Brunot, Corinna Cortes, JS Denker, Harris Drucker, I Guyon, UA Muller, Eduard Sackinger, et al. 1995. Comparison of learning algorithms for handwritten digit recognition. In *International conference on artificial neural networks*, Vol. 60. Perth, Australia, 53–60.

[28] Yang Li, Sandro Schulze, and Gunter Saake. 2018. Extracting features from requirements: Achieving accuracy and automation with neural networks. In *2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 477–481.

[29] Jie Lu, Vahid Behbood, Peng Hao, Hua Zuo, Shan Xue, and Guangquan Zhang. 2015. Transfer learning using computational intelligence: a survey. *Knowledge-Based Systems* 80 (2015), 14–23.

[30] Ali Mili, Rym Mili, and Roland T Mittermeir. 1998. A survey of software reuse libraries. *Annals of Software Engineering* 5, 1 (1998), 349–414.

[31] Helena Holmström Olsson and Ivica Crnkovic. [n. d.]. A Taxonomy of Software Engineering Challenges for Machine Learning Systems: An Empirical Investigation. In *Agile Processes in Software Engineering and Extreme Programming: 20th International Conference, XP 2019, Montréal, QC, Canada, May 21–25, 2019, Proceedings*. Springer, 227.

[32] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic Differentiation in PyTorch. In *NIPS Autodiff Workshop*.

[33] Ruben Prieto-Diaz and Peter Freeman. 1987. Classifying software for reusability. *IEEE software* 4, 1 (1987), 6.

[34] Joseph Reisinger, Kenneth O Stanley, and Risto Miikkulainen. 2004. Evolving reusable neural modules. In *Genetic and Evolutionary Computation Conference*. Springer, 69–81.

[35] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. In *Advances in neural information processing systems*. 3856–3866.

[36] Frank Seide and Amit Agarwal. 2016. CNTK: Microsoft's open-source deep-learning toolkit. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2135–2135.

[37] K. Simonyan and A. Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014).

[38] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.

[39] Lisa Torrey and Jude Shavlik. 2010. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI Global, 242–264.

[40] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks?. In *Advances in neural information processing systems*. 3320–3328.