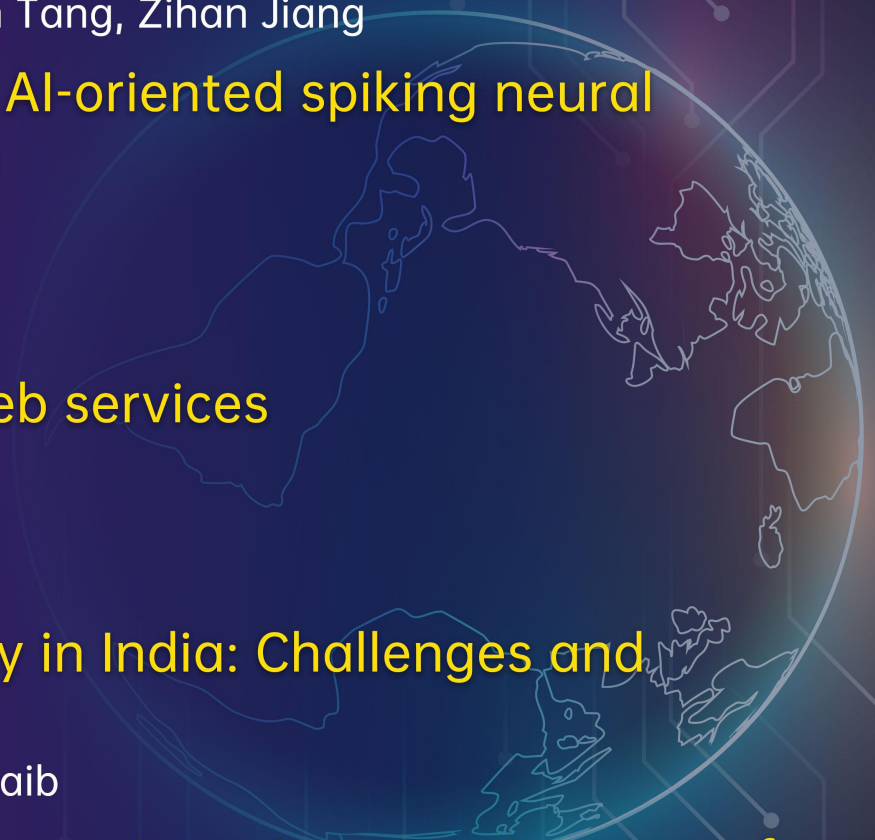
TBench
Volume 3, Issue 1
2023

BenchCouncil Transactions

on Benchmarks, Standards and Evaluations



Original Articles

Review Articles

Reports

ISSN: 2772-4859

BenchCouncil Transactions on Benchmarks, Standards and Evaluations (TBench) is an open-access multi-disciplinary journal dedicated to benchmarks, standards, evaluations, optimizations, and data sets. This journal is a peer-reviewed, subsidized open access journal where The International Open Benchmark Council pays the OA fee. Authors do not have to pay any open access publication fee. However, at least one of the authors must register BenchCouncil International Symposium on Benchmarking, Measuring and Optimizing (Bench) (https://www.benchcouncil.org/bench/) and present their work. It seeks a fast-track publication with an average turnaround time of one month.

Contents

BenchCouncil Transactions on Benchmarks, Standards and Evaluations 3 (2023) 100106

Contents lists available at ScienceDirect

BenchCouncil Transactions on Benchmarks, Standards and Evaluations

journal homepage: www.keaipublishing.com/en/journals/benchcouncil-transactions-on-benchmarks-standards-and-evaluations/

Full length article

ERMDS: A obfuscation dataset for evaluating robustness of learning-based malware detection system

Lichen Jia [a,b], Yang Yang [c], Bowen Tang [a,b], Zihan Jiang [d,*]

[a] State Key Lab of Processors, Institute of Computing Technology, CAS, China
[b] University of the Chinese Academy of Sciences, China
[c] Institute of Technology, China University of Petroleum (Beijing), Karamay Campus, China
[d] Huawei Beijing Research, China

ARTICLE INFO

Keywords:
Dataset
Malware detection system
Security
Machine learning
Adversarial malware

ABSTRACT

Learning-based malware detection systems (LB-MDS) play a crucial role in defending computer systems from malicious attacks. Nevertheless, these systems can be vulnerable to various attacks, which can have significant consequences. Software obfuscation techniques can be used to modify the features of malware, thereby avoiding its classification as malicious by LB-MDS. However, existing portable executable (PE) malware datasets primarily use a single obfuscation technique, which LB-MDS has already learned, leading to a loss of their robustness evaluation ability. Therefore, creating a dataset with diverse features that were not observed during LB-MDS training has become the main challenge in evaluating the robustness of LB-MDS.

We propose a obfuscation dataset ERMDS that solves the problem of evaluating the robustness of LB-MDS by generating malwares with diverse features. When designing this dataset, we created three types of obfuscation spaces, corresponding to binary obfuscation, source code obfuscation, and packing obfuscation. Each obfuscation space has multiple obfuscation techniques, each with different parameters. The obfuscation techniques in these three obfuscation spaces can be used in combination and can be reused. This enables us to theoretically obtain an infinite number of obfuscation combinations, thereby creating malwares with a diverse range of features that have not been captured by LB-MDS.

To assess the effectiveness of the ERMDS obfuscation dataset, we create an instance of the obfuscation dataset called ERMDS-X. By utilizing this dataset, we conducted an evaluation of the robustness of two LB-MDS models, namely MalConv and EMBER, as well as six commercial antivirus software products, which are anonymized as AV1-AV6. The results of our experiments showed that ERMDS-X effectively reveals the limitations in the robustness of existing LB-MDS models, leading to an average accuracy reduction of 20% in LB-MDS and 32% in commercial antivirus software. We conducted a comprehensive analysis of the factors that contributed to the observed accuracy decline in both LB-MDS and commercial antivirus software. We have released the ERMDS-X dataset as an open-source resource, available on GitHub at https://github.com/lcjia94/ERMDS.

1. Introduction

With the rapid progression of technology, machine learning is becoming increasingly sophisticated, prompting researchers to investigate its potential in detecting malware [1–5]. In recent years, machine learning techniques have been employed by researchers to devise more effective approaches for malware detection. One of the key advantages of utilizing machine learning for this purpose is its ability to attain high accuracy rates. This is attributed to the capacity of machine learning models to recognize patterns in malware code that may not be evident to human experts. By training these models on extensive datasets of known malware, researchers can formulate algorithms capable of identifying new malware variants that have not been encountered previously.

However, as the number and complexity of malware threats escalate, conventional signature-based detection methods are losing efficacy. To confront this challenge, commercial antivirus software providers are increasingly embracing machine learning techniques for malware detection [6,7]. By integrating machine learning algorithms into their software, these vendors can elevate the accuracy of their detection capabilities and keep pace with new and emerging threats.

Despite the promising outcomes of using machine learning for malware detection, it is crucial to acknowledge that these models are

* Corresponding author.
E-mail address: jiangzihan.ict@huawei.com (Z. Jiang).

https://doi.org/10.1016/j.tbench.2023.100106
Received 28 March 2023; Received in revised form 23 April 2023; Accepted 23 April 2023
Available online 5 May 2023

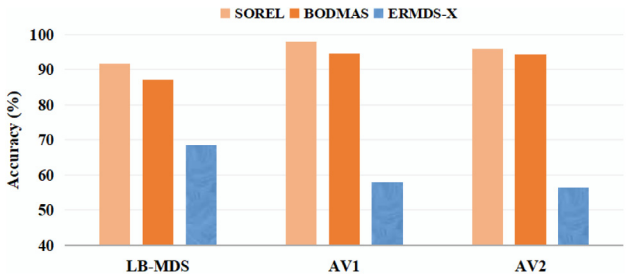


Fig. 1. Accuracy of a LB-MDS (EMBER) and two commercial antivirus softwares (AV1 and AV2) on ERMDS-X (Our dataset) and SOTA malware PE datasets (SOREL and BODMAS).

not impervious to adversarial attacks. Adversarial attacks are methods employed to deceive machine learning models by introducing subtle modifications to input data, with the intention of causing the model to misclassify the data.

Software obfuscation techniques can aid malwares in circumventing detection by LB-MDS. By utilizing obfuscation methods such as encryption and code virtualization, software obfuscation can alter malware features, including file size and API calls, thereby shielding it from detection by LB-MDS.

The concept of robustness in the field of malware detection refers to the ability of detection systems to identify various types of malware and withstand different adversarial attacks. However, the existing PE malware datasets are not sufficient to evaluate the robustness of LB-MDS. To address this, we conducted experiments using two state-of-the-art (SOTA) PE malware datasets, SOREL-20M and BODMAS, on an LB-MDS and two commercial antivirus software, AV1 and AV2. Commercial antivirus software can be categorized as LB-MDS as well, but they do not solely depend on machine learning to identify whether a program is malicious. To differentiate them from the machine learning-based malware detection systems, the term LB-MDS specifically refers to systems that rely entirely on machine learning. Hence, commercial antivirus software is still known as commercial antivirus software. The results are shown in Fig. 1. As seen in Fig. 1, LB-MDS and the two commercial antivirus softwares achieved an average accuracy of over 90% on both the SOREL and BODMAS datasets. Due to the fact that the malware in these datasets employs only a singular obfuscation technique, the features introduced by such obfuscation have already been learned by LB-MDS [8,9]. Consequently, it is impossible for these datasets to diminish the accuracy of LB-MDS. Therefore, they cannot be used to evaluate the robustness of LB-MDS. Since the features in these datasets have been fully assimilated by LB-MDS, rendering them ineffective in reducing the accuracy of the model and thus unsuitable for assessing its robustness.

Numerous researchers have proposed techniques for attacking LB-MDS [10–19]. MAB [10] suggested a series of actions, such as adding a new section to a PE file, and employed reinforcement learning algorithms to select a set of actions that could be applied to malware to produce adversarial examples. Similarly, MalFox [11] introduced a technique in which malware is encrypted and stored in a benign program's section, which is subsequently decrypted and executed at runtime. These approaches reveal the susceptibility of LB-MDS to adversarial attacks.

However, there are several issues associated with the use of these techniques to assess the robustness of LB-MDS. Firstly, the sample size employed in these methods is often insufficient to provide an equitable evaluation of the system's robustness. Typically, 100-1000 malware samples are utilized as input to generate corresponding adversarial examples, which may not accurately represent the extensive range of malware variants that exist in the real world. Secondly, these methods frequently rely on specific techniques, such as the use of encryption in the case of MalFox, to assess the system's robustness, which may not precisely reflect the system's ability to detect other types of obfuscation techniques, such as instruction substitution. Given that commercial antivirus software plays a critical role in security, it is essential to establish a standardized dataset and evaluation methodology to appraise the robustness of LB-MDS.

To address the robustness evaluation problem of LB-MDS, we propose the ERMDS dataset. Unlike prior methods, ERMDS aims to provide a more realistic evaluation of model performance by including a wide array of model-agnostic adversarial examples. These examples are designed to capture various failure modes of modern models, instead of exclusively focusing on worst-case scenarios. When designing this dataset, we created three types of obfuscation spaces, corresponding to binary-level obfuscation, source code-level obfuscation, and packing obfuscation. Each obfuscation space has multiple obfuscation techniques, each with different parameters. The obfuscation techniques in these three obfuscation spaces can be used in combination and can be reused. This enables us to theoretically obtain an infinite number of obfuscation combinations, thereby creating malwares with a diverse range of features that have not been captured by LB-MDS.

To evaluate the ability of the ERMDS obfuscation dataset, we used the obfuscation spaces to generate an instance of the obfuscation dataset called ERMDS-X. This dataset comprises 86,685 malware samples and 30,455 benign samples, with each sample labeled as either malicious or benign. We then used this dataset to evaluate two SOTA LB-MDS models (malConv [2] and EMBER [1]) and six commercial antivirus softwares (AV1-AV6). Through experimentation, we found that ERMDS-X can reduce the accuracy of LB-MDS by an average of 20%, and reduce the accuracy of commercial antivirus software by an average of 32%. By subjecting these detectors to a diverse set of samples, we were able to evaluate their resilience to different types of adversarial attacks and identify areas for improvement. The findings and insights gained from this evaluation are summarized in Table 1, which can inform the design of future LB-MDS.

Apart from presenting the ERMDS dataset, we have also discussed ways to enhance the robustness of malware detectors. We believe that the ERMDS dataset and the proposed methods to improve the robustness of malware detectors will be valuable resources for future research in developing more effective and resilient MDS. By enhancing the robustness of these systems, we can better protect users and organizations from the constantly evolving threat of malware and other cyber attacks.

In summary, this paper has made the following contributions:

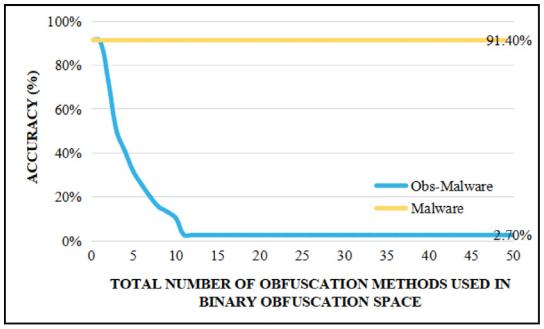
- We propose the ERMDS obfuscation dataset to address the problem that the existing PE malware dataset cannot be used to evaluate the robustness of LB-MDS. We provide a reference implementation of the dataset, ERMDS-X.
- The ERMDS dataset can be utilized for evaluating robustness, and in this study, we utilized ERMDS-X to evaluate the robustness of two LB-MDS models, namely MalConv and EMBER, as well as six commercial antivirus software products, which are anonymized as AV1-AV6. Our experimental results indicate that current LB-MDS models are susceptible to adversarial examples, underscoring the need to enhance their robustness. We have summarized the observations of the LB-MDS systems on the ERMDS dataset in Table 1 and analyzed the underlying reasons for the eight observations.
- We have discussed strategies for improving the robustness of current LB-MDS. We have released the ERMDS-X dataset as an open-source resource.

2. Background

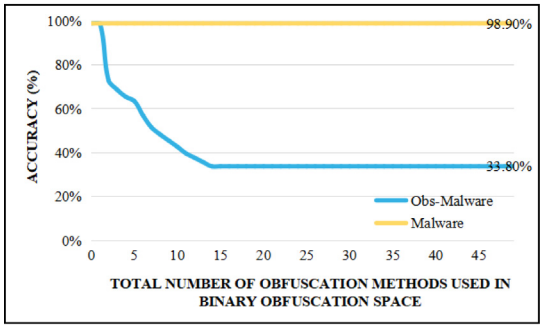
Numerous datasets containing PE malware have been utilized in malware detection research. The EMBER dataset [1], which was introduced in 2018, was the first standardized dataset created specifically for this purpose. It includes 80,000 malware samples collected

Table 1
A summary of major observations and insights grouped by section of the paper.

Observation	Proof	Insight/Explanation
The performance of LB-MDS and commercial antivirus software on the ERMDS-X dataset is much worse compared to their performance on the Clean dataset.	Table 5	Software obfuscation techniques can affect the features of both malicious and benign programs, leading to incorrect conclusions by LB-MDS and commercial antivirus software.
Binary-level obfuscation can significantly reduce the accuracy of LB-MDS by 60%–90%.	Fig. 2	Machine learning models are vulnerable to adversarial attacks, and binary-level obfuscations can more easily create effective adversarial examples.
Binary-level obfuscation only results in a 30% decrease in accuracy for commercial antivirus software.	Fig. 3	Binary obfuscation techniques only have limited ability to modify the code and data of the original program.
LB-MDS will mistake benign programs as malwares.	Fig. 4	Training on more benign program features can reduce false positives.
Source code-level obfuscation increases the probability of misjudging benign programs by LB-MDS.	Fig. 4	Source code-level obfuscation makes benign program code control flow more complex, which leads to misjudgment by LB-MDS.
The misjudgment rate of commercial antivirus software for benign programs is low.	Fig. 4	When unsure, commercial antivirus software tends to classify a program as benign.
Packing technology only decreases the accuracy of EMBER by about 10%.	Fig. 5	Packed programs can be identified by LB-MDS as containing unpacking code, which is considered a feature of malicious programs. This leads to a higher false positive rate for benign programs.
Packing technology can decrease the accuracy of commercial antivirus software by about 60%.	Fig. 6	Packing can completely conceal the features of malicious programs, and benign programs also use packing technology to protect privacy, making it difficult for commercial antivirus software to determine whether a program is malicious based on the presence or absence of unpacking code.



(a) MalConv



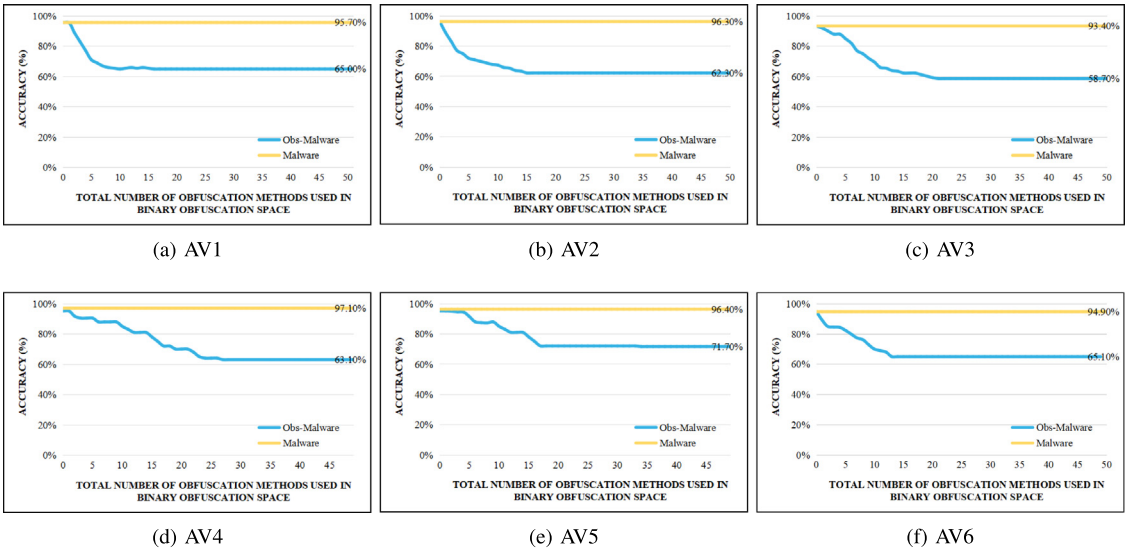
(b) EMBER

Fig. 2. The accuracy of LB-MDS on samples processed through binary obfuscation space.

Algorithm 1: Dataset instance generation algorithm.

Input: Malware dataset $malSet$, Num_b, Num_s, Num_p, l_b, r_b, l_s, r_s, l_p, r_p
Output: Obfuscation dataset $obSet$
$obSet \leftarrow \{\}$
for *each malware sample* $mal \in malSet$ **do**
  for $i = 1$ *to* N_b **do**
    $k \leftarrow randInt(l_b, r_b)$;
    $mal_{binary} \leftarrow$ Applying k rounds of binary obfuscation techniques from Table 2 to the malware;
    $obSet.append(mal_b)$
  end
  for $i = 1$ *to* N_s **do**
    $k \leftarrow randInt(l_s, r_s)$;
    $mal_{source} \leftarrow$ Applying k rounds of source code obfuscation techniques from Table 2 to the malware;
    $obSet.append(mal_s)$
  end
  for $i = 1$ *to* N_p **do**
    $k \leftarrow randInt(l_p, r_p)$;
    $mal_{pack} \leftarrow$ Applying k rounds of packing obfuscation techniques from Table 2 to the malware;
    $obSet.append(mal_{pack})$
  end
end
return $obSet$;

from 2017 to 2018, in addition to 750,00 benign files. Similarly, the SOREL-20M [20] dataset, released in 2019, contains 9 million malware samples collected from 2017 to 2019, as well as 9 million benign files. Although both datasets classify their samples as either malicious or benign, the malware samples in these datasets are sourced from detection websites such as VirusTotal. The malicious samples in these PE malware datasets are relatively outdated and may not represent the latest features of malicious samples. Additionally, the sample features in these datasets can be recognized by LB-MDS and therefore cannot be used to evaluate the robustness of LB-MDS.



(a) AV1 (b) AV2 (c) AV3

(d) AV4 (e) AV5 (f) AV6

Fig. 3. The accuracy of AVs on samples processed through binary obfuscation space.

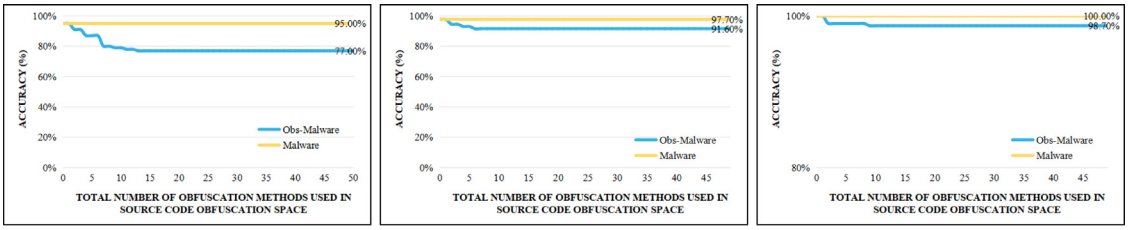
Table 2
Obfuscation methods.

Category	Name	Abbr	Description
Binary Level Obfuscation	Overlay Append	OA	Add additional sections at the end of a binary.
	Section Append	SP	Add randomly generated data to the unused space at the end of a section.
	Section Add	SA	Inserting new sections within the header of a binary.
	Section Rename	SR	Change the name of a section.
	Remove Certificate	RC	Remove the signed certificate.
	Remove Debug	RD	Remove the debug information.
	Break Checksum	BC	Zero out the checksum value.
	Code Randomization	CR	Replace instructions with semantically equivalent instructions.
Source Code Level Obfuscation	Instruction Substitution	IS	This technique involves replacing standard instructions with equivalent but less recognizable ones.
	Code Reordering	CR	This technique involves reordering the instructions in the code to make it harder for attackers to understand the logic of the code.
	Code Flattening	CF	This technique involves converting multi-level if-else statements into a single-level structure.
	Data encryption	DE	This technique involves encrypting sensitive data in the code, such as passwords, keys, and configuration files.
	Code obfuscation through comments	COTC	This technique involves adding comments to the code that are misleading or irrelevant, or that contain obfuscated information.
	Code Metamorphism	CM	This technique involves dynamically modifying the code at runtime, such as by generating code on-the-fly or by modifying the code in memory.
	Control Flow Flattening	CFF	This technique involves modifying the control flow of a program by introducing multiple conditional branches that can be executed in a random order.
	Variable Merging	VM	Combining multiple variables into a single variable to make the code harder to understand.
	Variable Splitting	VS	Splitting a variable into multiple variables to make the code harder to understand.
	Symbol Renaming	SR	Renaming variables, functions, and classes to random or meaningless names to make it harder for a human to understand their purpose and relationships.
	Junk Code Insertion	JOI	Inserting useless or redundant code into the application, making it harder to understand the function of the code.

(*continued on next page*)

Table 2 (*continued*).

Packing	Code Encryption	CE	This technique involves encrypting the executable code of a program in order to prevent it from being understood or modified by an unauthorized user.
	Code Virtualization	CV	This technique involves translating code into specific intermediate representations instead of native instructions and interpreting these representations during runtime.
	Binary Packing	BP	This technique will pack the program. During runtime, the packed program will be unpacked with a custom loader.
	Binary Packing to Benign	BPB	This technique will pack the program and store the packed program in a section of the benign program. During runtime, the packed program will be unpacked with a custom loader.
	API Obfuscation	AO	Hiding the function names and features used by an application programming interface (API), making it harder to understand how the code works.
	Code Compression	CC	Removing unnecessary characters such as whitespace, comments, and newlines to reduce the size of the code and increase analysis difficulty.
	Dynamic Loading	DL	Dividing the code into multiple modules and dynamically loading them when needed to increase code complexity and analysis difficulty.
	Anti-debugging	AD	Adding anti-debugging techniques to the code, such as detecting debuggers or changing the program's execution flow to prevent attackers from debugging and analyzing.
	Anti-decompilation	AP	Adding anti-decompilation techniques to the code, such as adding fake code and control flow to make the results of decompilation unusable.
	Anti-tampering	AT	These are techniques used to detect and prevent modifications to the code. Examples include checking the checksum or hash of the code.
	Anti-disassembly techniques	AS	These are techniques used to prevent an attacker from disassembling the code.
	Anti-emulation	AE	These are techniques used to prevent an attacker from running the code in an emulator.
	Self-modifying code	SMC	This technique involves modifying the code at runtime, making it more difficult to analyze or modify the code.
	Anti-memory Dumping	AMD	These are techniques used to prevent an attacker from dumping the contents of memory to analyze the code. Examples include encrypting memory.



(a) MalConv (b) EMBER (c) AV1-AV6

Fig. 4. The accuracy of LB-MDS on samples processed through source code obfuscation space.

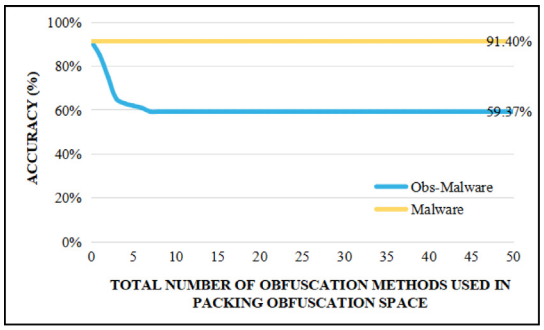
In contrast, the BODMAS dataset [21], released in 2020, contains 70,000 malware samples collected from 2019 to 2020 and 50,000 benign files. Unlike the previous two datasets, the malware samples in BODMAS are labeled with the specific type of malware they belong to, such as ransomware, trojan, or backdoor. This dataset is primarily intended to facilitate LB-MDS in identifying the malware type to which a sample belongs. Hence, it is also not appropriate for evaluating the robustness of LB-MDS.

2.1. Software obfuscation methods

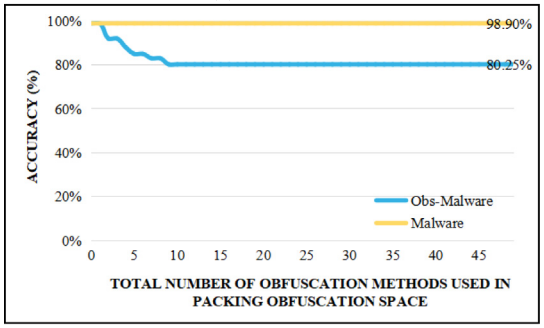
Software obfuscation techniques are utilized to safeguard software code against reverse engineering and analysis, thereby increasing the difficulty for adversaries to comprehend the software's functionality and internal mechanisms. These techniques alter malware while maintaining its functionality, causing LB-MDS to perceive the malicious software as benign programs. Software obfuscation techniques can be categorized into data obfuscation, dynamic code rewriting, and static code rewriting, as surveyed in [22].

2.1.1. Data obfuscation

Data obfuscation techniques [23] involve splitting or merging program data to hinder attackers from analyzing data in the program. For instance, variable splitting splits variables in the program, such as arrays, into multiple sub-arrays. Before accessing the array, the sub-arrays are combined to reform the original array.



(a) MalConv



(b) EMBER

Fig. 5. The accuracy of LB-MDS on samples processed through packing obfuscation space.

2.1.2. Dynamic code rewriting

Dynamic code rewriting techniques [9,24] involve modifying the code at runtime, enabling the dynamic alteration of program behavior. Examples of such techniques include the usage of packers like Ultimate Packer for Executables (UPX) [25] and Themida [26], which apply code obfuscation by encrypting and unpacking the binary program during runtime. Another example is SubVirt [27], which employs code virtualization. This technique transforms the program's code into a specific intermediate representation and interprets this representation at runtime to achieve the same functionality as the original program.

2.1.3. Static code rewriting

Static code rewriting techniques involve transforming the program's code during compilation, eliminating the need for additional modifications during runtime. One such technique is instruction substitution [23,28], which replaces instructions or instruction sequences with semantically equivalent alternatives. For instance, on the Intel x86 platform, the instruction *ADD EAX, 0x1* can be substituted with *SUB EAX, -0x1*.

Dead code insertion [23,28,29] involves constructing code sections that are never executed and injecting code into those sections. This technique aims to confuse or mislead reverse engineers by introducing code that serves no functional purpose.

Control flow obfuscation [28] techniques modify the program's control flow. Control flow flattening, for example, rearranges the program's basic blocks using a switch-case statement. This makes the control flow less transparent and harder to comprehend.

3. The ERMDS dataset

3.1. Methodology

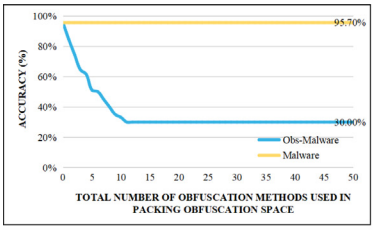
When designing ERMDS, we focused primarily on three questions:

- (q1) How to generate a dataset with diverse features that were not observed during LB-MDS training?
- (q2) How to ensure that each malicious sample has multiple adversarial samples with different features?
- (q3) How to ensure that the functionality of adversarial samples is consistent with that of the original samples?

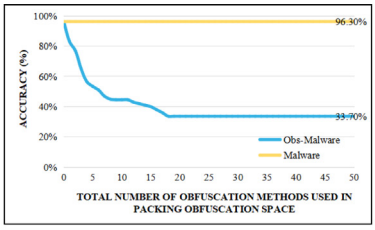
The first question was raised because if all the features in the dataset have already been learned by LB-MDS, then all the samples in the dataset will be correctly classified by LB-MDS, making it impossible to evaluate the robustness of LB-MDS and to determine which LB-MDS is more suitable for security-related applications.

The second question was raised to ensure that every sample in the dataset has multiple adversarial examples with diverse features, as ERMDS aims to provide a more realistic evaluation of model performance by including a wide array of model-agnostic adversarial examples. These examples are designed to capture various failure modes of modern models, instead of exclusively focusing on worst-case scenarios.

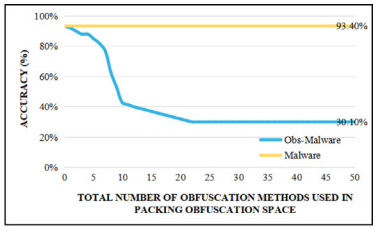
The third question was raised because we need to ensure that the samples in the ERMDS dataset are normal executable programs with intact functionality, so that the decrease in LB-MDS accuracy is not due to damaged sample functionality.



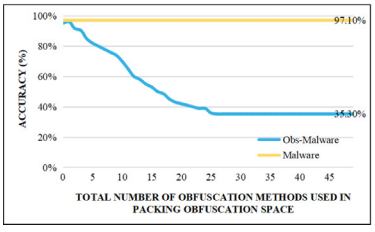
(a) AV1



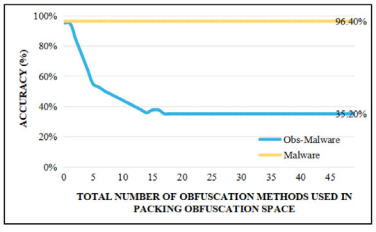
(b) AV2



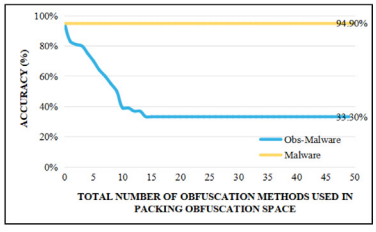
(c) AV3



(d) AV4



(e) AV5



(f) AV6

Fig. 6. The accuracy of AVs on samples processed through binary obfuscation space.

Table 3
Affected features by Obfuscation methods.

		Hash-Based features		Rule-based features							Data distribution
		File hash	Section hash	Section count	Section name	Section padding	Debug info	Checksum	API calls	Code sequence	Data distribution
Binary Level Obfuscation	OA	✓									✓
	SP	✓	✓			✓					
	SA	✓	✓	✓	✓						✓
	SR	✓			✓						
	RC	✓									
	RD	✓	✓	✓			✓				
	BC	✓						✓			
	CR	✓	✓							✓	
Source Code Level Obfuscation	IS	✓	✓							✓	
	CR	✓	✓							✓	
	CF	✓	✓	✓						✓	
	DE	✓	✓								✓
	COTC	✓	✓				✓				
	CM	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	CFF	✓	✓	✓	✓				✓	✓	✓
	VM	✓	✓			✓				✓	✓
	VS	✓	✓			✓				✓	✓
	SR	✓			✓						
	JOI	✓	✓	✓	✓	✓			✓	✓	✓
Packing Obfuscation	CE	✓	✓	✓	✓	✓			✓	✓	
	CV	✓	✓	✓	✓	✓			✓	✓	✓
	BP	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	BPB	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	AO	✓	✓						✓	✓	✓
	CC	✓	✓	✓	✓	✓			✓	✓	
	DL	✓	✓	✓					✓	✓	✓
	AD	✓	✓						✓	✓	✓
	AP	✓	✓						✓	✓	✓
	AT	✓	✓						✓	✓	✓
	AS	✓	✓						✓	✓	✓
	AE	✓	✓						✓	✓	✓
	SMC	✓	✓	✓	✓	✓		✓	✓	✓	✓
	AMD	✓	✓	✓	✓					✓	✓

3.2. The obfuscation space of ERMDS

Solution for q1. To address the first question, we designed three layers of obfuscation, which correspond to binary-level obfuscation, source code-level obfuscation, and packing obfuscation. Binary-level obfuscation modifies section names and removes debug tables from the malware to eliminate sensitive information. This makes it difficult for the MDS to identify the malware as harmful by matching specific strings or symbol information. Source code-level obfuscation rewrites the control or data flow of the malware to avoid detection based on its code or data features. This method is more advanced than binary-level obfuscation because it rewrites the code and data. Packing is the most advanced method that encrypts the entire malware, incorporates it into a benign program, and decrypts and executes it during runtime. Compared to the other two methods, packing is the most sophisticated technique since the MDS cannot obtain any features of the malware through static detection since it is encrypted.

As shown in Table 2, each layer of obfuscation includes at least eight different obfuscation methods, each with specific parameter settings. These obfuscation methods are orthogonal and can be applied repeatedly, allowing for theoretically infinite combinations of obfuscation methods. This enables us to generate a dataset with diverse features that were not observed during LB-MDS training.

In order to ensure that each obfuscation method in our obfuscation spaces can cover all features of the malware, we first divided the features of the program into Hash-based features, Rule-based features, and Data Distribution according to [10]. We also annotated which features each method would affect. If an obfuscation method *om* modifies the feature set $S = \{s_1, s_2, \ldots, s_k\}$ of a malware sample, the impact of various obfuscation methods on the affected features can be observed in Table 3 for our dataset. For example, consider the CR obfuscation method, which replaces instruction sequences with semantically equivalent ones. Referring to Table 3, we can see that this method affects the File Hash, Section Hash, and Code Sequence features. File Hash and Section Hash are affected by any modifications made to the file and section content, while Code Sequence is only affected if the CR obfuscation method modifies the instruction sequences in the code. As shown in Table 3, each obfuscation method in our obfuscation spaces can cover all features of the program, ensuring that ERMDS can generate malware with diverse features.

Solution for q2. For problem two, since the obfuscation combinations we generate are theoretically infinite, each combination applied to a malware produces a variant of the original malware. Therefore, ERMDS can theoretically produce an infinite number of variants for each malicious sample, ensuring that each sample has multiple adversarial samples with different features.

Solution for q3. Regarding problem three, the obfuscation methods selected in our obfuscation space are functionality-preserving. In theory, applying these methods to a program should not change its original functionality because these methods, as shown in Table 2, are all designed to preserve the program's functionality [10,11]. However, due to implementation issues, a small number of programs may become non-functional. To quantify the impact of implementation errors on program functionality, we conducted additional experiments to evaluate the effect of obfuscation methods in the ERMDS dataset on program functionality. Through this experiment, we discovered that the obfuscation methods in the ERMDS dataset can indeed affect the original functionality of functions. For details, please refer to the Functional Integrity Testing section in the appendix. Furthermore, other researchers can expand the ERMDS by incorporating additional obfuscation techniques, resulting in a more comprehensive and diverse dataset of malware samples.

Table 4
Overview of ERMDS-X dataset and quality thresholds on four datasets.

Level	malware nums	benign nums	Quality threshold of LB-MDS accuracy %	Quality threshold of commercial antivirus softwares accuracy %
Binary obfuscation	49714	16815	18.25	64.32
Source Code obfuscation	0	3841	84.3	98.7
Packing	36971	9799	79.81	32.93
ERMDS-X	86685	30455	62.35	62.51

3.3. Workflow of ERMDS

Algorithm 1 outlines the workflow of the ERMDS, which generates a obfuscation dataset $obSet$. For each malware sample mal in the $malSet$, three types of obfuscation techniques, namely binary obfuscation, source code obfuscation, and packer, are applied to mal to produce a set of obfuscated malware samples, which are then added to $obSet$. Num_b, Num_s, and Num_p denote the number of binary obfuscation samples, source code obfuscation samples, and packer samples that need to be generated for each mal, respectively. When generating the obfuscation samples, we randomly choose which obfuscation techniques to apply. To ensure the diversity of the generated samples, we perform k rounds of selection for each obfuscation technique, resulting in a sequence of obfuscation methods $O = \{O_1, O_2, \ldots, O_k\}$, where each element represents a specific obfuscation method. The value of k is a random number, and l_b and r_b are the upper and lower bounds for binary obfuscation techniques, l_s and r_s are the upper and lower bounds for source code obfuscation techniques, and l_p and r_p are the upper and lower bounds for packer techniques.

4. Implementation

4.1. Initial dataset description

As the majority of current datasets for malware analysis only contain samples from the period between 2017 and 2020, including the most recently released BODMS, there is a risk that these datasets may not accurately reflect recent malware behaviors. To address this issue, we intend to release a new malware dataset that covers samples from January to December 2022. Our initial dataset contains 10,000 malware samples, 5000 benign samples, and 300 benign samples with source codes, totaling 15,300 samples. We collected the malware samples from VirusShare [30], ensuring that they were collected between January 1, 2022, and December 30, 2022. The benign samples were collected from Github and Source Forge.

4.2. Generate ERMDS-X dataset instance

We used Algorithm 1 to generate an instance of our dataset, named ERMDS-X, with the following parameters: $Num_b = 30$, $Num_s = 30$, $Num_p = 30$, $l_b = 2$, r_{50}, $l_s = 2$, $r_s = 50$, $l_p = 2$, and $r_p = 50$. The malware dataset $malSet$ consisted of 4000 malware and 1500 benign samples, which were random sampled from the initial dataset. The parameters used in our ERMDS-X dataset instance, including $Num_b = 30$, $Num_s = 30$, $Num_p = 30$, $l_b = 2$, r_{50}, $l_s = 2$, $r_s = 50$, $l_p = 2$, and $r_p = 50$, were not chosen to minimize the accuracy drop of LB-MDS. Instead, the ERMDS dataset was designed to provide a more realistic evaluation of model performance by incorporating a diverse set of model-agnostic adversarial examples. These examples aim to capture various failure modes of modern models, rather than focusing solely on worst-case scenarios. In the Parameters section of the Appendix, we provide a set of optimal parameters that can minimize the accuracy drop of LB-MDS.

After filtering out some malware that could not be processed, we obtained a total of 86,685 malicious and 30,455 benign samples for the ERMDS-X dataset. We extracted the features from PE files using the LIEF [31] project and followed the same format as Ember [1], SOREL-20M [20], and BODMAS [21] to ensure compatibility with existing datasets. Each sample in the ERMDS-X dataset is labeled either "malware" or "benign", providing a ground-truth label for researchers. The techniques used are listed in Table 2, and were implemented using the following tools: Binary Ninja [32], Radare2 [33], IDA Pro [34], LLVM [35], Pin [36], Angr [37], and MAB [10] for binary obfuscation; OLLVM (Obfuscator-LLVM) [28], PreEmptive Protection - Dotfuscator [38], and ConfuserEx [39] for source code obfuscation; and Themida [26], UPX [25], ASProtect [40], Enigma [41], Virbox Protector [42], VMProtect [43], and MalFox [11] for packing obfuscation.

4.3. Description on ERMDS-X dataset instance

The ERMDS-X dataset serves as a valuable tool for evaluating the robustness of LB-MDS and facilitating the identification of potential vulnerabilities within the LB-MDS system. Moreover, LB-MDS can be trained again on the ERMDS-X dataset to enhance its robustness. An overview of the ERMDS-X dataset is provided in Table 4, which comprises three sub-datasets: the Binary obfuscation dataset, the Packing dataset, and the Source Code obfuscation dataset. Each of these sub-datasets includes samples that have undergone binary-level obfuscation, packing, and source code obfuscation, respectively. The ERMDS-X dataset is the combination of these three sub-datasets.

As a result of the absence of source code in malware, we limited the application of source-level obfuscation to benign programs that have source code. It is possible to achieve better accuracy reduction through source-level obfuscation if there is enough source code available for malware. Although this presents a drawback of ERMDS-X, our experiments have demonstrated that ERMDS-X is sufficient for evaluating the robustness of MDS. Other researchers can readily expand the source-level obfuscation dataset by providing malware with source code.

Additionally, we present the quality threshold of LB-MDS on these four datasets, including the quality threshold of LB-MDS and commercial antivirus software. The quality threshold of LB-MDS is determined by averaging the accuracy of two LB-MDS models, namely MalConv and Ember, selected during our experiments. The quality threshold of commercial antivirus software is determined by averaging the accuracy of six commercial antivirus software programs chosen during our experiments.

5. Evaluation

Our evaluation aims to address the following research questions:

- RQ1: How do LB-MDS and commercial antivirus software perform on the ERMDS-X dataset and the SOTA PE malware datasets?
- RQ2: What is the impact of the three categories of software obfuscation techniques, namely binary-level obfuscation, source code-level obfuscation, and packing, on the effectiveness of LB-MDS and commercial antivirus software?
- RQ3: Which malware features have the greatest impact on the classification results of LB-MDS?

Table 5
Accuracy of learning-based malware detection systems on ERMDS-X and Clean datasets.

Model	$E^{Network}_{Clean}$ (%)	$E^{Network}_{ERMDS}$ (%)
malConv	78.57	56.17
EMBER	85.37	68.53
AV1	94.62	65.18
AV2	96.87	66.29
AV3	95.45	58.01
AV4	95.88	56.45
AV5	93.33	72.83
AV6	96.41	56.31

Table 6
Accuracy of learning-based malware detection systems on SOREL-20M and BODMAS datasets.

Model	$E^{Network}_{SOREL}$ (%)	$E^{Network}_{BODMAS}$ (%)
malConv	87.9	82.2
EMBER	91.7	87.1
AV1	93.7	96.3
AV2	92.5	96.8
AV3	97.9	94.5
AV4	96.1	93.8
AV5	97.3	94.7
AV6	96.0	94.4

5.1. Evaluation metrics

In order to comprehensively evaluate the robustness of a MDS, we begin by selecting an MDS and then calculating its accuracy on a clean dataset, which serves as the initial dataset. This calculation is performed using Eq. (1) according to [44], where E^{MDS}_{Clean} denotes the MDS's accuracy on Clean dataset. Specifically, N^{C}_{Clean} represents the number of samples that are correctly predicted by the MDS, while N^{All}_{Clean} represents the total number of samples in the initial dataset.

$$E^{MDS}_{Clean} = N^{C}_{Clean} / N^{All}_{Clean} \tag{1}$$

Subsequently, the selected MDS is tested on the ERMDS-X dataset (denoted as "ERMDS"), and the accuracy, denoted as E^{MDS}_{ERMDS}, is calculated using Eq. (2) according to [44]. Here, N^{C}_{ERMDS} represents the number of samples that are correctly predicted, while N^{All}_{ERMDS} represents the total number of samples in the ERMDS-X dataset.

$$E^{MDS}_{ERMDS} = N^{C}_{ERMDS} / N^{All}_{ERMDS} \tag{2}$$

5.2. Attack targets

For our target models, we have chosen the following:

- EMBER [1] is an open-source LB-MDS. It utilizes LIEF [31] to extract features from both malicious software and benign programs. These features are then used by a LightGBM model to determine whether a program is malicious or not. We utilized the model provided by MLSEC 2019 as our target for the attack [45].
- MalConv [2], in contrast to EMBER, directly uses the binary byte stream of malicious software as training data. Based on this byte stream, it determines whether a program is malicious. We employed the model provided by MLSEC 2019 as our target for the attack [45].
- Commercial Antivirus Software. We selected six top commercial antivirus software as our evaluation targets, based on [46].

5.3. Evaluation on ERMDS-X dataset

This experiment demonstrates that ERMDS-X effectively exposes the robustness limitations of existing LB-MDS models. In Table 5, a comprehensive comparison of two LB-MDS models and six commercial antivirus software is presented based on their detection performance on both the ERMDS-X and Clean datasets. The Clean dataset is a collection of original data containing both malwares and benign programs. Notably, MalConv exhibits a significantly lower accuracy of only 78.57% on the Clean dataset in comparison to EMBER and the six commercial antivirus software, which all have an accuracy of over 85%. Our analysis suggests that this decrease in accuracy is due to the outdated dataset used by the MalConv model during training, which failed to capture the latest features of malwares. In addition, research conducted in [21] demonstrates that virus features change over time, and previously trained models may have decreased accuracy on new malwares.

The performance of the LB-MDS models and commercial antivirus software on the ERMDS-X dataset demonstrates an accuracy range of 56.17% to 78.83%, with an average accuracy of 62.47%. This lower accuracy is attributed to the different types of adversarial examples present in the ERMDS-X dataset, which aims to provide a more realistic evaluation of model performance by including a broad range of model-agnostic adversarial examples. These adversarial examples are designed to capture various failure modes of modern models, rather than focusing solely on worst-case scenarios. Thus, the performance of the LB-MDS models on the ERMDS-X dataset did not decrease to the lowest level, but an accuracy of 62.47% is still a relatively low value, demonstrating the ability of ERMDS-X to evaluate the robustness of existing LB-MDS models.

It is essential to note that LB-MDS systems predict whether a given software is malicious or benign. Even with random guessing, there is a 50% probability of correctly guessing. The accuracy of these systems on the ERMDS-X dataset is 62.47%, which is only 12.47% higher than random guessing. Thus, ERMDS-X can be used to evaluate the robustness of existing LB-MDS models effectively.

5.4. Evaluation on SOTA malware datasets

This experiment aims to demonstrate that the SOTA malware datasets are not suitable for evaluating the robustness of LB-MDS. We evaluated the accuracy of two SOTA PE malware datasets, SOREL-20M and BODMAS, using two LB-MDS models and six commercial antivirus software. To ensure a fair comparison, we randomly selected 10,000 samples from each dataset and employed them to attack the two LB-MDS models and six commercial antivirus software, with each experiment repeated five times to obtain average results.

Table 6 illustrates the accuracy of the two LB-MDS models and six commercial antivirus software on the SOREL-20M and BODMAS datasets. It is observed that all the commercial antivirus software exhibit an accuracy of over 90% for detecting samples from both SOREL and BODMAS datasets. In contrast to Section 5.3, the accuracy of commercial antivirus software for ERMDS-X ranges from 56.31% to 72.83%. Although the LB-MDS models attain an accuracy of over 80% for detecting samples from both SOREL and BODMAS datasets, their accuracy for ERMDS-X is only 56.17% and 68.53%, respectively. This implies that ERMDS-X can assess the robustness of MDS, whereas other datasets like SOREL and BODMAS cannot significantly impact the accuracy of MDS.

Moreover, as evident from Table 6, malConv and EMBER display lower accuracy on the BODMAS dataset than on the SOREL dataset. Additionally, Section 5.3 indicates that malConv and EMBER exhibit lower accuracy on the clean dataset. This is due to the fact that the clean dataset was gathered in 2022, which is subsequent to the data collection period of SOREL and BODMAS datasets (2017–2020). EMBER and malConv were trained on a relatively outdated dataset before 2018 and were unable to accurately capture the features of new viruses, resulting in their lower accuracy.

5.5. Evaluation on three obfuscation spaces

In order to evaluate the effectiveness of LB-MDS on three types of obfuscation spaces, namely binary obfuscation space (BOS), source code obfuscation space (SOS), and packing obfuscation space (POS), we generated adversarial examples using each of these three techniques and evaluated the accuracy of LB-MDS. Furthermore, we conducted an analysis on the samples that caused a decrease in the accuracy of LB-MDS in each obfuscation space, counted the frequency of each obfuscation method used in each obfuscation space, and included a set of parameters in the appendix that can induce the maximum decrease in LB-MDS accuracy, which can serve as a reference for other researchers.

5.5.1. Effect on binary obfuscation space

To evaluate the quality of adversarial examples generated by the BOS, we randomly selected 10000 malicious samples from the initial dataset and named it dataset-V. For each malicious sample in dataset-V, we applied obfuscation techniques from the binary obfuscation space iteratively until an effective adversarial example was produced. We used the obfuscation methods from the BOS to attack MalConv, EMBER, and six commercial antivirus software, and evaluated the number of obfuscation iterations needed to generate an effective adversarial example, as illustrated in Fig. 2. To ensure experimental accuracy, each experiment was repeated five times to obtain the average results.

Attacking LB-MDS. Fig. 2 illustrates the efficacy of BOS attacks on EMBER and MalConv. As depicted in Fig. 2(a), MalConv's original accuracy on malware is 91.40%. However, when detecting malware processed by BOS, the accuracy plummets to 2.70%, resulting in an 88.7% decline in the accuracy. In Fig. 2(b), we observe that EMBER's original accuracy on malware is 98.90%. However, after being processed by BOS, the accuracy drops to 33.8%, resulting in a 65.1% reduction in the accuracy. These results highlight that BOS attacks can easily deceive LB-MDS.

Furthermore, we evaluated the relationship between the number of obfuscation methods and the accuracy of LB-MDS. As depicted in Fig. 2, for MalConv, after undergoing ten binary obfuscation methods, the accuracy of malware decreased to its lowest point, dropping from 91.4% to 2.7%. Even when continuing to apply binary obfuscation techniques to the malware, MalConv's accuracy did not further decrease after ten obfuscation methods. For EMBER, after undergoing thirteen binary obfuscation methods, the accuracy of malware decreased to its lowest point, dropping from 98.9% to 33.8%. Even when continuing to apply binary obfuscation techniques to the malware, EMBER's accuracy did not further decrease after thirteen obfuscation methods. This experiment highlights the limitations of binary obfuscation techniques in combating LB-MDS, as they typically only add new content and have limited ability to modify the code and data of the original program. Therefore, binary obfuscation techniques cannot fully defeat LB-MDS.

Attacking Commercial Antivirus. We conducted a comprehensive evaluation of our framework using six commercially available antivirus engines. Fig. 3 displays the original accuracy of malware for AV1-AV6, which ranged from 93.40% to 97.10%, with AV4 achieving the highest rate at 97.10% and AV3 the lowest rate at 93.40%. After applying BOS processing to the malware and subjecting it to detection by AV1-AV6, we observed a significant reduction in accuracy for the processed malware. For instance, the accuracy for AV1 dropped from an original 95.70% to 65% after BOS processing, leading to 29.3% of the malware evading detection.

Our findings indicate that while binary obfuscation techniques exhibits a certain level of effectiveness against commercial antivirus software, it is not entirely successful in defeating it. This is mainly because commercial antivirus software uses multiple features to determine whether a program is malicious or benign. Binary obfuscation techniques have limited ability to modify the code and data of a program, such as being unable to modify the API calls feature of a program, leading to suboptimal performance in terms of adversarial effectiveness against commercial antivirus software.

Additionally, we analyzed the impact of the number of binary obfuscation techniques on the accuracy of commercial antivirus software. Fig. 3 shows that the number of binary obfuscation techniques had varying degrees of influence on the accuracy of the different antivirus engines. For example, malware detection probability for AV1 decreased to its lowest value of 65% after the 8th binary obfuscation, while for AV4, it decreased to its lowest value of 63.1% after the 27th binary obfuscation. This indicates that different commercial antivirus software exhibits varying degrees of sensitivity to binary obfuscation.

5.5.2. Effect on source code obfuscation space

To evaluate the quality of adversarial examples generated by the SOS, we were unable to use malware samples due to their lack of source code. Therefore, we only provided benign programs to SOS for obfuscation and tested whether the obfuscated benign programs were misclassified as malware by LB-MDS. We randomly selected 300 benign samples from the initial dataset and named it dataset-B. For each benign sample in dataset-B, we applied obfuscation techniques from the binary obfuscation space iteratively until an effective adversarial example was produced. We used the obfuscation methods from the SOS to attack MalConv, EMBER, and six commercial antivirus software, and evaluated the number of obfuscation iterations required to generate an effective adversarial example, as depicted in Fig. 4. To ensure experimental accuracy, each experiment was repeated five times to obtain the average results.

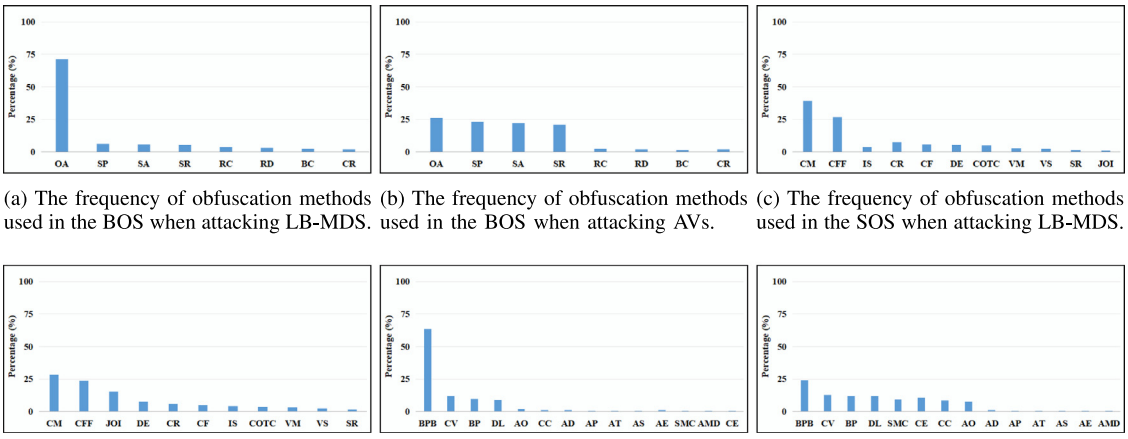
Fig. 5 illustrates the accuracy of EMBER, MalConv, and six commercial antivirus engines in detecting benign programs processed by SOS. MalConv and EMBER exhibit a certain false positive rate for benign programs, with accuracies of 95% and 97.7%, respectively, indicating that MalConv misclassifies 5% of benign programs as malicious and EMBER misclassifies 2.3%, as shown in (a) and (b). After SOS processing, 15 rounds of source code obfuscation cause MalConv's accuracy to decrease to its lowest value of 77%, and 6 rounds of source code obfuscation cause EMBER's accuracy to decrease to its lowest value of 91.6%. This may be due to the fact that source code obfuscation completely disrupts program control flow, making them appear unlike normal programs, leading LB-LDS to misclassify them as malicious.

For commercial antivirus software, the accuracy for benign programs is 100%, and it can also achieve an accuracy of 98.7% for benign programs processed by SOS. This, in conjunction with Table 5, indicates that although LB-LDS has high accuracy for malware, it also has a certain false negative rate for benign programs processed by obfuscation techniques. Furthermore, Fig. 2 shows that LB-LDS is vulnerable to adversarial examples. Therefore, commercial antivirus software outperforms LB-LDS in detecting obfuscations at both the binary and source code levels.

5.5.3. Effect on packing obfuscation space

To ensure fairness, we evaluated the quality of adversarial examples generated by the POS using the same dataset (dataset-V) as in the BOS evaluation. For each malicious sample in dataset-V, we applied obfuscation techniques from the POS iteratively until an effective adversarial example was produced. We used the obfuscation methods from the POS to attack MalConv, EMBER, and six commercial antivirus software. To ensure experimental accuracy, each experiment was repeated five times to obtain the average results.

Attacking LB-MDS. Fig. 5 illustrates the attack effects of the POS on MalConv and EMBER. As shown in the figure, the malicious samples processed by the POS can significantly reduce the accuracy of MalConv from 91.4% to 59.4%. However, the impact on the accuracy of EMBER is relatively small, only reducing its accuracy from 98.9% to 80.25%. The poor performance of the POS on LB-MDS is mainly due to the fact that the POS will pack the malicious software, which requires adding corresponding unpacking code to ensure the correct execution of the malicious software. LB-MDS can capture the features of the



(a) The frequency of obfuscation methods used in the BOS when attacking LB-MDS.
(b) The frequency of obfuscation methods used in the BOS when attacking AVs.
(c) The frequency of obfuscation methods used in the SOS when attacking LB-MDS.
(d) The frequency of obfuscation methods used in the SOS when attacking AVs.
(e) The frequency of obfuscation methods used in the POS when attacking LB-MDS.
(f) The frequency of obfuscation methods used in the POS when attacking AVs.

Fig. 7. The frequency of obfuscation methods used in generating samples that cause a decrease in the accuracy of LB-MDS.

unpacking code, which makes the performance of the POS not ideal when attacking LB-MDS.

Attacking Commercial Antivirus. Fig. 6 illustrates the effectiveness of POS in generating adversarial examples against six commercial antivirus software, namely AV1-Av6. In comparison to LB-MDS, POS demonstrates considerably superior performance in attacking commercial antivirus software. The accuracy of all six commercial antivirus software in detecting original malware samples ranges from 93.4% to 98.9%. However, when presented with malware obfuscated by POS, their accuracy drops dramatically to between 30.1% and 33.7%. This significant decrease in accuracy can be attributed to the fact that many packing obfuscation methods, such as BPB, encrypt the entire malware, resulting in modifications to all features of the malware. Moreover, the encrypted malware is stored in the section of benign programs, causing commercial antivirus software to misclassify it as benign software.

5.5.4. Most frequently used obfuscation methods analysis

In this experiment, we analyzed the samples that caused a decrease in the precision of LB-MDS in the three aforementioned experiments, and calculated the frequency of obfuscation methods used in these samples. Understanding the reasons for the decrease in accuracy of LB-MDS can help improve the robustness of a classifier against adversarial attacks. We have summarized the most frequently used obfuscation methods in Fig. 7 Based on this figure, we can infer the root cause of each evasion. Our findings indicate that:

- When attacking LB-MDS, OA in the binary obfuscation space, CM, CFF in the source code obfuscation space, and BPB in the packing obfuscation space are the most frequently used methods. Other obfuscation methods are rarely used. OA, CM, CFF, and BPB all affect the data distribution of the software, indicating that changes to the data distribution are the main reason for the decrease in LB-MDS accuracy.
- When attacking commercial antivirus software, OA, SP, SA, and SR in the binary obfuscation space are frequently used and mainly affect the program's data distribution, section padding, section name, and section hash features. This indicates that single obfuscation methods are no longer effective in reducing the accuracy of commercial antivirus software, and multiple obfuscation techniques are needed to modify multiple features of the program to decrease the accuracy of commercial antivirus software. Additionally, CM, CFF, JOI, and DE in the source code obfuscation space, and BPB, CV, BP, DL, SMC, CE, CC, and AO in the packing obfuscation space are frequently used, which supports the above conclusion that single obfuscation methods are insufficient.

6. Discussion

In this section, we will discuss methods to enhance the robustness of LB-MDS and analyze the current state of software obfuscation and malware detection. As previously demonstrated, commercial antivirus software exhibits poor performance in detecting packed malware due to encryption, which eliminates the feature and renders feature-based methods ineffective. However, packing can be utilized to protect intellectual property or important data, and simply labeling programs containing unpacking code as malware is not a practical solution. Packed malware necessitates decryption before regular execution, and the decrypted code and data of the malicious program are in plaintext in memory, enabling commercial antivirus software to detect malicious features using feature-based methods. Therefore, we suggest that antivirus software should primarily employ dynamic detection methods when determining whether a program is malicious or benign, as static detection cannot acquire the features of packed programs.

7. Conclusion

This paper presents a obfuscation dataset ERMDS that solves the problem of evaluating the robustness of LB-MDS. To evaluate the ability of the ERMDS obfuscation dataset, we used the obfuscation spaces to generate an instance of the obfuscation dataset called ERMDS-X. We then used this dataset to evaluate two LB-MDS models and six commercial antivirus softwares. Through experimentation, we found that ERMDS-X can reduce the accuracy of LB-MDS by an average of 20%, and reduce the accuracy of commercial antivirus software by an average of 32%. Finally, we analyzed the reasons for the decrease in accuracy for each LB-MDS and commercial antivirus software, and provided suggestions for improving robustness.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix

Parameters. The following parameters constitute a set that maximally reduces the accuracy of LB-MDS: $Num_b = 30$, $Num_s = 30$, $Num_p = 30$, $l_b = 30$, $r_b = 30$, $l_s = 10$, $r_s = 10$, $l_p = 25$, and $r_p = 25$. We recommend that Num_b, Num_s, and Num_p be set to at least 30 to ensure that each sample has multiple adversarial examples. $l_b = 30$ because

Table A.7
Functional testing of obfuscated malicious software.

Malware	*Functional Rate* (%)
malware1	91
malware2	95
malware3	93

over 30 rounds of binary obfuscation can minimize the accuracy of LB-MDS and commercial antivirus software AV1-AV6, as shown in Figs. 2 and 3. $r_b = 30$ because even if the number of binary obfuscation rounds is further increased, the accuracy of LB-MDS and commercial antivirus software will not decrease further after reaching 30 rounds. $l_s = 10$ because over 10 rounds of source code obfuscation can minimize the accuracy of LB-MDS and commercial antivirus software AV1-AV6, as shown in Fig. 4. $r_s = 10$ because even if the number of source code obfuscation rounds is further increased, the accuracy of LB-MDS and commercial antivirus software will not decrease further after reaching 10 rounds. $l_p = 25$ because over 25 rounds of packing obfuscation can minimize the accuracy of LB-MDS and commercial antivirus software AV1-AV6, as shown in Fig. 4. $r_b = 25$ because even if the number of packing obfuscation rounds is further increased, the accuracy of LB-MDS and commercial antivirus software will not decrease further after reaching 25 rounds.

Future Work. Comprehensively evaluating the robustness of MDS is a challenging task. In this paper, we primarily employ three types of obfuscation space to assess the performance of existing MDS under adversarial attacks. The study confirms that obfuscation techniques can be used to evaluate the robustness of MDS. However, the generation of adversarial samples is not limited to obfuscation techniques alone. For instance, in DeepMal [19], adversarial instructions were inserted into malware, allowing the generated adversarial samples to evade detection by CNN-based MDS. Such techniques can effectively capture the vulnerability of LB-MDS since small modifications to malware can deceive LB-MDS. Therefore, in future work, we will incorporate such adversarial attack techniques as an essential approach to generate more diverse samples in the EMBDR dataset and continuously enhance its richness.

Functional Integrity Testing This experiment aims to evaluate whether the combination of obfuscation techniques will compromise the functionality of malicious software. We randomly selected three malicious software programs with clear functionalities: Malware 1 encrypts files on the computer and extorts money, Malware 2 is a Trojan horse program client, and Malware 3 is a malicious advertisement plugin. Since we do not have access to the source code of these malicious software programs, we used obfuscation techniques from BOS and POS that were combined in various ways, with the number of combinations times between 3–15, to process these three malicious software programs. These obfuscation techniques were applied to each malicious software program to generate 100 different obfuscated versions. We then manually executed each obfuscated malicious software program to determine whether their functionalities had been altered. For example, we tested whether the obfuscated extortion software was still capable of encrypting files and extorting money. If the functionalities of these malicious software programs were not altered, it indicated that the combination of obfuscation techniques did not compromise the functionality of the programs.

From Table A.7, we can observe that even after undergoing various obfuscation techniques, most of the malicious software programs were still able to execute their original functionalities. Only 7% of the obfuscated malicious software programs lost their original functionalities after being processed. Analysis of the obfuscated malicious software programs that did not execute correctly led to the following conclusions: 1) To prevent tampering, Malware 1 calculated a checksum of certain parts of their code and checked whether the checksum was correct before execution. If our obfuscation method modified this part of the code, the malicious software would not execute because the checksum would have changed. 2) Malware 2 and 3 had a more complex PE format compared to Malware 1, with a large .rsrc resource section. Existing obfuscation tools such as Malfox could not correctly parse this section when processing Malware 2 and 3, causing it to not execute correctly.

References

[1] H.S. Anderson, P. Roth, EMBER: An open dataset for training static PE malware machine learning models, 2018, ArXiv e-prints arXiv:1804.04637.

[2] E. Raff, J. Barker, J. Sylvester, R. Brandon, B. Catanzaro, C.K. Nicholas, Malware detection by eating a whole EXE, 2017, ArXiv arXiv:1710.09435.

[3] G.E. Dahl, J.W. Stokes, L. Deng, D. Yu, Large-scale malware classification using random projections and neural networks, in: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 3422–3426, http://dx.doi.org/10.1109/ICASSP.2013.6638293.

[4] K. Rieck, P. Trinius, C. Willems, T. Holz_aff2n3, Automatic Analysis of Malware Behavior Using Machine Learning, 19 (4) (2011) 639–668.

[5] J. Saxe, K. Berlin, Deep neural network based malware detection using two dimensional binary program features, in: 2015 10th International Conference on Malicious and Unwanted Software, 2015, pp. 11–20, http://dx.doi.org/10.1109/MALWARE.2015.7413680.

[6] Avast 2018. AI & machine learning, 2018, https://www.avast.com/en-us/technology/aiand-machine-learning.

[7] M.D.A.R. Team., New machine learning model sifts through the good to unearth the bad in evasive malware, 2019, https://www.microsoft.com/security/blog/2019/07/25/new-machine-learning-model-sifts-through-the-good-to-unearth-the-bad-in-evasive-malware/.

[8] S.H. Ding, B.C. Fung, P. Charland, Asm2vec: Boosting static representation robustness for binary clone search against code obfuscation and compiler optimization, in: 2019 IEEE Symposium on Security and Privacy, SP, IEEE, 2019, pp. 472–489.

[9] X. Ren, M. Ho, J. Ming, Y. Lei, L. Li, Unleashing the hidden power of compiler optimization on binary code difference: An empirical study, in: Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation, 2021, pp. 142–157.

[10] W. Song, X. Li, S. Afroz, D. Garg, D. Kuznetsov, H. Yin, MAB-malware: A reinforcement learning framework for blackbox generation of adversarial malware, in: Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security, ASIA CCS '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 990–1003, http://dx.doi.org/10.1145/3488932.3497768.

[11] F. Zhong, X. Cheng, D. Yu, B. Gong, S. Song, J. Yu, MalFox: Camouflaged adversarial malware example generation based on C-GANs against black-box detectors, 2020, ArXiv, arXiv:2011.01509.

[12] A. Al-Dujaili, A. Huang, E. Hemberg, U.-M. OReilly, Adversarial Deep Learning for Robust Detection of Binary Encoded Malware, 2018, pp. 76–82, http://dx.doi.org/10.1109/SPW.2018.00020.

[13] H. Anderson, A. Kharkar, B. Filar, D. Evans, P. Roth, Learning to evade static PE machine learning malware models via reinforcement learning, 2018.

[14] R.L. Castro, C. Schmitt, G. Dreo, AIMED: Evolving malware with genetic programming to evade detection, in: 2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), 2019, pp. 240–247, http://dx.doi.org/10.1109/TrustCom/BigDataSE.2019.00040.

[15] L. Chen, Understanding the efficacy, reliability and resiliency of computer vision techniques for malware detection and future research directions, 2019, ArXiv, arXiv:1904.10504.

[16] L. Chen, Y. Ye, T. Bourlai, Adversarial machine learning in malware detection: Arms race between evasion attack and defense, in: 2017 European Intelligence and Security Informatics Conference, EISIC, 2017, pp. 99–106, http://dx.doi.org/10.1109/EISIC.2017.21.

[17] B. Kolosnjaji, A. Demontis, B. Biggio, D. Maiorca, G. Giacinto, C. Eckert, F. Roli, Adversarial malware binaries: Evading deep learning for malware detection in executables, in: 2018 26th European Signal Processing Conference, EUSIPCO, 2018, pp. 533–537, http://dx.doi.org/10.23919/EUSIPCO.2018.8553214.

[18] L. Jia, B. Tang, C. Wu, Z. Wang, Z. Jiang, Y. Lai, Y. Kang, N. Liu, J. Zhang, FuncFooler: A practical black-box attack against learning-based binary code similarity detection methods, 2022, http://dx.doi.org/10.48550/ARXIV.2208.14191, URL arXiv https://arxiv.org/abs/2208.14191.

[19] C. Yang, J. Xu, S. Liang, Y. Wu, Y. Wen, B. Zhang, D. Meng, DeepMal: maliciousness-preserving adversarial instruction learning against static malware detection, Cybersecurity 4 (2021) 16, http://dx.doi.org/10.1186/s42400-021-00079-5.

[20] R. Harang, E.M. Rudd, SOREL-20M: A large scale benchmark dataset for malicious PE detection, 2020, arXiv:2012.07634.

[21] L. Yang, A. Ciptadi, I. Laziuk, A. Ahmadzadeh, G. Wang, BODMAS: An open dataset for learning based temporal analysis of PE malware, in: 2021 IEEE Security and Privacy Workshops, SPW, 2021, pp. 78–84, http://dx.doi.org/10.1109/SPW53761.2021.00020.
[22] S. Schrittwieser, S. Katzenbeisser, J. Kinder, G. Merzdovnik, E. Weippl, Protecting software through obfuscation: Can it keep pace with progress in code analysis? ACM Comput. Surv. 49 (1) (2016) http://dx.doi.org/10.1145/2886012.
[23] C.S. Collberg, C.D. Thomborson, D. Low, Breaking abstractions and unstructuring data structures, in: Proceedings of the 1998 International Conference on Computer Languages (Cat. No.98CB36225), 1998, pp. 28–38.
[24] C. Nachenberg, Computer virus-antivirus coevolution, Commun. ACM 40 (1) (1997) 46–51, http://dx.doi.org/10.1145/242857.242869.
[25] L.M. Markus F.X.J. Oberhumer, J.F. Reiser, Ultimate packer for executables, 1996, https://upx.github.io/.
[26] O. Technologies, Themida overview, 2010, https://www.oreans.com/themida.php.
[27] S. King, P. Chen, SubVirt: implementing malware with virtual machines, in: 2006 IEEE Symposium on Security and Privacy (S&P'06), 2006, pp. 14–327, http://dx.doi.org/10.1109/SP.2006.38.
[28] P. Junod, J. Rinaldini, J. Wehrli, J. Michielin, Obfuscator-LLVM–software protection for the masses, in: 2015 IEEE/ACM 1st International Workshop on Software Protection, IEEE, 2015, pp. 3–9.
[29] M. Ollivier, S. Bardin, R. Bonichon, J.-Y. Marion, How to kill symbolic deobfuscation for free (or: Unleashing the potential of path-oriented protections), in: Proceedings of the 35th Annual Computer Security Applications Conference, ACSAC '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 177–189, http://dx.doi.org/10.1145/3359789.3359812.
[30] VirusShare, Virusshare, 2023, https://virusshare.com/.
[31] Quarkslab, LIEF: library for instrumenting executable files, 2017–2018, https://lief.quarkslab.com/.
[32] O. Technologies, An interactive decompiler, disassembler, debugger, 2015, https://binary.ninja/.
[33] Libre reversing framework for unix geeks, 2013, https://github.com/radareorg/radare2.
[34] A powerful disassembler and a versatile debugger, 2012, https://hex-rays.com/IDA-pro/.
[35] The LLVM compiler infrastructure, 2008, https://llvm.org/.
[36] A dynamic binary instrumentation tool, 2010, https://www.intel.com/content/www/us/en/developer/articles/tool/pin-a-dynamic-binary-instrumentation-tool.html.
[37] An open-source binary analysis platform, 2018, https://angr.io/.
[38] A professional app shielding and hardening solution, 2017, https://www.preemptive.com/.
[39] An free, open-source protector for net applications, 2015, https://mkaring.github.io/ConfuserEx/.
[40] A multifunctional EXE packing tool, 2010, http://www.aspack.com/asprotect32.html.
[41] A professional system for executable files licensing and protection, 2012, https://www.enigmaprotector.com/.
[42] A tool protect program from being reversed, 2010, https://shell.virbox.com/.
[43] VMProtect protects code by executing it on a virtual machine, 2012, https://vmpsoft.com/.
[44] D. Hendrycks, T. Dietterich, Benchmarking neural network robustness to common corruptions and perturbations, in: Proceedings of the International Conference on Learning Representations, 2019.
[45] Machine learning static evasion competition 2019, 2019, https://github.com/endgameinc/malware_evasion_competition.
[46] The best antivirus protection, 2020, https://www.pcmag.com/picks/thebest-antivirus-protection.

BenchCouncil Transactions on Benchmarks, Standards and Evaluations 3 (2023) 100108

Contents lists available at ScienceDirect

BenchCouncil Transactions on Benchmarks, Standards and Evaluations

journal homepage: www.keaipublishing.com/en/journals/benchcouncil-transactions-on-benchmarks-standards-and-evaluations/

Full length article

SNNBench: End-to-end AI-oriented spiking neural network benchmarking

Fei Tang *, Wanling Gao

Research Center for Advanced Computer Systems, State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences, China
University of the Chinese Academy of Sciences, China

ARTICLE INFO

Keywords:
Spiking Neural Networks
Artificial Intelligence
Deep learning
Benchmarking

ABSTRACT

Spiking Neural Networks (SNNs) show great potential for solving Artificial Intelligence (AI) applications. At the preliminary stage of SNNs, benchmarks are essential for evaluating and optimizing SNN algorithms, software, and hardware toward AI scenarios. However, a majority of SNN benchmarks focus on evaluating SNN for brain science, which has distinct neural network architectures and targets. Even though there have several benchmarks evaluating SNN for AI, they only focus on a single stage of training and inference or a processing fragment of a whole stage without accuracy information. Thus, the existing SNN benchmarks lack an end-to-end perspective that not only covers both training and inference but also provides a whole training process to a target accuracy level.

This paper presents SNNBench—the first end-to-end AI-oriented SNN benchmark covering the processing stages of training and inference and containing the accuracy information. Focusing on two typical AI applications: image classification and speech recognition, we provide nine workloads that consider the typical characteristics of SNN, i.e., the dynamics of spiking neurons, and AI, i.e., learning paradigms including supervised and unsupervised learning, learning rules like backpropagation, connection types like fully connected, and accuracy. The evaluations of SNNBench on both CPU and GPU show its effectiveness. The specifications, source code, and results will be publicly available from https://www.benchcouncil.org/SNNBench.

1. Introduction

Spiking neural networks (SNNs) have gained considerable attention as a novel technology under development and are considered the third generation of ANNs [1]. Compared to the second-generation—DNNs, SNNs are more closely aligned with biological neural networks and use spiking neurons as computational units. Thus, SNNs support processing time-series information naturally, without requiring additional structures like Recurrent Neural Networks (RNNs), indicating a huge potential for time-series tasks like speech and natural language processing. Moreover, unlike the DNNs that perform layer-by-layer computations, SNNs are driven by sparse spiking events and can achieve high parallelism through asynchronous computations. Overall, SNNs promise to achieve higher performance, lower power consumption, and stronger expression ability [2], making them a compelling option for a wide range of AI applications. At the preliminary stages of SNNs, benchmarks lay the foundation for exploring the design space of corresponding algorithms, systems, and architectures. However, existing SNN benchmarks cannot fulfill the benchmarking requirements of the AI scenarios considering the complexities of training and inference and the tradeoff between high performance and high model accuracy.

On the one hand, most SNN benchmarks mainly focus on brain science evaluation [3–5], which is a mainstream research direction that models and simulates computational neuroscience to understand the principles of the nervous system [6]. In contrast to the evaluation of SNNs in AI, brain science evaluation focuses on more accurate simulations of neural models. This includes capturing voltage variations over time and reproducing spike statistics with high precision, often employing the highly complex Hodgkin–Huxley neuron model [7]. In contrast, SNNs used for AI applications prioritize the ability to solve specific AI problems over accurately simulating the behavior of real neural models, including voltage variance and spike activity. As a result, these SNNs often rely on simple neural models such as the leaky integrate-and-fire (LIF) model [8] which, due to its simple structure, is easy to train while still retaining important spike features [9]. Hence, the benchmarks for brain science cannot suit the evaluation of SNNs for AI [10] since they do not consider the specific AI problems and have distinct neural network architectures and targets.

On the other hand, two benchmarks have been proposed to evaluate the SNNs for AI [11,12]. One benchmark from Ostrau et al. [11] focuses on the inference stage by converting a pre-trained DNN model to an SNN model and providing accuracy information. However, it does not consider the training phase or other learning rules. Although another benchmark from Kulkarni et al. [12] includes both the training and inference stages, it employs much simpler neural network architectures

* Corresponding author.
E-mail addresses: tangfei@ict.ac.cn (F. Tang), gaowanling@ict.ac.cn (W. Gao).

https://doi.org/10.1016/j.tbench.2023.100108
Received 2 March 2023; Received in revised form 25 April 2023; Accepted 8 May 2023
Available online 12 May 2023

Table 1
Implications from SNNBench.

Method	Learning Rule	Support Unsupervised Learning	Achieve State-of-the-Art Accuracy	Accuracy Loss	Easy to Use	Stable	Scalability	Well Mapped to GPUs
Train from scratch	STDP	Yes	No	Not applicable	No	No	No	No
	Surrogate Backprop	No	Yes	Not applicable	Medium	Yes	Yes	Yes
Converted from pre-trained models	DNN-to-SNN	Not applicable[a]	Yes	Low	Yes	Medium[b]	Partial[c]	Partial[c]

[a]Depends on original DNNs.
[b]Depends on conversion quality, influenced by factors like DNN architecture and conversion methods.
[c]Depends on original DNNs and conversion quality.

Table 2
SNNBench and other relevant SNN benchmarks.

		Kulkarni et al.	Ostrau et al.	SNNBench
Domain	Vision	✓	✓	✓
	Speech			✓
Learning paradigm	Supervised			✓
	Unsupervised			✓
Connection type	One-to-one			✓
	Fully connected	✓	✓	✓
	Convolutional			✓
	Recurrent			✓
Learning rule	STDP			✓
	Backpropagation	✓[a]		✓
	DNN-to-SNN		✓	✓
	Reservoir	✓[b]		
	Evolutionary	✓[b]		
Number of different spiking neurons		1	1	2
Inference		✓	✓	✓
Training to quality				✓
Open source		×	✓	✓
Number of workloads		5	5	9

[a]This benchmark mentioned the backpropagation-based learning rule while only provided the inference stage (forward pass).
[b]Partial training process without accuracy information.

compared to realistic ones and simulates only a partial training process. Consequently, it does not offer any accuracy information, which is a crucial metric for AI. Moreover, even with a long enough training process, these neural network architectures are not verified to achieve convergent accuracy. In this condition, they fail to evaluate the training and inference performance of SNN comprehensively, and further cannot answer these questions: (1) how to design systems and architectures for SNN that achieve both high performance and high accuracy? (2) whether to train an SNN model or convert one from a pre-trained DNN model? (3) how to choose different training strategies like supervised or unsupervised, recurrent connection or fully connection?

In this paper, we propose an end-to-end AI-oriented benchmarking methodology. Here end-to-end has two-fold meanings: end-to-end evaluation for a real-world AI problem that covers both training and inference stages; end-to-end training that considers diverse strategies and achieves a target accuracy. Based on the methodology, we propose SNNBench, the first end-to-end AI-oriented SNN benchmark. Focusing on two typical AI applications: image classification and speech recognition, SNNBench provides nine workloads that cover the representative characteristics of SNN and AI. Specifically, from the perspective of SNN, we consider the dynamics of spiking neurons. In terms of AI, we consider diverse training strategies, including learning paradigms, i.e., supervised and unsupervised learning, four typical connection types, i.e., one-to-one, fully connected, convolutional, and recurrent, and three widely-used learning rules, i.e., STDP, backpropagation, and DNN-to-SNN. Table 2 provides a comparison of SNNBench with the other two relevant benchmarks.

Our experiments show the effectiveness of SNNBench. Through the evaluations on both CPU and GPU, we have several observations as follows:

(1) The workload characterization on SNNBench shows its diversity and representativeness. The workloads within SNNBench cover ten groups of diverse operators and have different dominant ones. Moreover, the experiments show the good reproducibility of SNNBench.
(2) Different from the previous work [13], we find that using STDP learning rule (88%) is hard to achieve the state-of-the-art convergence accuracy (99.91%) compared to the backpropagation (98%) and conversion-based learning rules (96.72%). Moreover, the convergence accuracies using STDP have much larger fluctuations than the other two rules, with a standard deviation higher than 2.4%, while the value is below 0.3% for the other two. In terms of the training cost, to train the same number of images, using STDP occupies 73X longer time compared to using backpropagation and 1559X compared to using conversion.
(3) GPU is not always the best for SNN. In our experiments, we found that when the number of neurons in a layer of SNN is small, like 400, the CPU performs better than the GPU. This could be due to the small size of the SNN networks, which leads to short GPU computation times that cannot offset the synchronization overhead between the CPU and GPU, or the software framework used for simulating SNNs may not be optimally designed for exploiting the full potential of GPUs. For recurrent networks, the training time on GPU using LIF neurons is 1.37 times that of on CPU. Using LSNN neurons, the gap is 1.22 times. In future work, we plan to explore larger SNN networks and further optimization of both the software framework and the mapping of SNN workloads to the GPU hardware.
(4) Even though SNNs have great potential for asynchronous parallelism, the corresponding hardware, software systems, and SNN network architectures fail to exploit this advantage and thus face poor inter-operator and intra-operator scalability currently.

Based on experiments from SNNBench, we present some insights from SNNBench in Table 1. Surrogate backpropagation and conversion-based methods are recommended, as they can achieve comparable accuracy to DNNs and require minimal modifications to existing DNNs. However, using surrogate backpropagation necessitates choosing suitable smooth functions and loss functions, which may require some professional expertise. There are existing conversion tools to convert DNNs to SNNs, so one may not even need to modify the existing DNNs, but the conversion quality can be affected by various factors. If only unlabeled data is available, STDP is the only choice, as it supports unsupervised learning, but it suffers from instability issues. We also find that the surrogate backpropagation method can utilize GPUs, while STDP is not well-mapped to GPUs.

We organize the rest of the paper as follows. Section 2 explains the related work. Section 3 illustrates the design methodology and implementation of SNNBench. Section 4 shows the experiments. Finally, we draw a conclusion in Section 5.

2. Related work

Several benchmarks have been proposed to study computational neuroscience, which employs mathematical models and computer simulations to understand how electrical and chemical signals process and represent information in the brain [6]. Brette et al. [3] simulated a network containing 4000 neurons, 80% of which were excitatory and 20% were inhibitory neurons, randomly connected with a probability of 2%. They proposed four benchmarks, each with the same network architecture but different combinations of spiking neurons and synaptic types, and provided simulation specifications that include

Hodgkin–Huxley (HH) and integrate-and-fire (IF) neuron models, as well as current-based and conductance-based synaptic types. These simulations were implemented using different simulators. Tikidji-Hamburyan et al. [4] simulated two networks, called Classical Pyramidal InterNeuron Gamma (PING) [14] and PostinhIbitory Rebound-InterNeuron Gamma (PIR-ING) [15]. These two networks are implemented using LIF and HH neurons, respectively. Van Albada et al. [5] modeled a network under one mm^2 of the surface of generic early sensory cortex, organized into multiple layers, including 77,169 neurons connected via approximately 3×10^8 synapses, which is a huge network for simulation for that time. The network architectures in these benchmarks are biologically realistic; they are not directly applicable to SNNs for AI, as spiking neuron models used for ANNs are highly abstract and only include basic features of spiking neurons, such as spike trains, thresholds, and spike firing. Thus, these computational neuroscience benchmarks are unsuitable for evaluating SNNs for AI.

There are also some benchmarks for AI tasks. Kulkarni et al. [12] selected five workloads to evaluate the performance of simulators and claimed that the benchmark could represent computer science and machine learning workloads instead of computational neuroscience. However, whether the network architecture used in the workload can achieve reasonable accuracy on real-world tasks has not been validated, which fails to reflect the state-of-the-art or state-of-the-practice works. Additionally, it only simulates the training or inference process, adopts indirect metrics such as operations per second, and ignores accuracy. Ostrau et al. [11] uses a converted SNN model from DNNs to measure the performance of neuromorphic hardware. It completely neglects the training process on neuromorphic hardware. And converting DNNs to SNNs is only one method of using SNNs, lacking the representativeness of different learning rules. These two benchmarks only cover a few aspects of SNNs oriented toward AI. This paper proposes a comprehensive and representative SNN benchmarking methodology—SNNBench. Table 2 lists these two benchmarks and SNNBench.

3. SNNBench design and implementation

In this section, we first introduce the requirements of SNN benchmarks and then illustrate the SNNBench methodology. Finally, we present the implementation of SNNBench in detail.

3.1. The requirements for SNN benchmarks

The existing SNN benchmarks either focus on brain science benchmarking instead of the ability to solve AI problems or only cover a partial evaluation of these abilities. Hence, we aim to benchmark the SNNs for real-world applications like artificial intelligence. To achieve this goal, the SNN benchmark needs to satisfy the following requirements.

(1) Covering representative real-world applications. A benchmark should have relevance to its target domain [16]. Thus, we should choose representative tasks and datasets for evaluation.
(2) Covering the typical characteristics of SNN. The SNN benchmark should consider the dynamics of spiking neurons, which contain the change of the membrane potential and firing spikes. Meanwhile, considering the benchmark is AI-oriented, suitable spiking neuron models should be selected.
(3) Covering the typical characteristics of deep learning. On the one hand, the benchmark should consider different learning approaches, like supervised, unsupervised, semi-supervised, and reinforcement learning. On the other hand, the benchmark should contain both training and inference phases. Important factors should be considered for different phases, like the diverse learning rules, spiking neurons, connection types, etc.
(4) Meeting the reproducibility and usability requirements of benchmarking. Reproducibility is of great significance for both benchmarking and deep learning communities. For benchmarking, good reproducibility guarantees the fairness and consistency of the evaluation results. For deep learning, considering its stochastic intrinsic [17], good reproducibility assures the stability of the performance and accuracy. Usability is another requirement for benchmarking. The benchmarks should be simple and have affordable evaluation costs [18].

3.2. SNNBench methodology

The methodology underpinning SNNBench is illustrated in Fig. 1. SNNBench represents an end-to-end benchmarking methodology in two distinct aspects. Firstly, it encompasses both training and inference, providing a comprehensive solution for AI tasks. Secondly, it trains the model to the target accuracy (to a convergent state) instead of simply mimicking a limited number of training iterations, which would be insufficient as accuracy can only be determined upon reaching the convergent state.

SNNBench selects representative AI tasks and datasets from vision, speech, and natural language processing (NLP) domains. Two representative tasks – image classification and speech recognition – and their corresponding datasets are chosen. Image data serves as an exemplar of static data, while speech data exemplifies temporal data.

In addition, SNNBench takes into account the characteristics of both SNNs and deep learning, including training paradigms, connection types, spiking neurons, and learning rules. Training paradigms encompass supervised and unsupervised learning. While supervised learning is widely employed in deep learning, unsupervised learning is less common in recognition tasks. However, due to learning rules such as STDP in SNNs, unsupervised learning can be applied to recognition tasks. Consequently, it is crucial to consider training paradigms.

SNNs exhibit various neuron organization types, and we select representative connection types from [13], as well as other commonly used connection types in deep learning. We exclude connection types that employ reservoir computing and evolutionary optimization learning rules (c and d in Fig. 2 in [19]) because these rules are infrequently used, and there is a promising trend toward combining deep learning and SNNs [20–23].

Moreover, SNNBench should account for different spiking neurons. Ultimately, SNNBench primarily includes learning rules such as STDP learning, surrogate backpropagation, and DNN-to-SNN conversion, while excluding reservoir computing and evolutionary optimization for the same reasons stated earlier regarding the choice of connection types.

SNNBench also addresses benchmarking requirements of usability and reproducibility. Further details on the methodology will be provided in this section.

3.2.1. Considering an end-to-end solution for AI tasks

The basic paradigm of AI is to train a model using the training dataset first and then make inferences on the test dataset. It is well-known that training and inference are different. The most significant difference is that training involves weight updating and even the evolution of network architectures, such as neural architecture search [24] and evolutionary optimization [25]. On the other hand, weights and architecture remain unchanged during inference. Hence, they have different workload characteristics, as verified in Section 4.2. As a direct result, hardware architectures designed for training and inference differ, like various DNN accelerators. If one only considers one of the phases, it may mislead the hardware design. Therefore, it is not sufficient to only consider training or inference.

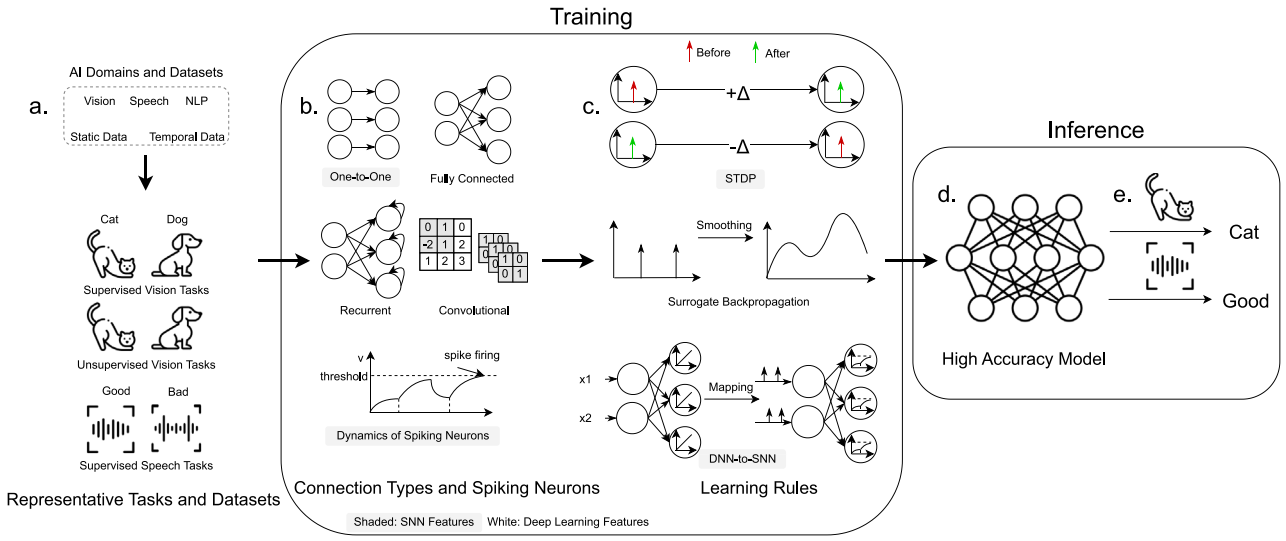


Fig. 1. SNNBench methodology. SNNBench is an end-to-end benchmarking methodology designed to cover both training and inference phases for SNNs while training the model to the target accuracy. (a) SNNBench selects representative tasks and datasets from vision, speech, and NLP domains, as well as static (image) and temporal (speech) data (upper part of (a)), while considering different training paradigms—supervised and unsupervised learning (bottom part of (a)). (b) A variety of network architectures can be built by combining different connection types, including one-to-one, fully connected, recurrent, and convolutional (upper and middle parts of (b)), and spiking neurons (bottom part of (b), where the potential dynamics of LIF neurons are used to represent spiking neurons). (c) SNNBench also takes into account different learning rules, such as STDP (upper part of (c), where connections are strengthened if pre-synaptic spikes occur before post-synaptic spikes, and vice versa), surrogate backpropagation (middle part of (c)), and DNN-to-SNN conversion (bottom part of (c)). (d) Once trained to the target accuracy (and converted to an SNN model if necessary), a high-accuracy SNN model is obtained. (e) The SNN model can then be employed to perform inference tasks on test data.

3.2.2. Training the model to the target accuracy

Many benchmarks use indirect metrics like operations per second because they are easy to measure. However, these metrics may not reflect whether a hardware system can solve real AI tasks. Different design strategies like float point precision can lead to non-objective assessments. For example, a hardware system that achieves high operations per second may not be able to train to the target accuracy if it adopts a low float point precision implementation. To address this issue, SNNBench trains the model to the target accuracy and uses accuracy as an important metric.

3.2.3. Covering representative real-world applications

SNNBench is designed for AI applications and focuses on image classification and speech recognition as benchmarking tasks. These two tasks are widely used in deep learning and serve as representative benchmarks. The benchmark suite includes state-of-the-art, state-of-the-practice, classical spiking neural architectures that are widely accepted and highly cited in SNN research. For the image classification task, SNNBench uses the Modified National Institute of Standards and Technology (MNIST) handwritten digits database [26]. For the speech recognition task, SNNBench uses the Speech Commands v2 dataset [27], which contains 105,829 audio files of spoken words and is widely used for simple speech recognition tasks.

3.2.4. Covering the typical characteristics of SNN

The SNNBench benchmark suite is designed to represent the typical characteristics of SNNs. SNNs use spike timing as the input and encode it as a series of 0 s and 1 s that indicate whether a post-synaptic neuron has received a spike from the pre-synaptic neuron at a given time. Upon receiving a spike, the membrane potential of the post-synaptic neuron increases. If it reaches a threshold, the neuron fires a spike, resetting the potential to the resting potential. The post-synaptic neuron is unresponsive to incoming spikes during the subsequent refractory period.

Many different neural models range from simple LIF models to complex Hodgkin–Huxley models. While more complex models are more accurate in simulating the neurons in the brain, they also have more parameters, which can make them difficult to train in a neural network. As a result, LIF-based models are widely used in the AI field due to their simplicity. These models have been implemented in various neuromorphic architectures, such as IBM's TrueNorth [28] and Intel's Loihi [29]. Complex models like the Hodgkin–Huxley model contain many components suitable for studying neural dynamics but are too complex to implement in cognitive neuromorphic architectures.

Therefore, SNNBench focuses on LIF-based spiking neurons and provides different LIF-based models for comparison in the speech recognition task.

3.2.5. Covering the typical characteristics of deep learning

SNNBench is a benchmark suite designed specifically for AI applications and, therefore, must capture the essence of deep learning. It covers multiple facets of AI, including various phases of learning, learning types, learning rules, and connection types. Unlike previous works that only consider the inference phase, SNNBench includes both the training and inference phases. The four primary types of AI learning are supervised, unsupervised, semi-supervised, and reinforcement learning. SNNBench supports supervised and unsupervised learning and plans to incorporate reinforcement learning in a future release. Supervised learning is the dominant paradigm in artificial intelligence, but its reliance on manual labeling of data presents a challenge in terms of cost and scalability. On the other hand, unsupervised learning does not require labeled data and more closely resembles how the brain learns.

Three main learning rules are used to train SNNs: spike-timing–dependent plasticity (STDP), surrogate backpropagation, and DNN-to-SNN conversion. The STDP learning rule is based on the biological principle of spike-timing–dependent plasticity [13] and is most similar to the way the brain learns. Backpropagation, which has been widely used in traditional ANNs and has produced remarkable results in fields such as computer vision, natural language processing, and robotics, cannot be directly applied to SNNs due to the discrete and non-differentiable nature of spikes. However, some variants of backpropagation, such as surrogate backpropagation, have been proposed and have achieved performance close to the state-of-the-art on some datasets [19]. The DNN-to-SNN conversion method involves mapping the weights of an DNN to spike firing probabilities. Common techniques to mitigate accuracy loss include using the ReLU [30] activation function, fixing the bias to zero during training, weight normalization, and

Table 3
Workloads from SNNBench. Except for Image-Conversion, which only includes inference workloads, all other workloads consist of both training and inference workloads, resulting in a total of 9 workloads.

Benchmark	Task	Dataset	Network layout	Learning paradigm	Spiking neuron	Connection type	Learning rule	Spike encoding	Accuracy	Note
Image-STDP	Image classification	MNIST	One input layer, one excitatory layer and one inhibitory layer	Unsupervised	LIF	Fully connected, One-to-one	STDP	Rate encoding	95%	STDP learning strategy is more similar as the brain works, and this task is the representative of the unsupervised task
Image-Backprop	Image classification	MNIST	Two convolutional layers and one fully connected layer	Supervised	LIF	Convolutional	Surrogate Backpropagation	Rate encoding	95%	Using surrogate backpropagation is another common way to use SNN
Image-Conversion	Image classification	MNIST	Three fully connected layers	Supervised	LIF	Convolutional	Backpropagation	Rate encoding	95%	One commonly method to use SNN
Speech-LIF	Speech recognition	Google Speech Commands v2	One input layer, one recurrent layer and one readout layer	Supervised	LIF	Recurrent	Surrogate Backpropagation	Rate encoding	90%	Recurrent SNN with the LIF neuron
Speech-LSNN	Speech recognition	Google Speech Commands v2	One input layer, one recurrent layer and one readout layer	Supervised	LSNN	Recurrent	Surrogate Backpropagation	Rate encoding	90%	Recurrent SNN with the LSNN neuron

threshold tuning [31,32]. This conversion-based approach has achieved competitive performance compared to traditional DNNs and is widely used. SNNBench includes workloads with all three learning rules.

In the brain, neurons with similar functions are organized into a neural population group. Similarly, in deep learning, neurons are organized layer-by-layer into distinct connection types, such as fully connected, convolutional, and recurrent layers. SNNBench provides workloads with different connection types to take advantage of the mature and efficient connection types developed in deep learning.

3.2.6. Meeting the reproducibility and usability requirements of benchmarking

To ensure the reproducibility and usability of our benchmarks, we have taken several steps to address the intrinsic stochastic nature of AI. Firstly, AI algorithms typically rely on randomness, such as choosing a random initial state for training and shuffling the input data, to enhance their robustness. However, these methods can also make it challenging to reproduce benchmark results. Secondly, the operations used in deep learning algorithms, such as convolution and matrix multiplication, have various implementations, which can further increase the volatility of benchmark results [33]. To mitigate this, we have set the same random seed for all benchmarks, including PyTorch, Python, and NumPy random libraries, ensuring that the initial states and input order remain consistent with each run. We have also disabled the cuDNN benchmarking feature that selects the most efficient convolution implementation in time. Instead, we have made PyTorch choose a deterministic algorithm for all operations, ensuring that the same algorithms and implementations are used for each run. Additionally, to improve usability, we have used Docker to set up a consistent experiment environment for each run of the benchmark.

3.3. SNNBench implementation

Table 3 outlines the workloads from SNNBench, which includes image classification and speech recognition tasks. The MNIST dataset is utilized for the image classification task, while the Google Speech Commands v2 dataset is used for the speech recognition task, both of which are described in detail in Section 3.2.3. Except for Image-Conversion, which only contains inference workload, all the benchmarks contain both training and inference workloads. In this subsection, we delve into the implementation of SNNBench.

3.3.1. Image-STDP

This benchmark involves an image classification task that trains an SNN network using the STDP learning rule. We have selected the most classic and widely cited STDP learning rule [13], which is representative of unsupervised learning tasks and has significantly impacted subsequent research. Our implementation is based on the BindsNet framework [34].

The network architecture consists of input, excitatory, and inhibitory layers, and the excitatory and inhibitory layer contain the same number of neurons. The input layer contains 28×28 neurons, corresponding to the MNIST dataset's image size. The excitatory and inhibitory layers simulate excitatory and inhibitory neurons, respectively.

The connection between the input layer and the excitatory layer is a fully connected one, while the connection between the excitatory and inhibitory layers follows a one-to-one map-style pattern, with each inhibitory neuron connecting to all excitatory neurons except the one it is connected to. The input layer accepts 28×28 spike trains generated using a Poisson distribution, where the parameter α is proportional to the corresponding pixel's intensity.

Workload #1: Image-STDP Training workload uses the STDP learning rule to update the synaptic weights, where stronger connections are formed if pre-synaptic neurons consistently lead to post-synaptic neurons firing, and weaker connections are formed if the opposite is true. This reflects the impulsiveness of the brain, where some neurons enhance the reaction while others prevent it. The excitatory and inhibitory neurons both have LIF behavior but opposite weight-updating strategies. We check the accuracy of every iteration and train the model to the convergent state, and check if it reaches the target accuracy.

Workload #2: Image-STDP Inference workload uses the trained model to infer on the test dataset. After training, the label of the image that elicits the most firing in a neuron is assigned to that neuron. During inference, the label of the image is assigned based on the most fired neurons. Counting the spiking distributions can get the inference result.

We study the STDP learning rule in different scales by implementing SNNs with 100, 400, 1600, and 6400 neurons in the excitatory and inhibitory layers.

3.3.2. Image-Backprop

This benchmark is also an image classification task but uses surrogate backpropagation. Backpropagation is the basis of the success of deep learning, and large models and big-volume data can be trained using this learning rule. However, backpropagation cannot be directly applied to SNNs because of the non-differentiable spikes. Even though the STDP learning rule can be used to train SNNs, the training has many problems that need to be solved; for example, the neurons close to the output layer rarely fire in deep SNNs so that SNNs' networks cannot be constructed as deep as current deep neural networks. Since the lack of efficient learning rules, many researchers have focused on training SNNs through backpropagation using a workaround approach. Surrogate gradient descent is one of the popular methods. It replaces the non-differentiable part with a differentiable function, for example, using an approximate function to replace the derivative of the spike to the membrane potential. SuperSpkie [35] is one of the representatives of this method. It uses the fast sigmoid function to approximate the Dirichlet function of firing spikes so that the gradient can be calculated in the backward pass. And it does not modify the forward pass, so the

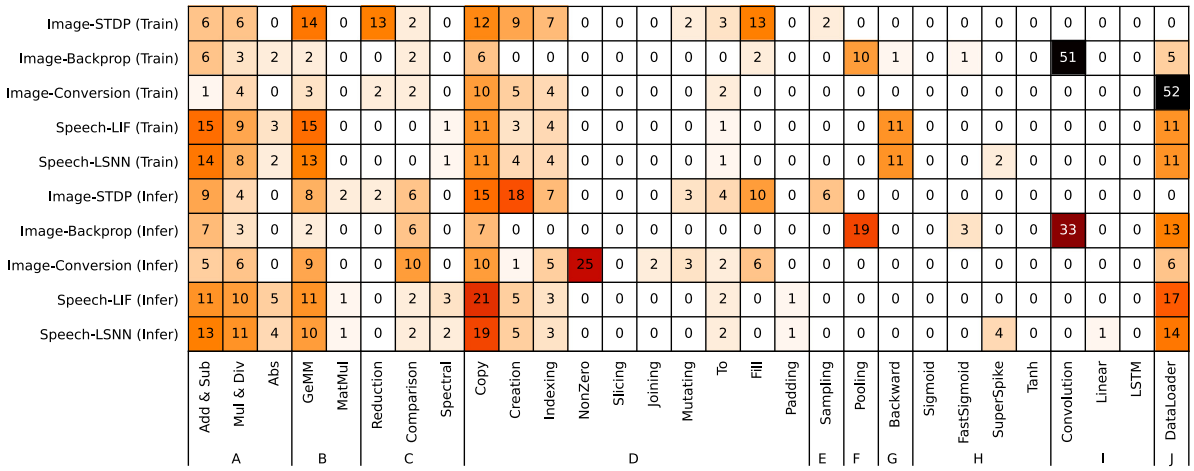


Fig. 2. Operator heatmap of SNNBench. All operators are divided into ten groups: Pointwise (A), BLAS (B), Math (C), Tensor-Related (including Creation, Indexing, Slicing, Joining, and Mutating) (D), Sampling (E), Pooling (F), Backward (G), Activation (H), Layer-Computing (I), and Data-Loading (J), as depicted in the bottom part of the figure. The left part displays individual workloads, with "(Train)" indicating a training workload and "(Infer)" indicating an inference workload. The time cost of each operator is counted, and the numbers in the grid represent the percentage of time consumed by each operator for each workload. To enhance visibility, a deeper color is used to indicate a larger percentage.

network architecture is similar to DNN, except it has different computing units. We adopt an implementation that is a simple convolution network containing two convolution and max pooling layers and a fully connected layer as the output layer. All the computing units are LIF neurons.

Workload #3: Image-Backprop Training workload uses the fast sigmoid to smooth the spiking train and trains the neural work using backpropagation.

Workload #4: Image-Backprop Inference workload does not involve the smoothing progress used in the training workload, and it directly computes the result through the forward pass.

3.3.3. Image-Conversion

This benchmark is an image classification task that uses the DNN-to-SNN conversion method, which is one of the strategies to overcome the challenge of training SNNs. The conversion-based method trains a DNN model first and then maps the trained model to an SNN model. The common approach to mapping deep neural networks (DNNs) to spiking neural networks (SNNs) is to replace real values in the DNNs with spiking frequencies and to replace activation functions with spiking neurons. In the training phase, Diehl et al. [32] use ReLUs as activation functions for all network units and eliminate biases. The ReLU function ensures that the values are non-negative, which SNNs cannot represent, while the biases are fixed at zero. To further improve the performance of the SNN, Diehl et al. use two weight normalization methods: model-based normalization and data-based normalization. Model-based normalization normalizes weights based on the trained model's weights, while data-based normalization normalizes weights based on the training data. In this workload, a multi-layer perception network with three layers is used, and each layer is followed by a ReLU activation function except for the output layer. The DNN model is trained to convergence, then converted to an SNN model and normalized using data-based normalization. The converted SNN model is then used for inference on the test dataset. This conversion-based method is a popular approach to overcoming the challenges of training SNNs and has attracted much attention from researchers.

Workload #5: Image-Conversion Inference workload is performed after converting the trained DNN model to the SNN model. We also check the accuracy drop after converting.

3.3.4. Speech-LIF and Speech-LSNN

We present two benchmarks that use the same network architecture and learning rule but with different spiking neurons to assess the impact on performance. Both workloads utilize the surrogate backpropagation method and LIF-based neurons. One workload uses traditional LIF neurons, while the other uses the LSNN neuron, a LIF variant with adaptive thresholds that are dynamic during training but fixed during inference. Additionally, we evaluate a workload that uses standard activation functions as computing units, serving as the baseline for deep neural network (DNN) performance.

Workload #6: Speech-LIF Training workload uses the recurrent neural network, is built with LIF neurons, and uses the surrogate backpropagation learning rule.

Workload #7: Speech-LIF Inference workload uses the same neural architecture as the training workload and infers on the test dataset.

Workload #8: Speech-LSNN Training workload is the same as Speech-LIF Training workload, but is built with LSNN neurons.

Workload #9: Speech-LSNN Inference workload is the same as Speech-LIF Inference workload, but is built with LSNN neurons.

4. Experiment

In this section, we conduct a series of experiments to show the effectiveness of SNNBench. First, we perform workload characterization on SNNBench in Section 4.2 and show the reproducibility of SNNBench in Section 4.3. Then, we compare the impact of learning rules and computing units on the SNN training and inference performance in Section 4.4 and Section 4.5, respectively. Finally, we evaluate the scalability of SNNBench in Section 4.6.

4.1. Experiment setup

We deploy SNNBench on a server node equipped with two Intel Xeon E5-2620 v3 @ 2.40 GHz CPUs and four Nvidia GeForce RTX 2080-Ti GPU cards. Each CPU contains six physical cores and enables hyper-threading. The software versions are CUDA toolkit 10.2, Python 3.10 and Pytorch 1.12. We have mainly investigated three AI-oriented SNN frameworks: BindsNet, snnTorch, and Norse. Due to the distinct APIs and functions provided by these frameworks, we have adopted the following strategy for simplicity and convenience: We utilized BindsNet 0.3.1 for both STDP and DNN-to-SNN methods in the image classification task, employed snnTorch 0.5.3 for surrogate backpropagation in the image classification task, and used Norse 0.0.7 for all speech recognition workloads. To make the experimental environment consistent for each run and avoid the performance drop due to default security settings, we use Docker to build the environment and disable Docker's seccomp security option.

Table 4
Variations in loss and accuracy after each epoch during the training process for speech recognition tasks. This table focuses exclusively on speech recognition tasks, as these metrics remain consistent across each run for image classification tasks. The differences among three runs after each epoch are highlighted in bold, illustrating that the discrepancies are minimal.

Epoch	Speech-LIF						Speech-LSNN					
	CPU			GPU			CPU			GPU		
0	3.54**58** (4%)	3.54**77** (3%)	3.59**45** (4%)	3.59**63** (2%)	3.52**07** (3%)	3.56**59** (4%)	2.70**21** (4%)	2.71**95** (3%)	2.72**90** (4%)	2.72**19** (4%)	2.71**14** (3%)	2.72**10** (4%)
1	1.36**90** (63%)	1.36**98** (63%)	1.36**93** (63%)	1.36**63** (63%)	1.35**87** (63%)	1.36**14** (63%)	1.91**93** (62%)	1.92**06** (62%)	1.91**76** (62%)	1.91**89** (62%)	1.91**82** (62%)	1.91**68** (62%)
2	1.31**85** (63%)	1.30**76** (63%)	1.30**36** (63%)	1.30**72** (63%)	1.30**03** (63%)	1.30**61** (63%)	1.85**03** (62%)	1.85**40** (62%)	1.85**23** (62%)	1.85**04** (62%)	1.85**04** (62%)	1.84**97** (62%)
3	1.27**94** (63%)	1.27**28** (63%)	1.27**09** (63%)	1.26**79** (63%)	1.26**82** (63%)	1.27**11** (63%)	1.80**77** (62%)	1.81**00** (62%)	1.80**95** (62%)	1.80**63** (63%)	1.80**81** (63%)	1.81**18** (62%)
4	1.25**00** (63%)	1.24**66** (63%)	1.24**86** (63%)	1.24**60** (63%)	1.24**59** (63%)	1.24**66** (63%)	1.78**31** (63%)	1.78**98** (62%)	1.79**11** (63%)	1.78**47** (63%)	1.78**90** (63%)	1.79**31** (63%)
5	1.22**98** (63%)	1.22**04** (63%)	1.21**98** (63%)	1.21**82** (64%)	1.22**28** (63%)	1.22**38** (63%)	1.76**25** (63%)	1.76**96** (62%)	1.77**08** (63%)	1.76**37** (63%)	1.77**13** (63%)	1.77**15** (63%)

We use PyTorch Profiler [36] to collect the runtime information for all the experiments, including the involved operators, the input size of each operator, and the time consumption. We report the profile data on the CPU for simplicity and veracity since a mass of operations like memory synchronization on GPUs would interfere with the analysis. We run each benchmark ten times and report the average values, each containing a two-batch warm-up stage (see Fig. 2).

4.2. Workload characterization

In this experiment, we conduct a top-down analysis for each workload. Considering that training and inference are the two most consuming parts, we only profile these two phases and exclude other phases like model initialization.

Fig. 2 shows the operator heat map of SNNBench workloads. We classify these operators into ten groups, and they are Pointwise (A), BLAS (B), Math (C), Tensor-Related (Creation, Indexing, Slicing, Joining, and Mutating) (D), Sampling (E), Pooling (F), Backward (G), Activation (H), Layer-Computing (I), and Data-Loading (J). From the result, we find that for most workloads, the Pointwise, BLAS, Tensor-Related (especially copy, creation, and indexing), and Data Loading operators consume a lot of time. The framework uses tensor as the basic data structure and implements many operators based on BLAS libraries so that Pointwise and BLAS operations consume most reasonably. However, Image-Backprop (Train), Image-Backprop (Infer), and Image-Conversion (Train) consume little time on BLAS operations. This is because the Image-Backprop use convolution connections, so the convolution operator occupies a lot of time while the BLAS operation takes up very little. As for Image-Conversion (Train), it uses only three fully-connected layers, and the input sizes are small, thus, the computing process is fast, and the general matrix multiplication (GeMM) operator occupies relatively much less compared to data loading and copy operators. When the data loading time ratio decreases, the GeMM time ratio correspondingly increases a lot (from 2% to 9% in Image-Conversion (Infer)). From the perspective of each operator category, for Pointwise operators, add, sub, mul, and div operators consume the most time. And as for BLAS operators, GeMM takes up almost all the time while matmul barely does. For Tensor Related operators, tensor copy, creation, and indexing occupy the most time. From the perspective of workloads, Image-STDP does not spend much time in data loading and uses sampling operators to construct the inputs; these characteristics are different from other workloads. Image-Backprop is more like a traditional convolution neural network, except that it uses the LIF neuron as the computing unit, so it spends much time in convolution and pooling operators. Image-Conversion trains an DNN model and converts the well-trained DNN model to an SNN model, and uses the SNN model to infer, so its training process is exactly the same as the traditional fully connected networks. In the inference phase, it spends much more time in the nonzero operator than other directly trained SNN models. The cause is that the converted SNN model has high spike firing rate than other directly trained SNN models. Thus the converted model has more non-zero values in a tensor; in other words, data have higher density. And the more dense the data, the more time the nonzero counting operator spends. So the nonzero operator occupies high. We did a small validation experiment for a size of (10000, 64) tensor using random initialization and zeros initialization, performing the nonzero operator on these two tensors costs 603.57 us and 56.32 us, respectively. The performance gap is one order of magnitude. This also implies that the impact of different input characteristics on performance is enormous. The workload characteristics are almost the same for the two speech recognition tasks because they use similar learning rules and are only different in computing units. They differ from image classification tasks in that they need to process speech data so that they contain the spectral operator (FFT transformation in this case). The backward operator also occupies a high total time that image classification tasks do not.

In this experiment, we perform a top-down analysis for each workload. Our findings reveal that the majority of workloads allocate considerable time to Pointwise, BLAS, Tensor-Related, and Data Loading operators. Image-Backprop and Image-Conversion workloads display distinct characteristics compared to others, such as reduced BLAS operation time and elevated spike firing rate in the converted SNN model, impacting performance. The workload characteristics remain consistent between the two speech recognition tasks, as they employ similar learning rules and vary only in computing units. Additionally, they incorporate the spectral operator, which is absent in image classification tasks.

4.3. Reproducibility

SNNBench applies several strategies to eliminate the randomness mentioned in Section 3.2.6. In this subsection, we evaluate the reproducibility of SNNBench by running each benchmark multiple times on the same hardware and software systems.

Our experiments show that the image classification task obtains exactly the same results for different runs. The deviations for the speech recognition task are very slight. Table 4 shows the changes in loss and accuracy after each epoch during the training process for the speech recognition task. Note that we do not list the results of the three image classification workloads because their changes in loss and accuracy are exactly the same. The results show that the stochasticity of SNNBench is small enough to meet the benchmarking requirement for reproducibility.

4.4. Comparison of different learning rules

In this subsection, we use the three image classification workloads as an example to compare different learning rules.

4.4.1. Accuracy comparison

Fig. 3(a) demonstrates the training progress of the three image classification workloads with different learning rules and the upper part of Table 5 lists the convergent accuracy and time to achieve that accuracy. We trained MNIST for 10 epochs on each workload to ensure that all of them reached a convergent state and achieved the respective accuracy range for each learning rule, thereby ensuring convergence for all workloads. For Image-Backprop and Image-Conversion workloads, the accuracy reached approximately 98%, with no further improvement. For the Image-STDP workloads, we checked the accuracy and updated the weights every 250 samples during 10 epochs, resulting in a total of 2400 accuracy checks and weight updates (only 50 of which are presented in Fig. 3(a) for clarity). This training process took almost

Table 5
Accuracy, epochs to convergent accuracy, and time to achieve accuracy on testsets.

	Accuracy on testset	Epochs to accuracy	Time to accuracy on CPU (s)	Time to accuracy on GPU (s)
Image-STDP (100)	75%	1	25858.14	34812.06
Image-STDP (400)	83%	1	32379.48	31739.34
Image-STDP (1600)	88%	4	363866.4	137905.2
Image-STDP (6400)	84%	6	12248568	355468.32
Image-Backprop	**98.59%**	3	1060.43	221.36
Image-Conversion	97.4%	5	**82.91**	**58.23**
Speech-LIF	67%	40	22773.35	31191.47
Speech-LSNN	63%	1	**679.20**	**828.51**
Speech-LSTM	**90%**	43	19908.03	8201.03

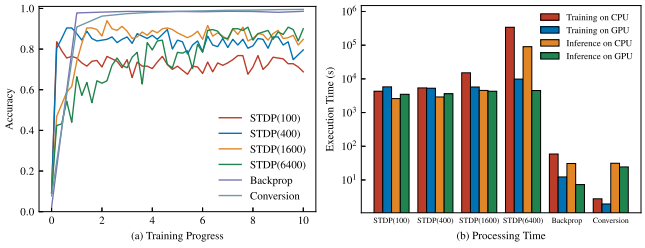


Fig. 3. Changes of accuracy and execution time using three different learning rules on the image classification task. For the Image-STDP workload, the accuracy is presented five times per epoch, and for other workloads, once per epoch. The numbers inside the brackets after STDP indicate the STDP workloads for different network sizes, which correspond to the number of spiking neurons in the excitatory and inhibitory layers. The time in the figure is to process the train (10000 images) and test (10000 images) dataset and has been processed using logarithms.

a week to complete on a 2080 Ti GPU for the network of 6400 neurons (6.86 days). In conclusion, the accuracy of all workloads remained stable, with no significant improvement observed over time, indicating that they reached a convergent state.

For the Image-STDP workload, the origin paper [13] achieved 82.9%, 87.0%, 91.9%, 95.0% accuracy when the number of excitatory and inhibitory neurons are set to 100, 400, 1600, and 6400, respectively. We achieve 75%, 83%, 88%, and 84% accuracy after convergence and get 84.8%, 93.2%, 96.0%, and 94.8% best accuracy during the training using 100, 400, 1600, and 6400 neurons, respectively. The number in the bracket is the number of spiking neurons in the excitatory and inhibitory layers. We can see that the higher the number of spiking neurons, the more epochs required to the convergent state, and the higher accuracy relatively, which is intuitive. When the number of neurons used is below 400, the accuracy reaches a relatively stable state after only one epoch. When the number of neurons is more than 400, there are more epochs needed to train to the convergence state. When the number is 6400, there even need 6 epochs to the convergence state. However, the training process is far less stable than other learning rules. After training to the convergence state, the standard deviation of accuracy is bigger than 2.4%, while other learning rules are less than 0.3%.

For the Image-Backprop workload, after training one epoch, the accuracy of the model reaches 97.73%, which can be compared with similar architecture DNNs. And the training and inference time are only 1/73 and 1/84 of the STDP learning rule. We think this is because the surrogate backpropagation can train the data batch by batch, but the STDP learning rule can only train the data in one sample once. The surrogate backpropagation can fully use the underlying parallel computing hardware.

For the Image-Conversion workload, after training the network to the convergent state, we get 97.76% accuracy. We export the trained model, transform the model to the SNN model, and apply the data-based weight normalization, we test that the accuracy of the converted SNN model is 96.72%, which is a 1.03% accuracy drop. This is acceptable for applications that are not sensitive to accuracy.

4.4.2. Performance comparison

Fig. 3(b) shows the processing time for training and inferring 10000 images on CPU and GPU, respectively. For training, when the number of neurons is 100, processing images on the CPU is faster than on GPU, but when the number of neurons exceeds 100, training on GPU is faster. And When the number of neurons changes from 400 to 1600 to 6400, the time cost on the CPU increases significantly. However, the time cost on the GPU is always at the same level for 100, 400, 1600, and 6400 neurons. For inference, when the number of neurons is 1600, processing on the CPU costs more time than on the GPU. This phenomenon is related to how GPUs work. When the number of neurons is small, the GPU's large number of parallel computing units may not be fully utilized, leading to less efficient performance compared to the CPU. Additionally, the lower clock frequency of the GPU could contribute to the observed performance difference. While memory synchronization latency may also play a role, it is likely not the primary reason for the performance discrepancy in this case. As a result, the approach of using GPU acceleration may not be applicable or beneficial for all cases, particularly when working with smaller neural networks.

4.4.3. Overall comparison

Table 1 shows the comparison of different learning rules. We cannot achieve the same level of accuracy as the state-of-the-art work using the STDP learning rule as described in [13], and the accuracy the STDP learning rule (88%) achieved is below the surrogate backpropagation (98%) and conversion (96.72%). Meanwhile, the training progress of the STDP learning rule fluctuates greatly even after reaching the convergence state. The standard deviation of accuracy is high as 2.4%, while other learning rules do not exceed 0.3%. This shows that the STDP learning rule is hard to train. STDP is unsupervised, making it considerably more complex to achieve good results compared to supervised backpropagation. This complexity might be the reason for the observed instability. On the other hand, surrogate backpropagation and conversion-based learning rules are easy to train, and there is a huge advantage that they have less modification to existing work than the STDP learning. Thus, they can fully use the current deep learning work, including mature architectures like ResNet [37], EfficientNet [38], and Transformer [39] and high-performance hardware and software systems, like GPUs, TensorFlow, and PyTorch. Despite the disadvantages compared to those two learning rules, the most significant advantage of the STDP learning rule is that unsupervised learning can be used and can avoid the tedious manual labeling of the training dataset. And it has the potential to construct more robust networks that can also perform well for unseen data.

Based on the experimental results presented above, we can summarize the following recommendations for using SNNs:

- Employing Surrogate-Backpropagation and DNN-to-SNN methods is highly recommended, as these two approaches can achieve accuracy comparable to that of DNNs.
- When using the Surrogate-Backpropagation method, it is crucial to find an appropriate activation function that smooths spike peaks well. In the case of DNN-to-SNN, an effective conversion

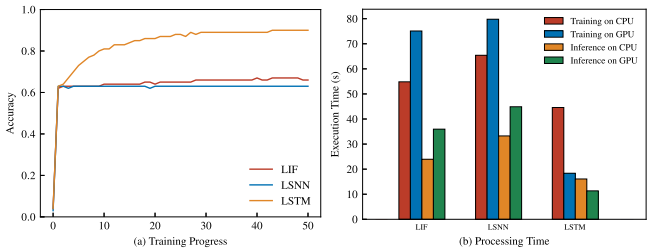


Fig. 4. Changes of accuracy and execution time using different spiking neurons compared to the LSTM unit on the speech recognition task. The time in the figure is to process the train (10000 spoken voices) and test (10000 spoken voices) dataset.

method is required. Otherwise, a significant decrease in accuracy may occur. However, both methods can fully exploit existing DNN works, including algorithms, hardware, and software. The specific choice between these two can be determined based on the actual situation.

- The STDP training rule is relatively challenging to use. Firstly, it is difficult to train and may not necessarily achieve accuracy comparable to the previous two methods. Secondly, the training speed is too slow, and current SNN frameworks cannot efficiently map the STDP algorithm to GPUs. However, if unsupervised learning is desired, the STDP learning rule is the only option.

4.5. Comparison of different spiking neurons

In addition to the two SNN workloads, we add a DNN workload with the same architectures but use the LSTM unit for comparison. We train the three different neural networks on the Google Speech Commands v2 dataset to the convergence state, and Fig. 4 shows the result. And the bottom part of Fig. 5 lists the convergent accuracy and the time taken to achieve that accuracy. The accuracy reaches 62%∼63% after the first epoch for all these three neural networks. After that, the accuracy of neural networks using LIF and LSNN remains at the same level as the first epoch, but the accuracy of the neural network with the LSTM unit reaches 90% after reaching the convergence state. Using the same network architecture, two SNN networks achieve lower accuracy than DNN networks, this may be because the parameters of the SNN network have not been sufficiently tuned, such as the threshold value of firing spikes. [40] indicated that LSNN has better performance than LIF, but the accuracy of LIF and LSNN are similar, which may prove that the parameters of the SNN networks are not tuned well. However, this paper primarily focuses on benchmarking rather then the algorithm, we did not spend much time on parameter optimization, as it is beyond the scope of this work. Even though we can fine-tune the parameters the final accuracy gap is large (67% vs. 90%). Hence we can conclude that for the same architecture, SNN can hardly achieve the accuracy that DNN achieved. In terms of ease of use, SNNs are currently not comparable to DNNs because even though the user does not carefully adjust the LSTM parameters, good results can be achieved.

In terms of processing speed, the LSTM neural network is the fastest in both training and inference. For training one epoch, the time spent on the LIF, LSNN, and LSTM is 779.787 s, 828.515 s, and 190.721 s. The time of the LIF is 4.09 times that of the LSTM. For inferring the test dataset, the time spent on the LIF, LSNN, and LSTM is 39.5661 s, 49.4007 s, and 12.4881 s. The time of the LIF is 3.17 times that of the LSTM. For the same network architecture, using spiking neurons speeds more times than that using non-spiking neurons. This may be due to the framework's inability to effectively map the SNN operators to GPUs.

4.6. Scalability

There are many parallelism methods for accelerating deep learning, such as data parallelism and model parallelism. We can generalize all parallelism patterns into inter-operator parallelism and intra-operator parallelism. Inter-operator parallelism means that operators are mapped to different computing units to execute, and intra-operator parallelism means that one operator is sliced to multiple parts mapped to different computing units. Thus data parallelism and model parallelism can be regarded as one type of inter-operator parallelism. The basic computing unit is a hardware thread within a node or within a distributed system. Reasonably splitting operators and slicing operators according to workloads' characteristics and mapping them to different threads are the key to fully utilizing the system. In this subsection, we investigate the potential of these workloads.

To improve inter-operator parallelism, a common method is to schedule non-dependent operators to execute in parallel. However, this is dependent on the degree of parallelizability of the network architecture, and the execution plan of operators needs to be carefully orchestrated to ensure that dependencies are respected and synchronization points are properly managed. To improve intra-operator parallelism, it is important to consider the differences between stateful networks like SNNs and RNNs, and stateless networks like CNNs and fully connected networks. In stateful networks, SNNs and RNNs are computed sequentially, with states stored for the next computation, making it more challenging to improve intra-operator parallelism. In contrast, CNNs and fully connected networks are stateless, allowing input samples to be split into multiple parts and computed independently, which makes it easier to improve intra-operator parallelism. However, for operators like matrix multiplication and element-wise operations, parallel execution can be achieved on different hardware threads using mature BLAS libraries. This allows for full hardware performance without the need for an elaborate execution plan.

To assess the inter-operator and intra-operator parallelism, we control the thread number of PyTorch's inter-operator and intra-operator thread pool for scheduling inter-operator and intra-operators separately. Although we choose the PyTorch framework as our experimental environment, other frameworks are also applicable, such as TensorFlow, which also has the same thread pool for scheduling inter-operators and intra-operators. For measuring the inter-operator and intra-operator parallelism, we fix the number of inter- and intra-operator threads to 1, respectively, and adjust the number of intra- and inter-operator threads from 1 to 12, respectively, to measure the training time for 10 iterations. Fig. 5 illustrates the inter- and intra-operator parallelism for the SNNBench workloads. As we analyze above, improving the inter-operator parallelism is hard, so we can find that increasing the thread number of inter-operators hardly changes the training time in all the workloads. However, we can find that increasing the thread number of intra-operators only improves the performance of Image-Backprop. For other workloads, more intra-operator threads have little impact and even downgrade the performance, which is counter-intuitive. To explain this phenomenon, we need to analyze the intra-operator execution at a finer granularity.

We count the time of every operator and compare the time executed in a different number of intra-operator threads. We select the most consumed operators in each workload and show the results in Fig. 6. We can find that the most consumed operators in the Image-Backprop workload have good scalability as the number of threads increases so that it can get better performance with more threads. As for other workloads, most workloads have worse performance overall, including Image-Conversion, Speech-LIF, Speech-LSNN, and Speech-LSTM workloads. Different operators have opposite reactions when increasing the number of threads; thus, the overall effect on performance is unpredictable. For the Image-STDP workload, the most three cost operators can be accelerated by increasing the number of threads, but the performance can no longer be improved when the number of

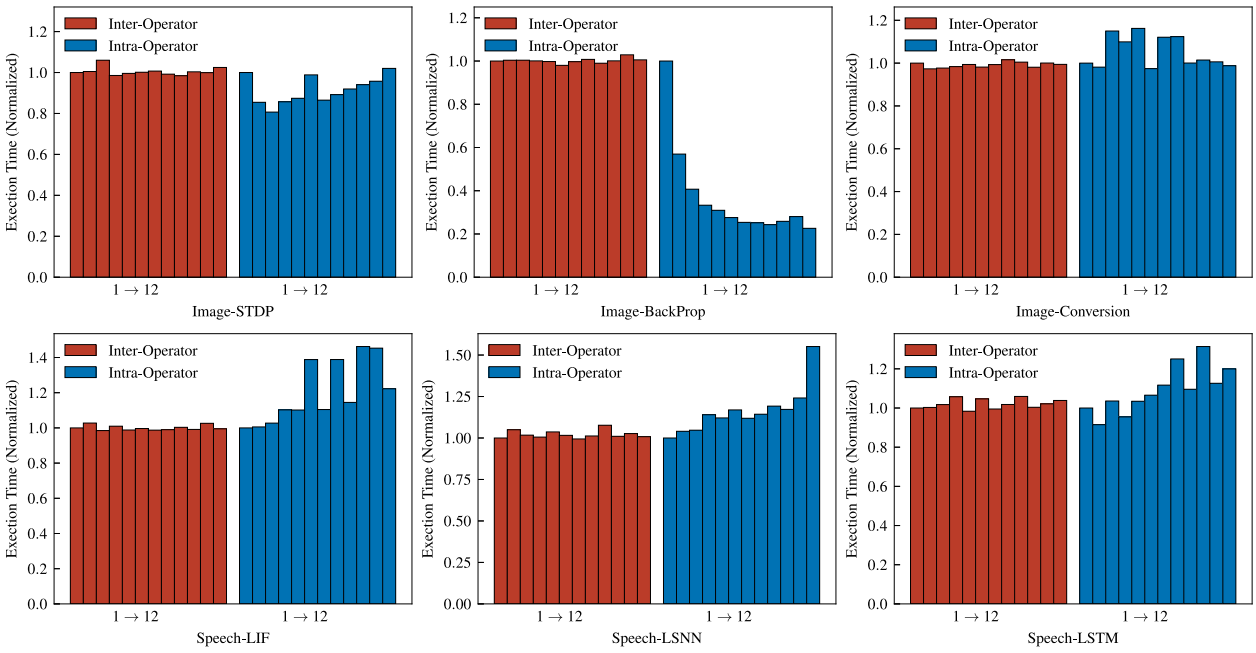


Fig. 5. Operator parallelisms of SNNBench workloads. All workloads are training workloads, without any inference workload. Image-Conversion and Speech-LSTM are included here for comparison to highlight the differences in operator parallelism between mature DNN training workloads and SNN training workloads.

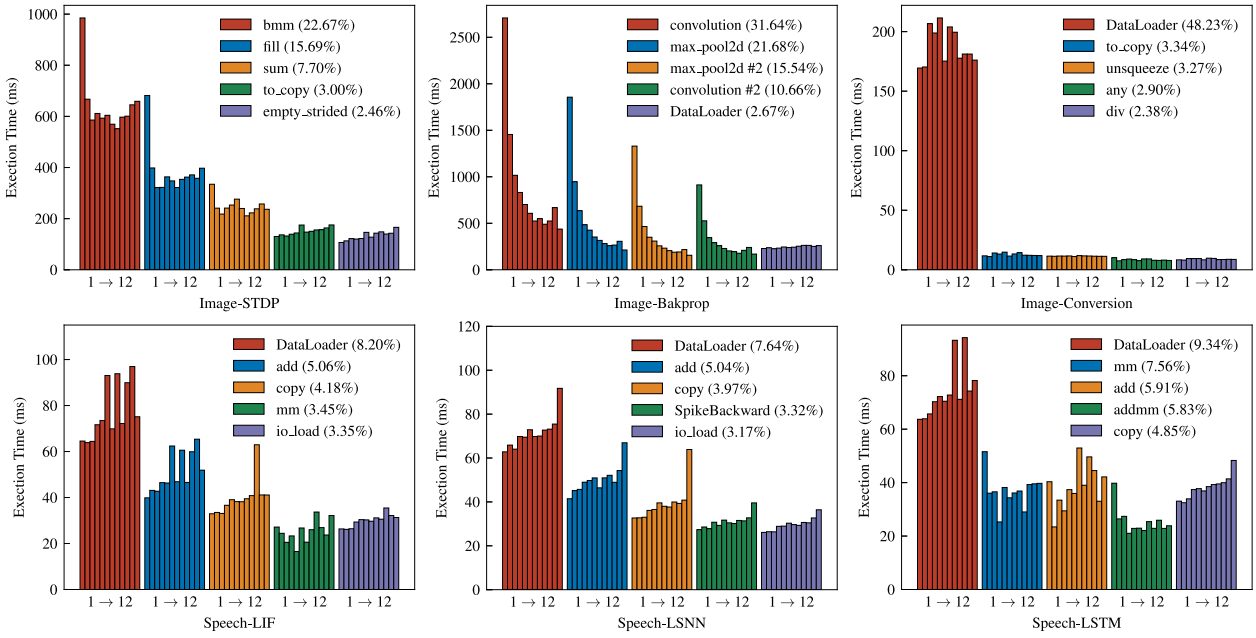


Fig. 6. Intra-operator executing on different threads. All workloads are training workloads, without any inference workload. Image-Conversion and Speech-LSTM are included here for comparison to highlight the differences in operator scalability between mature DNN training workloads and SNN training workloads. We select the most time-consuming operators, and the percentage numbers in brackets represent the proportion of time consumed by each operator.

threads exceeds three. This may be related to the input size of the three operators since the Image-STDP workload does not support batch training, so the data input size is small, and performance is best with three threads. If the data input size becomes larger, it will benefit from more hardware computing units, as we discussed in Section 4.4. From the result, the number of intra-operator threads that gains the best performance is also the thread number that the most cost operators get the best performance. These optimal numbers of threads are 3, 8, 6, 1, 1, and 2, respectively. Therefore, it cannot be simply said that the more hardware threads, the better. Different operators and different data input sizes correspond to different optimal thread numbers. In practice, we can perform an ahead small-size training for searching the optimal thread number before the long training job.

In this section, we analyze the scalability and parallelism potential of SNN workloads. We experiment with varying thread numbers for inter- and intra-operator parallelism and evaluate their impact on training time across different workloads. Our findings indicate that increasing inter-operator threads has negligible effects on training time. However, adjusting the number of intra-operator threads enhances the performance of the Image-Backprop workload but has little or negative impact on others. We observe that different operators and input sizes require different optimal thread numbers, and more hardware threads do not always guarantee better performance. To improve performance,

we suggest employing a strategy that uses varying intra-operator parallel thread numbers for different operators. By optimizing thread count based on the specific operator and input size, each operator's performance can be enhanced, consequently boosting overall neural network training efficiency. In practice, a small-size training run can help determine the optimal thread number before starting a longer training job.

5. Conclusion

SNN, as a promising technology for AI, needs in-depth explorations and further developments to achieve both high performance and high model accuracy—two requisites for AI scenarios. Benchmarks play foundational roles in locating the bottlenecks and making improvements. Existing benchmarks either focus on brain science benchmarking, which has totally different targets and solutions, or cover partial aspects of AI without accuracy information and thus fail to fulfill the above two requisites. This paper proposes an end-to-end AI-oriented benchmarking methodology and presents SNNBench, the first end-to-end SNN benchmark suite. The end-to-end represents two-fold meanings: (1) SNNBench focuses on typical AI applications and covers both training and inference stages for end-to-end evaluation; (2) SNNBench provides an end-to-end training process, covering diverse training strategies and achieving a target accuracy. In total, SNNBench provides nine workloads and covers two learning paradigms, including supervised and unsupervised learning, four connection types, including one-to-one, fully connected, convolutional, and recurrent, and three learning rules, including STDP, backpropagation, and DNN-to-SNN. Our experiments on CPU and GPU show the effectiveness of SNNBench.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to express profound gratitude to Mr. Zhengxin Yang. His insightful discussions and valuable suggestions greatly contributed to the improvement of Fig. 1 in this paper.

References

[1] W. Maass, Networks of spiking neurons: the third generation of neural network models, Neural Netw. 10 (1997) 1659–1671.
[2] K. Roy, A. Jaiswal, P. Panda, Towards spike-based machine intelligence with neuromorphic computing, Nature 575 (2019) 607–617.
[3] R. Brette, M. Rudolph, T. Carnevale, M. Hines, D. Beeman, J.M. Bower, M. Diesmann, A. Morrison, P.H. Goodman, F.C. Harris, et al., Simulation of networks of spiking neurons: a review of tools and strategies, J. Comput. Neurosci. 23 (2007) 349–398.
[4] R.A. Tikidji-Hamburyan, V. Narayana, Z. Bozkus, T.A. El-Ghazawi, Software for brain network simulations: a comparative study, Front. Neuroinform. 11 (2017) 46.
[5] S.J. Van Albada, A.G. Rowley, J. Senk, M. Hopkins, M. Schmidt, A.B. Stokes, D.R. Lester, M. Diesmann, S.B. Furber, Performance comparison of the digital neuromorphic hardware spinnaker and the neural network simulation software nest for a full-scale cortical microcircuit model, Front. Neurosci. 12 (2018) 291.
[6] T.J. Sejnowski, C. Koch, P.S. Churchland, Computational neuroscience, Science 241 (1988) 1299–1306.
[7] A.L. Hodgkin, A.F. Huxley, A quantitative description of membrane current and its application to conduction and excitation in nerve, J. Physiol. 117 (1952) 500.
[8] C. Koch, I. Segev, Methods in Neuronal Modeling: From Ions To Networks, MIT Press, 1998.
[9] N. Brunel, M.C. Van Rossum, Lapicque's 1907 paper: from frogs to integrate-and-fire, Biol. Cybernet. 97 (2007) 337–339.
[10] M. Davies, Benchmarks for progress in neuromorphic computing, Nat. Mach. Intell. 1 (2019) 386–388.
[11] C. Ostrau, J. Homburg, C. Klarhorst, M. Thies, U. Rückert, Benchmarking deep spiking neural networks on neuromorphic hardware, in: International Conference on Artificial Neural Networks, Springer, 2020, pp. 610–621.
[12] S.R. Kulkarni, M. Parsa, J.P. Mitchell, C.D. Schuman, Benchmarking the performance of neuromorphic and spiking neural network simulators, Neurocomputing 447 (2021) 145–160.
[13] P.U. Diehl, M. Cook, Unsupervised learning of digit recognition using spike-timing-dependent plasticity, Front. Comput. Neurosci. 9 (2015) 99.
[14] N. Brunel, X.J. Wang, What determines the frequency of fast network oscillations with irregular neural discharges? I. Synaptic dynamics and excitation-inhibition balance, J. Neurophysiol. 90 (2003) 415–430.
[15] R.A. Tikidji-Hamburyan, J.A. Martínez, C.C. Canavier, Resonant interneurons can increase robustness of gamma oscillations, J. Neurosci. 35 (2015) 15682–15695.
[16] J. Gray, Benchmark Handbook: For Database and Transaction Processing Systems, Morgan Kaufmann Publishers Inc, 1992.
[17] J. Zhan, L. Wang, W. Gao, R. Ren, Benchcouncil's view on benchmarking ai and other emerging workloads, 2019, arXiv preprint arXiv:1912.00572.
[18] Z. Jiang, W. Gao, F. Tang, L. Wang, X. Xiong, C. Luo, C. Lan, H. Li, J. Zhan, Hpc ai500 v2, 0: The methodology, tools, and metrics for benchmarking hpc ai systems, in: 2021 IEEE International Conference on Cluster Computing, CLUSTER, IEEE, 2021, pp. 458–477.
[19] C.D. Schuman, S.R. Kulkarni, M. Parsa, J.P. Mitchell, B. Kay, et al., Opportunities for neuromorphic computing algorithms and applications, Nat. Comput. Sci. 2 (2022) 10–19.
[20] J.H. Lee, T. Delbruck, M. Pfeiffer, Training deep spiking neural networks using backpropagation, Front. Neurosci. 10 (2016) 508.
[21] A. Tavanaei, M. Ghodrati, S.R. Kheradpisheh, T. Masquelier, A. Maida, Deep learning in spiking neural networks, Neural Netw. 111 (2019) 47–63.
[22] Y. Jin, W. Zhang, P. Li, Hybrid macro/micro level backpropagation for training deep spiking neural networks, Adv. Neural Inf. Process. Syst. 31 (2018).
[23] W. Zhang, P. Li, Temporal spike sequence learning via backpropagation for deep spiking neural networks, Adv. Neural Inf. Process. Syst. 33 (2020) 12022–12033.
[24] T. Elsken, J.H. Metzen, F. Hutter, Neural architecture search: A survey, J. Mach. Learn. Res. 20 (2019) 1997–2017.
[25] D. Simon, Evolutionary Optimization Algorithms, John Wiley & Sons, 2013.
[26] L. Deng, The mnist database of handwritten digit images for machine learning research [best of the web], IEEE Signal Process. Mag. 29 (2012) 141–142.
[27] P. Warden, Speech commands: A dataset for limited-vocabulary speech recognition, 2018, arXiv preprint arXiv:1804.03209.
[28] F. Akopyan, J. Sawada, A. Cassidy, R. Alvarez-Icaza, J. Arthur, P. Merolla, N. Imam, Y. Nakamura, P. Datta, G.J. Nam, et al., Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 34 (2015) 1537–1557.
[29] M. Davies, N. Srinivasa, T.H. Lin, G. Chinya, Y. Cao, S.H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, et al., Loihi: A neuromorphic manycore processor with on-chip learning, IEEE Micro 38 (2018) 82–99.
[30] A.F. Agarap, Deep learning using rectified linear units (relu), 2018, arXiv preprint arXiv:1803.08375.
[31] Y. Cao, Y. Chen, D. Khosla, Spiking deep convolutional neural networks for energy-efficient object recognition, Int. J. Comput. Vis. 113 (2015) 54–66.
[32] P.U. Diehl, D. Neil, J. Binas, M. Cook, S.C. Liu, M. Pfeiffer, Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing, in: 2015 International Joint Conference on Neural Networks, IJCNN, IEEE, 2015, pp. 1–8.
[33] PyTorch Documentation, b. Reproducibility. URL: https://pytorch.org/docs/stable/notes/randomness.html.
[34] H. Hazan, D.J. Saunders, H. Khan, D. Patel, D.T. Sanghavi, H.T. Siegelmann, R. Kozma, Bindsnet: A machine learning-oriented spiking neural networks library in python, Front. Neuroinform. 89 (2018).
[35] F. Zenke, S. Ganguli, Superspike: Supervised learning in multilayer spiking neural networks, Neural Comput. 30 (2018) 1514–1541.
[36] PyTorch Documentation, a. Profiler. URL: https://pytorch.org/docs/stable/profiler.html.
[37] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[38] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 6105–6114.
[39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).
[40] G. Bellec, D. Salaj, A. Subramoney, R. Legenstein, W. Maass, Long short-term memory and learning-to-learn in networks of spiking neurons, Adv. Neural Inf. Process. Syst. (2018) 787–797.

BenchCouncil Transactions on Benchmarks, Standards and Evaluations 3 (2023) 100092

Contents lists available at ScienceDirect

BenchCouncil Transactions on Benchmarks, Standards and Evaluations

journal homepage: www.keaipublishing.com/en/journals/benchcouncil-transactions-on-benchmarks-standards-and-evaluations/

Review article

Enabling hyperscale web services

Akshitha Sriraman

Carnegie Mellon University, United States of America

ARTICLE INFO

Keywords:
Hyperscale computing
Computer architecture
Software systems

ABSTRACT

Modern web services such as social media, online messaging, and web search support billions of users, requiring data centers that scale to hundreds of thousands of servers, i.e., *hyperscale*. The key challenge in enabling hyperscale web services arise from (1) an unprecedented growth in data, users, and service functionality and (2) a decline in hardware performance scaling. We highlight a dissertation's contributions in bridging the software and hardware worlds to realize more efficient hyperscale services despite these challenges.

Contents

1. Introduction

Modern web services such as social media, online messaging, web search, video streaming, and online banking often support billions of users, requiring data centers that scale to hundreds of thousands of servers, i.e., *hyperscale* [2]. In fact, the world continues to expect hyperscale computing to drive more futuristic, complex applications such as virtual reality, self-driving cars, conversational AI, and the Internet of Things. This survey paper highlights technologies detailed in the author's PhD dissertation [1] that will enable tomorrow's web services to meet the world's expectations.

The key challenge in enabling hyperscale web services arises from two important trends. First, over the past few years, there has been a radical shift in hyperscale computing due to an unprecedented growth in data [3], users [4], and service functionality [5]. Second, modern hardware can no longer support this growth in hyperscale trends due to a steady decline in hardware performance scaling [6]. To enable this new hyperscale era, hardware architects must become more aware of hyperscale software requirements and software researchers can no longer expect unlimited hardware performance scaling. In short, systems researchers can no longer follow the traditional approach of building each layer of the stack separately. Instead, they must rethink the synergy between the software and hardware worlds. The dissertation [1] creates such a synergy to enable future hyperscale web services.

The dissertation [1] bridges the software and hardware worlds, demonstrating the importance of that bridge in realizing efficient hyperscale web services via solutions that span the systems stack. The specific goal is to (1) design software that is aware of new hardware constraints and (2) architect hardware that efficiently supports new software requirements. To this end, the dissertation [1] spans two broad thrusts: (1) a software and (2) a hardware thrust to analyze the complex software and hardware hyperscale design space to develop efficient cross-stack solutions for hyperscale computation.

In the software thrust, the dissertation [1] contributes *μSuite*, the first open-source benchmark suite of modern web services built with a new hyperscale software paradigm [7]. Next, we[1] use *μSuite* to study software threading design implications in light of today's hardware reality and identify new insights in the age-old research area of software threading [8]. Driven by these insights, we demonstrate how software threading models must be redesigned at hyperscale by presenting an automated approach and tool, *μTune*, that makes intelligent threading decisions during system runtime [8].

In the hardware thrust, the dissertation [1] architects both commodity and custom hardware to efficiently support hyperscale software needs. First, we study the shortcomings in *commodity hardware* running hyperscale services, revealing insights that influenced commercial CPUs [9]. Based on these insights, we present a design tool, *SoftSKU*, that enables cheap commodity hardware to efficiently support new

E-mail address: akshitha@cmu.edu.

[1] When using the "we" pronoun, we refer to the author's dissertation's contributions [1].

https://doi.org/10.1016/j.tbench.2023.100092
Received 20 December 2022; Received in revised form 4 March 2023; Accepted 12 March 2023
Available online 1 April 2023

hyperscale software paradigms, improving the efficiency of real-world services that serve billions of users, saving millions of dollars, and meaningfully reducing the carbon footprint [9].

Next, the dissertation [1] studies how *custom hardware* must be designed at hyperscale, resulting in industry-academia benchmarking efforts, commercial hardware changes, and improved software development [2]. Based on this study's insights, the dissertation presents *Accelerometer*, an analytical model that estimates realistic gains from hardware customization [10].

2. Research goals and limitations of the state-of-the-art

Current software and hardware systems were conceived when we had scarce compute resources, limited data and users, and easy hardware performance scaling. These assumptions are not true today. Today, the world is undergoing a technological revolution where web services require *hyperscale* data centers to efficiently process requests from billions of users. These hyperscale services are facing an unprecedented growth in data [3], users [4], and functionality [5]. Unfortunately, hyperscale computing is emerging at a time when hardware is facing a steady decline in performance scaling [11].

Today, to enable web services, systems researchers typically follow the traditional approach of building each systems stack layer separately. As examples, in the application layer, to support the unprecedented growth in data, users, and functionality, there is shift towards a granular, modular application architecture, with services built with distributed application paradigms like microservices and serverless [12–16]. In the software layers, there is a shift towards light-weight abstractions (e.g., containers) [17–20] in place of heavy-weight ones (e.g., virtualization) [21,22]. In the hardware layer, due to the decline in hardware performance scaling, there is a shift towards building specialized hardware for various "killer" services [23–26].

To design efficient computing systems in light of modern hyperscale service trends and today's hardware reality, systems researchers can no longer afford to build each layer of the stack separately. In short, computer architects must now be aware of software requirements, and software developers can no longer expect continued hardware performance scaling. For example, in addition to the state-of-the-art trend of building custom hardware [23–25], architects must now build hardware that is aware of new service paradigms (e.g., microservices) and software trends (e.g., new threading models). Hence, *the dissertation's [1] first research goal is to rethink the synergy between the software and hardware worlds from the ground up*.

The main challenge in establishing synergy between software and hardware is a large and complex software and hardware design space that makes it intractable to manually identify optimal designs. For example, we discovered that the software threading design space has complex implications induced by the decline of hardware performance scaling, making it impractical for an expert software developer to manually identify the best threading design [8].

Manually navigating this vast and complex design space to make efficient design decisions is often intractable at hyperscale as (1) design implications vary across service loads, (2) trial-and-error methods or experience-based intuition do not systematically capture design space implications, (3) service code evolves quickly, (4) synthetic experiments do not capture production behavior, etc. Hence, to enable futuristic web services, we must achieve *the dissertation's [1] second research goal of automatically navigating, i.e., self-navigating, the complex software and hardware hyperscale design space*.

Given the widespread need for web services, to achieve both these research goals, it is critical to devise mechanisms that can automatically (1) bring new hardware insights when designing software stack layers and (2) draw on fundamental software design principles to systematically architect the hardware layer. Hence, *the dissertation [1] bridges the software and hardware worlds, demonstrating the importance of that bridge in enabling hyperscale web services via efficient self-navigating solutions that span the systems stack. Our vision is to (1) redesign web service software based on new overheads induced by the decline in hardware performance scaling and (2) rearchitect data center commodity and custom hardware to support new software requirements due to the unprecedented growth in data, users, and services.*

To achieve this research vision in a way that self-navigates the complex software and hardware design space, the dissertation [1] spans two thrusts: (1) a software and (2) a hardware thrust. In the software thrust, we ask: how do we design hyperscale web service software based on today's hardware overheads? In the hardware thrust, we ask: how do we architect data center commodity and custom hardware to support the unprecedented growth in hyperscale software trends? It is critical to systematically answer both questions to enable tomorrow's hyperscale web services.

3. Key research contributions

We detail the dissertation's [1] key contributions below.

Enabling the study of modern web services. Modern web services are increasingly built using microservice architectures, wherein a complex web service is composed of numerous distributed microservices such as HTTP connection termination, key–value serving [27], query rewriting [28], access-control management, and protocol routing [29]. Whereas monoliths face greater than 100 ms Service Level Objectives (SLOs) (e.g., ∼300 ms for web search [30]), microservices must often achieve sub-ms SLOs (e.g., ∼100 µs for protocol routing [31]), as many microservices must be invoked serially to serve a user's query. Hence, sub-ms–scale OS/network overheads (e.g., a context switch cost of 5–20 µs [32]) are often insignificant for monoliths. However, the microservice regime differs fundamentally: OS/network overheads (e.g., context switches, network protocol delays, inefficient thread wakeups, and lock contention) that are often minor with monolithic request service times of 100s of milliseconds, can dominate microservice latency distributions. For example, even a single 20 µs context switch implies a 20% latency penalty for a request to a 100 µs-response latency protocol routing microservice [31]. Hence, it is critical to revisit prior conclusions on sub-ms–scale OS/network overheads for the microservice regime [33].

Initially, there existed no representative, open-source benchmarks to study microservices. Widely-used academic data center benchmark suites [34,35], were unsuitable for characterizing sub-ms–scale overheads in microservices as they use monolithic rather than microservice architectures and largely have request service times that are greater than 100 ms. Hence, there was a real need for open-source benchmarks that enable the study of microservices.

To study microservices, as part of the dissertation's software contributions, it introduces the first open-source benchmark suite of end-to-end modern web services composed of microservices, called *µSuite* [7]. *µSuite* includes four end-to-end web services: a content-based high dimensional search for image similarity—`HDSearch`, a replication-based protocol router for scaling fault-tolerant key–value stores—`Router`, a service for performing set algebra on posting lists for document retrieval—`Set Algebra`, and a user-based item recommender system for predicting user ratings—`Recommend`. *µSuite* has been used by researchers in academia and industry (e.g., MIT, UIUC, UT Austin, Georgia Tech, Cornell, ARM, and Intel).

The dissertation uses *µSuite* to study the OS/network performance overheads incurred by microservices. This study reveals that threading interactions with the OS and network layers introduce microsecond-scale overheads that significantly affect microservices, but are insignificant to their monolithic counterparts. Hence, intelligent thread scheduling and better threading models can greatly improve microservice performance.

Redesigning software based on underlying data center hardware constraints. The dissertation's study of OS/network performance overheads using *µSuite* showed that microservices can benefit from better threading designs. These threading-induced overheads are due to today's hardware reality, where network devices have sped up while CPU

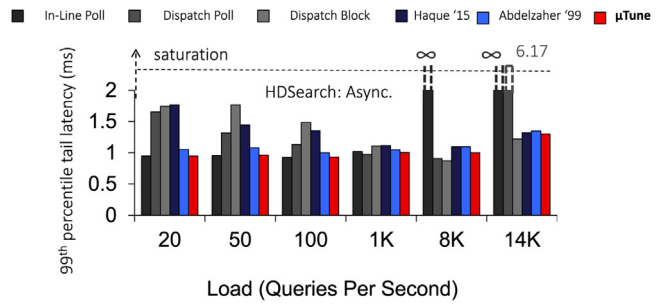


Fig. 1. *µTune*'s latency compared to existing techniques [37,38]: *µTune* lowers latency by 1.9×.

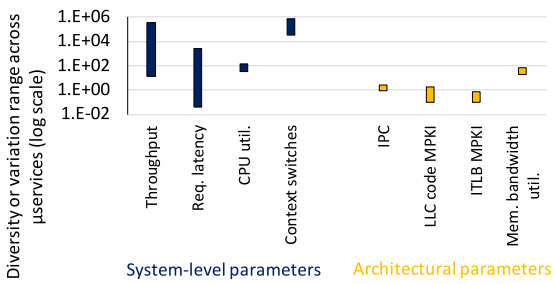


Fig. 2. Variation in system-level & architectural traits across microservices: Production microservices face diverse bottlenecks.

performance scaling has nearly stopped [36]. Today, a CPU thread's accesses to the underlying OS/network stacks cause threading-induced overheads that arise from thread contention on locks, thread wakeup delays, and context switching. Hence, analyzing software threading designs' implications and rethinking threading models for modern microservices has become a deeply important problem.

To study threading-induced software overheads that arise due to hardware constraints, there is a need to systematically analyze the sub-ms–scale OS and network overheads that arise from threading and concurrency design decisions. As part of the dissertation's software contributions [1], we use *µSuite* to systematically introduce and characterize a *taxonomy of threading models* [8]. This taxonomy is composed of software threading dimensions commonly used to build a microservice, such as synchronous or asynchronous RPCs, in-line or dispatched RPC handlers, and interrupt- or poll-based network reception. We also vary thread pool sizes dedicated to the various functionalities, i.e., network polling, RPC handling, and response execution. These threading design axes yield a rich space of microservice software threading architectures that interact with the underlying OS and hardware in starkly varied ways. Hence, this threading taxonomy and analysis enables expert and novice developers alike to guide their service threading designs.

The dissertation [1] makes the important observation that no single threading model is best across all load conditions, paving the way for an automatic load adaptation system that tunes threading models to improve performance. Specifically, our threading model study demonstrates that the relationship between optimal threading model and service load is complex—one could not expect a developer to pick the best threading model a priori. For example, at low load, models that poll for network traffic perform best, as they avoid thread wakeup delays. Conversely, at high load, models that separate network polling from RPC execution enable higher service capacity and blocking outperforms polling for incoming network traffic as it avoids wasting CPU on fruitless poll loops. Hence, exploiting these inherent threading model trade-offs during system runtime can significantly improve microservice latency.

To exploit threading trade-offs at runtime, the dissertation [1] presents and makes open source a system, *µTune* [8], that features a framework that builds upon open-source RPC platforms [39] to abstract threading model design from service code. *µTune*'s second feature is an intelligent run-time system that determines load via event-based monitoring and automatically adapts to time-varying service load by self-navigating the threading design space, i.e., tuning threading models and scaling thread pool sizes. As shown in Fig. 1, both features enable *µTune* to dynamically reduce microservice latency by 1.9× over static peak load-sustaining threading models (that an expert developer might have picked) and state-of-the-art adaptation techniques [37,38,40].

Architecting commodity hardware for new service paradigms. At global user population scale, key web services composed of numerous microservices can account for an enormous installed base of physical hardware. For example, across Facebook's global server fleet, seven key microservices in four service domains run at hyperscale, occupying a large portion of the fleet [9]. In light of this new microservice software paradigm, it is important to answer the question: do commodity server platforms serve microservices well? Are there common bottlenecks across microservices that we might address when designing future server architectures?

To identify whether commodity hardware efficiently supports microservices, the dissertation [1] undertakes comprehensive system-level and architectural analyses of Facebook's key production microservices serving live traffic. As shown in Fig. 2, we find that service functionality distribution across microservices has resulted in enormous diversity in system (e.g., request latency and CPU utilization) and architectural requirements (e.g., Instructions Per Cycle and LLC code misses per kilo instruction), with new CPU bottlenecks (e.g., high I/O processing latency and I-cache misses). Our identified bottlenecks made hardware vendors reconsider the benchmarks they used for decades to evaluate new servers.

As examples, we find that caching microservices [41] require intensive I/O and microsecond-scale response latency and frequent OS context switches constitute 18% of CPU time. In contrast, a `Feed` [42] microservice computes for seconds per request with minimal OS interaction. Facebook's `Web` [43] microservice exhibits massive instruction footprints, leading to astonishing I-cache misses and branch mispredictions, while other microservices exhibit smaller code footprints. Some microservices depend heavily on floating-point performance while others have no floating-point instructions.

The great diversity in hardware bottlenecks across microservices might suggest a strategy to specialize CPU architectures to suit each microservice's distinct needs. However, hyperscale enterprises have strong economic incentives to limit hardware platforms' diversity to (1) maintain fungibility of hardware resources, (2) preserve procurement advantages that arise from economies of scale, and (3) limit the overhead of qualifying/testing myriad hardware platforms. As such, there is an immediate need for strategies that extract greater performance from existing commodity server architectures to efficiently support diverse microservices on commodity hardware.

As part of the dissertation's hardware contributions, it introduces an automated approach and tool to improve hyperscale microservice performance on cheap commodity server architectures (often called "SKUs", short for "Stock Keeping Units") [9]. This approach called *SoftSKU* is a design-time strategy that tunes coarse-grain (e.g., boot time) OS and hardware configuration knobs available on commodity processors to help a processor platform or SKU better support its assigned microservice. OS and CPUs provide several specialization knobs; we focus on seven: (1) core frequency, (2) uncore frequency, (3) active core count, (4) code vs. data prioritization in the last-level cache ways, (5) hardware prefetcher configuration, (6) use of transparent huge pages, and (7) use of statically-allocated huge pages. The dissertation also proposes new CPU knobs (e.g., Branch Target Buffer ways) that can be made configurable to create finer-grained soft SKUs.

Manually identifying a microservice-specific *SoftSKU* is impractical as the design space is large, code evolves quickly, synthetic load tests do not often capture production behavior, and the effects of tuning a single

knob are often small. Hence, we build an automated design tool—μSKU—that self-navigates the hardware configuration design space to optimize a hardware SKU for each microservice. μSKU automatically varies configurable server knobs, by searching within a predefined design space via A/B testing, where it compares the performance of two identical servers that differ only in their knob configuration. μSKU collects copious fine-grain performance measurements while conducting automated A/B tests on production systems serving live traffic to search for statistically significant performance gains. We evaluate μSKU on hyperscale production microservices and show that the ensuing soft SKUs outperform stock and production server configurations by up to 7.2% and 4.5% respectively, with no additional hardware requirement [9].

SoftSKU demonstrates that before resorting to hardware customization, there is still significant performance to be extracted from cheap commodity CPUs by tuning their OS and hardware knobs. In this manner, soft SKUs significantly improve the performance efficiency of real-world Facebook microservices that serve billions of users, saving millions of dollars and meaningfully reduce the global carbon footprint [44]. Since this work [9], several hyperscale enterprises have dedicated teams of engineers to explore additional configurable hardware/OS soft-SKU knobs (e.g., SIMD width).

Architecting custom hardware for new service paradigms. The *SoftSKU* work [9] revealed that microservices are so diverse that they could benefit from custom hardware. In fact, to improve hardware efficiency, several architects today work on developing numerous specialized hardware accelerators for important microservice domains (e.g. Machine Learning tasks). Designing such custom hardware accelerators for each microservice operation might improve performance or energy. However, designing custom hardware for each microservice operation is prohibitively expensive at hyperscale since data center operators lose procurement advantages that arise from economies of scale and must also develop and test on myriad custom hardware platforms. Hence, an important question arises: *which microservice software operations consume the most CPU cycles and are worth accelerating in the hardware?*

To build specialized accelerators for these key microservice operations, it is important to first systematically identify which type of accelerator meets microservice requirements and is worth designing and deploying. Deploying specialized hardware is risky at hyperscale, as the hardware might under-perform due to performance bounds from the microservice's software interaction with the hardware, resulting in high monetary losses. To make well-informed hardware decisions, it is crucial to systematically answer the following question early in the design phase of a new accelerator to determine whether the new accelerator is worth designing: *how much can the accelerator realistically improve its targeted microservice overhead?*

To answer the first question posed above, we undertake a comprehensive study of how microservices spend their CPU cycles (as part of the dissertation's hardware contributions). In Fig. 3, we study seven key hyperscale Facebook microservices in four diverse service domains that run across hundreds of thousands of servers, occupying a large portion of the global server fleet. Our study reveals that microservices spend only a small fraction of CPU cycles executing their main application functionality (e.g., a Machine Learning task); the remaining cycles are spent in common *orchestration overheads*, i.e., operations that are not critical to the main microservice functionality (e.g., I/O notification, logging, and compression). Accelerating such common building blocks can greatly improve performance. Already, a few hardware vendors have used this study's insights to influence hardware customization for orchestration overheads [2] (e.g., this study's insights brought about the Intel's Infrastructure Processing Unit [45]).

Our characterization drove a hardware vendor to consider more representative benchmarks (in place of traditional ones used for decades) when evaluating hardware designs [2]. This study resulted in an industry-academia joint collaborative effort to design and open-source representative data center benchmarks. Additionally, our characterization tool has been integrated into Facebook's fleet-wide performance

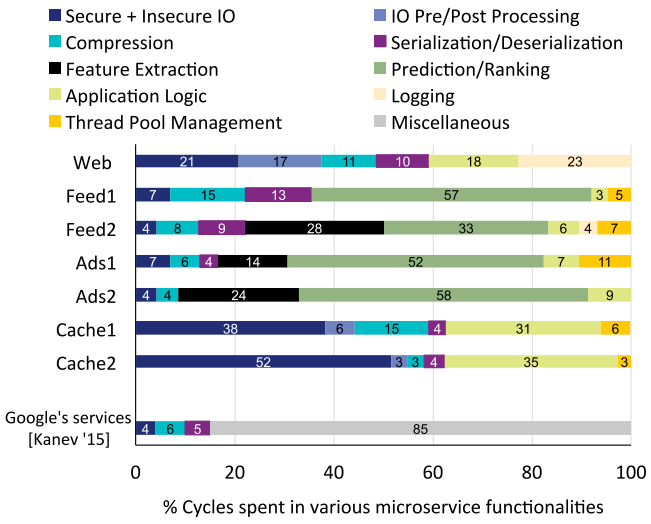


Fig. 3. Breakdown of cycles spent in Facebook production service operations: Orchestration overheads are significant and common.

monitoring infrastructure; it curates statistics from hundreds of thousands of servers to help developers visualize the performance impact of their code changes at hyperscale [2].

To answer the second question posed above, we develop *Accelerometer*,[2] an analytical model for hardware acceleration [10]. *Accelerometer* estimates realistic gains from hardware acceleration by self-navigating the various performance bounds that arise from a microservice's software interactions with the hardware. *Accelerometer* identifies performance bounds and design bottlenecks early in the hardware design cycle, and provides insight into which hardware acceleration strategies may alleviate these bottlenecks.

Accelerometer models both synchronous and asynchronous microservice software interactions for three hardware acceleration strategies—on-chip, off-chip, and remote. It assumes an abstract system with (1) a *host*: a general-purpose CPU, (2) an *accelerator*: custom hardware to accelerate a kernel, and (3) an *interface*: the communication layer between the host and the accelerator (e.g., a PCIe link). It models the microservice throughput speedup and the per-request latency reduction. We validate *Accelerometer*'s utility via three retrospective case studies conducted on production systems, by comparing model-estimated speedup with real service speedup—*Accelerometer* estimates the real microservice speedup with an error that is $\leq$3.7%. We also use *Accelerometer* to project speedup with new accelerators.

As services evolve, Accelerometer's generality makes it more suitable in determining new hardware requirements early in the design phase. Since we validated *Accelerometer* in production and made it open-source, it has been adopted by many hyperscale enterprises (e.g., with developing encryption/compression accelerators) to make well-informed hardware decisions [2].

Overall, the dissertation's primary, unique contribution is bridging the software and hardware worlds and demonstrating the importance of that bridge in realizing efficient hyperscale services via cross-stack solutions. Specifically, through the software and hardware contributions below, we realize efficient services from analytical models on paper to deployment at hyperscale.

- Software contributions.
  - The dissertation is the first to present an open-source benchmark suite of microservices that facilitates future academic and industry research [7].

[2] *Accelerometer* was recognized for its long-term impact potential with an IEEE Micro Top Picks distinction (one of top 12 computer architecture papers in 2020) [2].

- The dissertation identifies new insights in the age-old research area of software threading models that led to redesigning threading models for hyperscale web services [8].

• Hardware contributions.

- The dissertation analyzes shortcomings in commodity hardware running hyperscale services that influenced the design of commercial CPUs [9].
- The dissertation demonstrates how commodity hardware can be used efficiently to enable hyperscale services that led to real-world data centers prioritizing this approach over today's hardware customization trend [9].
- The dissertation presents a systematic understanding of hardware customization opportunities at hyperscale that enabled industry-academia joint benchmarking efforts, influenced commercial hardware design, and improved software development [2,10].
- The dissertation presents a rigorous, analytical alternative to ad hoc hardware customization approaches that enabled real-world hyperscale data centers to make well-informed hardware investments [2, 10].

4. Future directions

There are many exciting avenues of future work that follow from the research presented in the dissertation; some of these are summarized below.

Enabling cross-stack designs for emerging service paradigms. Apart from the microservice paradigm studied in the dissertation [1], modern web systems are being built with newer service paradigms such as serverless. Each new paradigm introduces unique overheads that affect hyperscale efficiency. For example, unlike microservices, serverless systems introduce new inefficiencies from container launch and warm-up delays, increased communication, and greater scalability issues. Techniques developed in the dissertation can help future systems support emerging service paradigms.

Rethinking hardware–software co-design for hyperscale overheads. The dissertation's study of real-world microservices revealed several system overheads that particularly arise at hyperscale. The dissertation's work reduced a few predominant overheads such as I-cache misses and I/O event notification. Going forward, there is a need to optimize other overheads identified in the dissertation. For example, apart from improving I/O event notification, we must optimize the end-to-end I/O processing path to efficiently (1) receive/send a large number of I/O, (2) operate the CPU when awaiting IO and (3) process large I/O just as well as small I/O transfers.

Mitigating the killer microsecond problem in modern services. As the dissertation [1] shows, modern servers have mechanisms to effectively hide nanosecond-scale stalls (e.g., OoO cores) and millisecond-scale stalls (e.g., context switching), but lack efficient support to hide microsecond-scale stalls that critically affect modern services. To mitigate microsecond-scale stalls (often called the "killer microsecond" [46]), we must study various microsecond-scale accesses' (e.g., modern networking, non-volatile memories, and accelerator accesses') impact on efficiency, to develop cross-stack solutions. For example, we must build "microsecond-aware" stacks with reduced lock contention, fast interrupts, efficient spin-polling, and better scheduling.

Using machine learning to self-navigate the hyperscale design space. As the hyperscale software/hardware design space continues to become more complex, we foresee empirical systems leveraging recent improvements in ML models to manage design complexity, to use ML techniques to self-navigate complex software/hardware design spaces such as resource allocation, request scheduling, and bottleneck identification.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Enabling Hyperscale Web Services (Ph.D. thesis).
[2] A. Sriraman, A. Dhanotia, Understanding acceleration opportunities at hyperscale, IEEE Micro (2021).
[3] What's causing the exponential growth of data? https://insights.nikkoam.com/articles/2019/12/whats_causing_the_exponential.
[4] Digital 2020: 3.8 billion people use social media. https://wearesocial.com/blog/2020/01/digital-2020-3-8-billion-people-use-social-media.
[5] The Top 12 future web development trends in 2021. https://dev.to/adhyaswarnali/the-top-12-future-web-development-trends-in-2021-25k5.
[6] G.E. Moore, Cramming More Components onto Integrated Circuits, McGraw-Hill, New York, 1965.
[7] A. Sriraman, T.F. Wenisch, μSuite: A benchmark suite for microservices, in: IEEE International Symposium on Workload Characterization, 2018.
[8] A. Sriraman, T.F. Wenisch, μTune: Auto-tuned threading for OLDI microservices, in: USENIX Conference on Operating Systems Design and Implementation, 2018.
[9] A. Sriraman, A. Dhanotia, T.F. Wenisch, SoftSKU: Optimizing server architectures for microservice diversity @scale, in: The International Symposium on Computer Architecture, 2019.
[10] A. Sriraman, A. Dhanotia, Accelerometer: Understanding acceleration opportunities for data center overheads at hyperscale, in: International Conference on Architectural Support for Programming Languages and Operating Systems, 2020.
[11] M.M. Waldrop, The chips are down for Moore's law, Nat. News 530 (7589) (2016) 144.
[12] A brief history of microservices. https://www.dataversity.net/a-brief-history-of-microservices/.
[13] Y. Gan, Y. Zhang, D. Cheng, A. Shetty, P. Rathi, N. Katarki, A. Bruno, J. Hu, B. Ritchken, B. Jackson, K. Hu, M. Pancholi, Y. He, B. Clancy, C. Colen, F. Wen, C. Leung, S. Wang, L. Zaruvinsky, M. Espinosa, R. Lin, Z. Liu, J. Padilla, C. Delimitrou, An open-source benchmark suite for microservices and their hardware-software implications for cloud & edge systems, in: International Conference on Architectural Support for Programming Languages and Operating Systems, 2019.
[14] S. Kanev, K. Hazelwood, G.-Y. Wei, D. Brooks, Tradeoffs between power management and tail latency in warehouse-scale applications, in: IEEE International Symposium on Workload Characterization, 2014.
[15] N. Dmitry, S.-S. Manfred, On micro-services architecture, Int. J. Open Inf. Technol. (2014).
[16] I. Nadareishvili, R. Mitra, M. McLarty, M. Amundsen, Microservice architecture: Aligning principles, practices, and culture, 2016.
[17] About DPDK. https://www.dpdk.org/about/.
[18] A. Belay, G. Prekas, A. Klimovic, S. Grossman, C. Kozyrakis, E. Bugnion, IX: A protected dataplane operating system for high throughput and low latency, in: USENIX Conference on Operating Systems Design and Implementation, 2014.
[19] M. Marty, M. de Kruijf, J. Adriaens, C. Alfeld, S. Bauer, C. Contavalli, M. Dalton, N. Dukkipati, W.C. Evans, S. Gribble, N. Kidd, R. Kononov, G. Kumar, C. Mauer, E. Musick, L. Olson, E. Rubow, M. Ryan, K. Springborn, P. Turner, V. Valancius, X. Wang, A. Vahdat, Snap: A microkernel approach to host networking, in: ACM Symposium on Operating Systems Principles, 2019.
[20] Key components of a software defined data center. https://www.evolvingsol.com/2018/04/17/components-software-defined-data-center/.
[21] Dawn of the data center operating system. https://www.infoworld.com/article/2906362/dawn-of-the-data-center-operating-system.html.
[22] Containers. https://a16z.com/2015/01/22/containers/.
[23] A. Putnam, A.M. Caulfield, E.S. Chung, D. Chiou, K. Constantinides, J. Demme, H. Esmaeilzadeh, J. Fowers, G.P. Gopal, J. Gray, M. Haselman, S. Hauck, S. Heil, A. Hormati, J.-Y. Kim, S. Lanka, J. Larus, E. Peterson, S. Pope, A. Smith, J. Thong, P.Y. Xiao, D. Burger, A reconfigurable fabric for accelerating large-scale datacenter services, in: International Symposium on Computer Architecuture, 2014.
[24] N.P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers, R. Boyle, P.-l. Cantin, C. Chao, C. Clark, J. Coriell, M. Daley, M. Dau, J. Dean, B. Gelb, T.V. Ghaemmaghami, R. Gottipati, W. Gulland, R. Hagmann, C.R. Ho, D. Hogberg, J. Hu, R. Hundt, D. Hurt, J. Ibarz, A. Jaffey, A. Jaworski, A. Kaplan, H. Khaitan, D. Killebrew, A. Koch, N. Kumar, S. Lacy, J. Laudon, J. Law, D. Le, C. Leary, Z. Liu, K. Lucke, A. Lundin, G. MacKean, A. Maggiore, M. Mahony, K. Miller, R. Nagarajan, R. Narayanaswami, R. Ni, K. Nix, T. Norrie, M. Omernick, N. Penukonda, A. Phelps, J. Ross, M. Ross, A. Salek, E. Samadiani, C. Severn, G. Sizikov, M. Snelham, J. Souter, D. Steinberg, A. Swing, M. Tan, G. Thorson, B. Tian, H. Toma, E. Tuttle, V. Vasudevan, R. Walter, W. Wang, E. Wilcox, D.H. Yoon, In-datacenter performance analysis of a tensor processing unit, in: International Symposium on Computer Architecture, 2017.

[25] E. Chung, J. Fowers, K. Ovtcharov, M. Papamichael, A. Caulfield, T. Massengill, M. Liu, D. Lo, S. Alkalay, M. Haselman, M. Abeydeera, L. Adams, H. Angepat, C. Boehn, D. Chiou, O. Firestein, A. Forin, K.S. Gatlin, M. Ghandi, S. Heil, K. Holohan, A. El Husseini, T. Juhasz, K. Kagi, R.K. Kovvuri, S. Lanka, F. van Megen, D. Mukhortov, P. Patel, B. Perez, A. Rapsang, S. Reinhardt, B. Rouhani, A. Sapek, R. Seera, S. Shekar, B. Sridharan, G. Weisz, L. Woods, P. Yi Xiao, D. Zhang, R. Zhao, D. Burger, Serving DNNs in real time at datacenter scale with project brainwave, IEEE Micro 38 (2) (2018) 8–20.

[26] B. Abali, B. Blaner, J. Reilly, M. Klein, A. Mishra, C.B. Agricola, B. Sendir, A. Buyuktosunoglu, C. Jacobi, W.J. Starke, H. Myneni, C. Wang, Data compression accelerator on IBM POWER9 and Z15 processors, in: International Symposium on Computer Architecture, 2020.

[27] B. Fitzpatrick, Distributed caching with memcached, Linux J. (2004).

[28] M. Barhamgi, D. Benslimane, B. Medjahed, A query rewriting approach for web service composition, IEEE Trans. Serv. Comput. (2010).

[29] Mcrouter. https://github.com/facebook/mcrouter.

[30] B. Vamanan, J. Hasan, T. Vijaykumar, Deadline-aware datacenter TCP (D2TCP), in: ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, 2012.

[31] Y. Zhang, D. Meisner, J. Mars, L. Tang, Treadmill: Attributing the source of tail latency through precise load testing and statistical inference, in: International Symposium on Computer Architecture, 2016.

[32] D. Tsafrir, The context-switch overhead inflicted by hardware interrupts (and the enigma of do-nothing loops), in: Workshop on Experimental Computer Science, 2007.

[33] L. Barroso, M. Marty, D. Patterson, P. Ranganathan, Attack of the killer microseconds, Commun. ACM (2017).

[34] M. Ferdman, A. Adileh, O. Kocberber, S. Volos, M. Alisafaee, D. Jevdjic, C. Kaynak, A.D. Popescu, A. Ailamaki, B. Falsafi, Clearing the clouds: A study of emerging scale-out Workloads on modern hardware, in: International Conference on Architectural Support for Programming Languages and Operating Systems, 2012.

[35] PerfKit benchmarker. https://github.com/GoogleCloudPlatform/PerfKitBenchmarker.

[36] A. Danowitz, K. Kelley, J. Mao, J.P. Stevenson, M. Horowitz, CPU DB: Recording microprocessor history, Commun. ACM (2012).

[37] M.E. Haque, Y.h. Eom, Y. He, S. Elnikety, R. Bianchini, K.S. McKinley, Few-to-many: Incremental parallelism for reducing tail latency in interactive services, in: International Conference on Architectural Support for Programming Languages and Operating Systems, 2015.

[38] T.F. Abdelzaher, N. Bhatti, Web server QoS management by adaptive content delivery, in: International Workshop on Quality of Service, 1999.

[39] gRPC. https://github.com/heathermiller/dist-prog-book/blob/master/chapter/1/gRPC.md.

[40] K. Langendoen, J. Romein, R. Bhoedjang, H. Bal, Integrating polling, interrupts, and thread management, in: Symposium on the Frontiers of Massively Parallel Computing, 1996.

[41] N. Bronson, Z. Amsden, G. Cabrera, P. Chakka, P. Dimov, H. Ding, J. Ferris, A. Giardullo, S. Kulkarni, H.C. Li, et al., TAO: Facebook's distributed data store for the social graph, in: USENIX Annual Technical Conference, 2013.

[42] M. Zuckerberg, R. Sanghvi, A. Bosworth, C. Cox, A. Sittig, C. Hughes, K. Geminder, D. Corson, Dynamically providing a news feed about a user of a social network. https://patents.google.com/patent/US7669123B2/en.

[43] G. Ottoni, HHVM JIT: A profile-guided, region-based compiler for PHP and hack, in: Conference on Programming Language Design and Implementation, 2018.

[44] Accelerometer & SoftSKU: Improving HW performance for diverse microservices. https://engineering.fb.com/data-center-engineering/accelerometer-and-softsku/.

[45] P. Kummrow, The IPU: A new, strategic resource for cloud service providers, 2021, https://itpeernetwork.intel.com/ipu-cloud/. [Online; accessed 22-August-2021].

[46] A. Mirhosseini, A. Sriraman, T.F. Wenisch, Enhancing server efficiency in the face of killer microseconds, in: International Symposium on High Performance Computer Architecture, 2019.

BenchCouncil Transactions on Benchmarks, Standards and Evaluations 3 (2023) 100107

Contents lists available at ScienceDirect

BenchCouncil Transactions on Benchmarks, Standards and Evaluations

journal homepage: www.keaipublishing.com/en/journals/benchcouncil-transactions-on-benchmarks-standards-and-evaluations/

Case report

e₹—The digital currency in India: Challenges and prospects

Md. Asraful Haque *, Mohd Shoaib

Department of Computer Engineering, Z.H. College of Engineering & Technology, Aligarh Muslim University, Aligarh 202002, India

ARTICLE INFO

Keywords:
Cryptocurrency
CBDC
e-Rupee(e₹)
Digital Rupee
RBI
Blockchain

ABSTRACT

The Reserve Bank of India (RBI) has recently launched the country's first pilot project for the digital currency known as the digital rupee or e-Rupee (e₹). The launch of the digital rupee represents a significant advancement in the "Digital India" revolution. It will be a fantastic opportunity for India since it might make conducting business easier while enhancing the security and resilience of the overall payments system. Digital currency attempts to rapidly progress monetary policy to disrupt physical money, lower the cost of financial transactions, and reshape how the money will circulate. Although the effects of digital currency cannot be foreseen, it is extremely important to thoroughly research digital currency and its effects on the operational stage. The development of a digital currency infrastructure has some challenges in terms of performance, scalability, and different usage scenarios. The article clarifies what e₹ is. How does it work? What makes it different from cryptocurrencies? What are the major challenges and prospects for it in India?

1. Introduction

The shape and purposes of money have changed over time as a result of how the economy and payment system have developed. The evolution of the concept of money from Commodity to Digital Currency is shown in Fig. 1.

India has made remarkable strides in digital payment innovation. Digital currency is not a new concept. We already make regular payments using digital methods such as Real Time Gross Settlement (RTGS), National Electronic Funds Transfer (NEFT), and Immediate Payment Service (IMPS). They are secure, effective, and accessible 24 × 7. Recently, UPI (Unified Payments Service), a revolutionary payment system, has had a significant impact on the nation's economic system and has become a model for other nations looking to develop a scalable, convenient, and real-time payment system. The objective of all digital payment methods is to offer consumers an alternative mode of paying physical cash [1]. Cryptocurrency is a type of digital currency where transactions are verified and records are kept by a decentralized system employing encryption [2]. The showrunner of cryptocurrency is the blockchain [3] or a distributed ledger that keeps track of transactions and distributes access to the authorized users. There are already thousands of different digital currencies, which are collectively referred to as cryptocurrencies. The most well-known example of a fully decentralized, peer-to-peer cryptocurrency is Bitcoin. The Bitcoin debut in 2009 is still a favorite among investors and miners. It sparked the "revolution" in cryptocurrencies that gave rise to many well-known coins including Ethereum, Litecoin, Tether, XRP, etc. India opposed the use of Bitcoin and other cryptocurrencies for many reasons. One reason is that the government is concerned about the potential for money laundering and financing of illegal activities using these digital assets. The Reserve Bank of India (RBI) does not have any control over the transactions of cryptocurrencies. Another reason is that the use of cryptocurrencies could potentially lead to a decline in demand for traditional fiat currencies, such as the Indian rupee, which could in turn have negative impacts on the country's economy. In addition, there are concerns about the volatility of the prices of cryptocurrencies and the lack of regulatory oversight in the market. People were alerted in April 2018 that cryptocurrencies are not accepted as legal currency in India [4]. In 2019, the finance ministry drafted a bill prohibiting cryptocurrency mining, ownership, sales, issuance, transfers, and use in India. A person might face a hefty fine or up to 10 years in jail if proven guilty of violating the law. The Supreme Court of India, however, removed the restriction in March 2020 [5]. Then the finance ministry declared that cryptocurrencies would be subject to a 30% tax as well as the launch of India's own CBDC, known as the digital rupee, in the Union Budget 2022–2023. The Reserve Bank of India (RBI) launched the country's first pilot project for digital currency e₹ (e-Rupee) on 1st December, 2022 [6].

e-Rupee, a digital version of Indian Rupee would be a central bank digital currency (CBDC) backed by RBI. The introduction of a new CBDC may worry the cryptocurrency world because of the confusion around it. With the intention of cutting away the middleman and designing a system of trust independent of any organization, cryptocurrencies were created. The e-Rupee, which is only the digital equivalent

Abbreviations: CBDC, Central Bank Digital Currency

* Corresponding author.
E-mail addresses: md_asraf@zhcet.ac.in (M.A. Haque), md.shoaibs@zhcet.ac.in (M. Shoaib).

https://doi.org/10.1016/j.tbench.2023.100107
Received 17 April 2023; Received in revised form 2 May 2023; Accepted 2 May 2023
Available online 4 May 2023

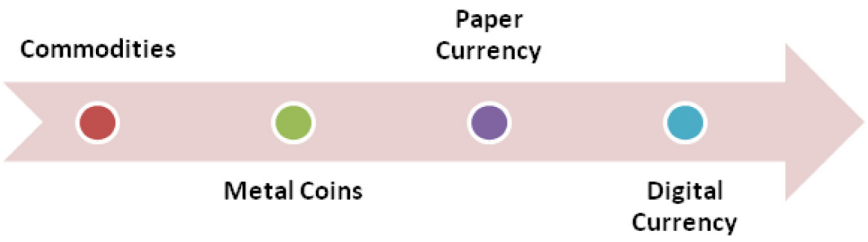


Fig. 1. The evolution of payment system.

of fiat currency, would once again rely on RBI. It is possible that the e-Rupee would be available for use through a digital wallet or banking app. The digital rupee could help to reduce the reliance on cash by providing a secure and convenient alternative for making payments. The success of e-Rupee, will depend on a number of factors, including user adoption, merchant acceptance, regulatory backing, and public confidence. It would probably need to take into account a variety of technological, legal, and regulatory challenges and implement the necessary solutions.

The structure of our paper is organized as follows. Section 2 provides an overview of digital currencies. Section 3 explains operating procedure of e-Rupee. Section 4 suggests a general architecture of an e-Rupee system based on a private blockchain network. Section 5 analyzes the feasibility of the scheme. Section 6 compares e-Rupee with some other CBDCs. Section 7 discusses the future prospects of e-Rupee. Section 8 highlights the limitations of e-Rupee. Section 9 concludes the paper with some remarks.

2. Digital currencies

Digital currencies are a type of currency that exists solely in digital form, without a physical counterpart like paper money or coins. They are often used for online transactions and can be transferred electronically between parties. Digital currencies continue to gain popularity and are increasingly being adopted by businesses and consumers around the world. Digital currencies can be broadly divided into two categories: Central Bank Digital Currency (CBDC) and Cryptocurrency.

2.1. Central Bank Digital Currency (CBDC)

CBDCs are digital versions of traditional fiat currencies issued and backed by central banks. CBDCs can be used for a variety of purposes, such as facilitating cross-border payments, reducing the costs of cash management, and improving financial inclusion. CBDC can use blockchain technology, but it is not a requirement [7,8]. CBDCs can be implemented using a variety of technological approaches. Blockchain technology is one possible approach for implementing CBDCs, as it provides a distributed and transparent ledger that can be used to track transactions and ensure the integrity of the currency. Some central banks are exploring the use of blockchain technology for CBDCs, such as the Central Bank of the Bahamas which launched the Sand Dollar, a digital version of the Bahamian dollar that uses blockchain technology [9,10]. However, other central banks are exploring alternative approaches for implementing CBDCs, such as using a centralized database or a hybrid solution that combines centralized and decentralized elements. The choice of technological approach will depend on various factors, including the specific goals of the CBDC, the existing financial infrastructure, and the regulatory environment in which it will operate.

2.2. Cryptocurrency

Cryptocurrencies are digital or virtual currencies that use cryptographic techniques to secure transactions and control the creation of new units. Most cryptocurrencies are built on blockchain technology, which is a decentralized, distributed ledger that records transactions in a verifiable and permanent way [11]. A blockchain consists of a series of blocks that contain a set of transactions, and each block is connected to the previous one in a chain-like manner. Blockchain-based currencies are a revolutionary development in the world of finance and technology [12]. Transactions are validated by a network of users, who are rewarded with new units of the cryptocurrency for their efforts. Cryptocurrencies are decentralized, so they are not controlled by any central authority or financial institution, meaning that they are not subject to the same level of regulation as CBDCs or traditional fiat currencies. Public and private key cryptography is used to secure transactions, with public keys being used to receive payments and private keys being used to access and control a user's cryptocurrency holdings. Mining is the process by which new units of a cryptocurrency are created and added to the blockchain, with users solving complex mathematical problems to validate transactions. This process is computationally intensive and requires a significant amount of computational power, which is why some cryptocurrencies require specialized hardware (i.e. high-end GPU-equipped computer) to mine. Everything that is recorded on the blockchain is transparent and unchangeable, meaning that it cannot be altered by any means. Cryptocurrencies use consensus mechanisms to validate transactions and to ensure that the blockchain remains secure and tamper-proof. There are several different consensus mechanisms, including Proof of Work (PoW), Proof of Stake (PoS), and Delegated Proof of Stake (DPoS). Each mechanism has its own advantages and disadvantages, and the choice of mechanism can have a significant impact on the speed, scalability, and security of the network. Cryptocurrency wallets allow users to store, manage, and transfer their cryptocurrency holdings with different levels of security and convenience. Cryptocurrencies are often used as a means of exchange, and they can be bought and sold on online exchanges or traded peer-to-peer. There are many different cryptocurrencies, each with its own unique technical specifications and features. The transaction process of Bitcoin has been shown in Fig. 2 [13]. The key differences between the CBDCs and cryptocurrencies are shown in Table 1.

Table 1
CBDCs vs. Cryptocurrencies.

Digital Currencies	CBDCs	Cryptocurrencies
Issuance:	issued and backed by the central bank	created through a process called "mining"
Regulation:	same level of regulation as traditional fiat currencies	generally less regulated
Decentralization:	may use a private blockchain	operate on a public blockchain in a decentralized network
Use cases:	exchange and store of value	exchange, store of value, as a platform for decentralized applications etc.

3. How will e₹ work?

e₹ (e-Rupee or digital Rupee) aims to provide a simple, secure, and convenient payment system that can be used by all sections of society,

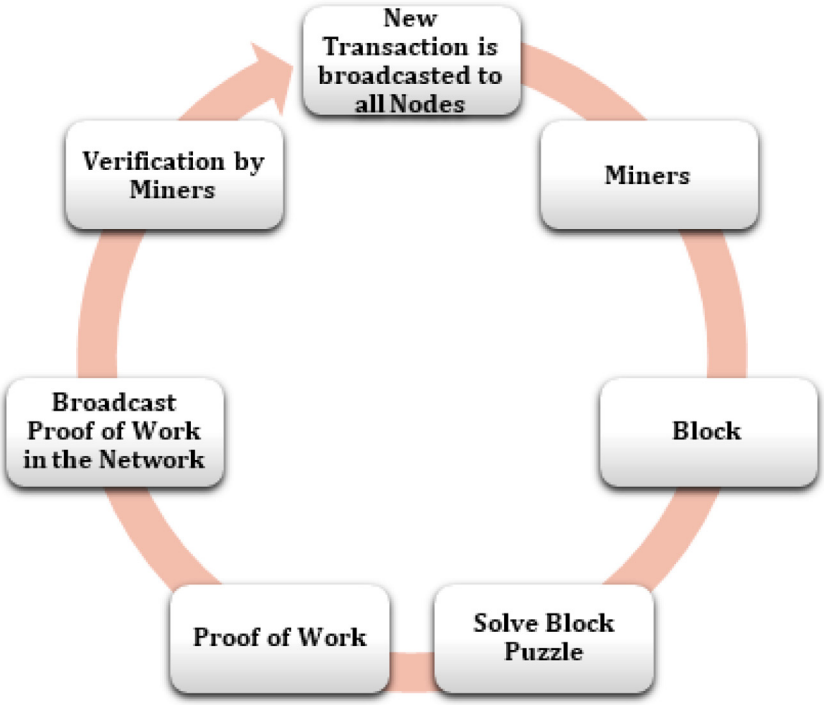


Fig. 2. Bitcoin transaction process.

Table 2
Token-based e₹ vs. Account-based e₹.

Token-based e-Rupee	Account-based e-Rupee
It is represented as a digital token on a blockchain.	It is not represented as a digital token on a blockchain.
It is a unique digital asset and can be transferred from one person to another through transactions.	It is not a standalone digital asset, but rather a balance that is associated with a user's account.
It can be exchanged for other cryptocurrencies or fiat currencies	It can only be used within the system in which it is issued, and it cannot be exchanged for other currencies.

including those who do not have access to traditional banking services. By promoting digital payments, e₹ is expected to help reduce the use of cash in the economy. Many people believe that e₹ is India's own version of cryptocurrency. However, the e-Rupee and cryptocurrency are not exactly the same. Like other CBDCs, the digital Rupee would be a digital version of physical cash and could be used in the same way as physical cash is used. It would be intended to be used as a means of exchange and store of value, similar to traditional fiat currencies. The e-Rupee will be given through the intermediate banks in the same denominations as coins and paper notes. Users have the option of purchasing digital Rupee through the designated banks, the official app, or the website. It is important to note that the digital Rupee is still in its infancy and that its functioning is not yet completely known. There are two different ways in which e-Rupee can be implemented namely Token-based e₹ and Account-based e₹. The differences between these two have been provided in Table 2.

It was made clear by the RBI that both Person to Person (P2P) and Person to Merchant (P2M) transactions are allowed. All transactions may adhere to one of the following models like other CBDCs [14].

(i) Direct Model: In this model, all transactions are processed by a central authority (RBI). The central authority is responsible for issuing e-Rupee, maintaining the ledger of all transactions, and ensuring that the e-Rupee supply is kept in check. It is also called the single tier model (Fig. 3).

(ii) Two Tier Model: It is also known as the indirect model (Fig. 4). In this model, all transactions are processed by both a central authority (RBI) and a network of decentralized nodes. The central authority is responsible for issuing e-Rupee and maintaining the ledger of all transactions, while the decentralized nodes are responsible for verifying and recording transactions on the ledger.

(iii) Hybrid Model: The hybrid model is a type of e-Rupee architecture that combines elements of both the single tier and two-tier models (Fig. 5). In this model, a central authority (RBI) is responsible for issuing e-Rupee and maintaining the ledger of all transactions, while a network of decentralized nodes is responsible for verifying and recording transactions on the ledger. The hybrid model is designed to provide the benefits of both the single tier and two-tier models, while minimizing their respective drawbacks.

The general procedures for generating and distributing a central bank digital currency (CBDC) backed by the Reserve Bank of India (RBI) are depicted in Fig. 6. The technologies used in CBDCs can vary depending on the design and implementation of the specific CBDC, but some common technologies that can be used include:

• Distributed Ledger Technology (DLT): CBDCs can be built on DLT platforms such as blockchain, which allows for decentralized and secure record-keeping of transactions.

• Smart Contracts: CBDCs can use smart contracts, which are self-executing contracts with the terms of the agreement directly written into code. This can help automate processes and reduce transaction costs.

• Cryptography: CBDCs can use cryptography to secure transactions and prevent fraud. Techniques such as digital signatures, hashing, and encryption can be used to ensure the authenticity and confidentiality of transactions.

• Application Programming Interfaces (APIs): CBDCs can use APIs to integrate with existing payment systems and infrastructure. This can enable seamless transactions between different payment systems and increase interoperability.

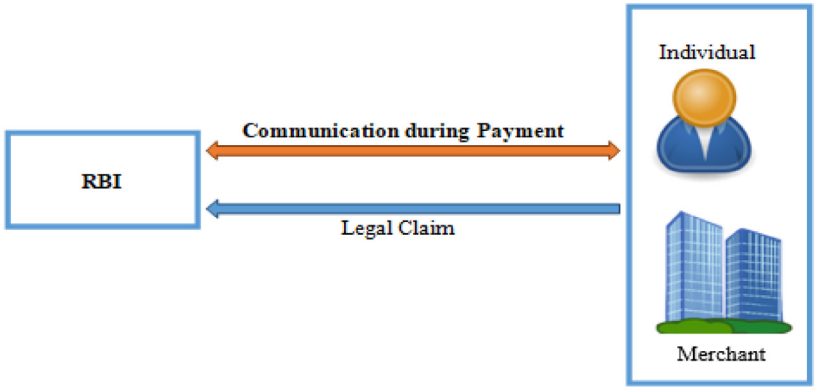


Fig. 3. Direct model.

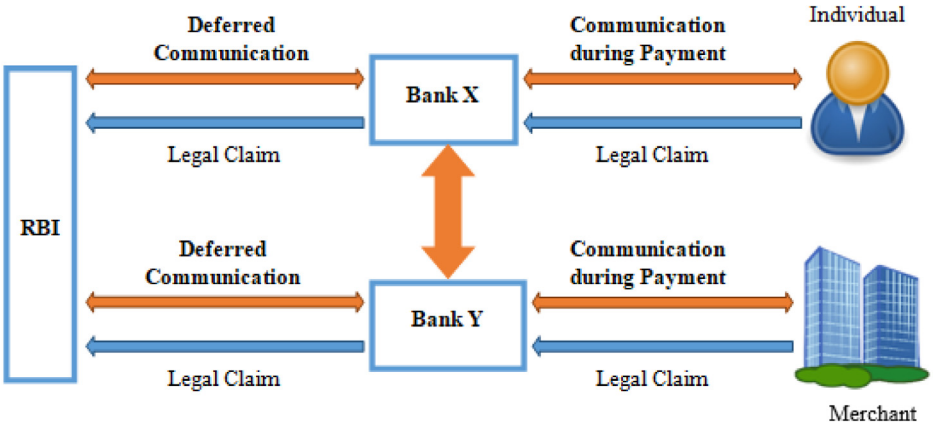


Fig. 4. Two-tier model.

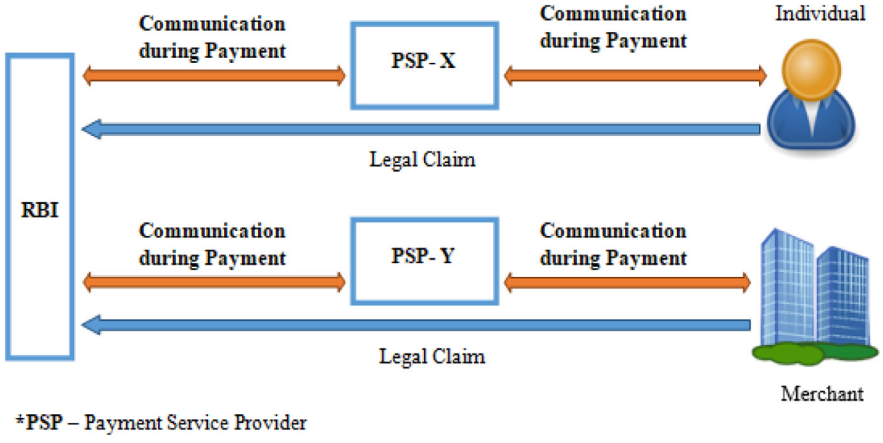


Fig. 5. Hybrid model.

• Mobile Wallets: CBDCs can be stored and transacted using mobile wallets, which can be downloaded as mobile applications. Mobile wallets can provide a user-friendly and accessible interface for CBDC transactions.

• Digital Identity: CBDCs can be linked to digital identity systems, such as biometric identification or national identification systems. This can help ensure that CBDC transactions are secure and authentic.

Overall, the technologies used in CBDCs are designed to provide a secure, reliable, and accessible digital currency that can be used for everyday transactions. By leveraging advanced technologies, CBDCs can offer many advantages over traditional fiat currencies, such as faster transaction speeds, reduced transaction costs, increased security, and greater financial inclusion.

4. Proposed e₹-Architecture

The specific architecture of an e-Rupee system will depend on a variety of factors, including the goals and requirements of the system, the technological capabilities of the system, and the regulatory environment in which the system operates. We propose an overall system architecture that can be split into two parts: a Private Blockchain network that runs the Reserve Bank and all licensed banks, and a Consortium Blockchain that runs the transactions between customers,

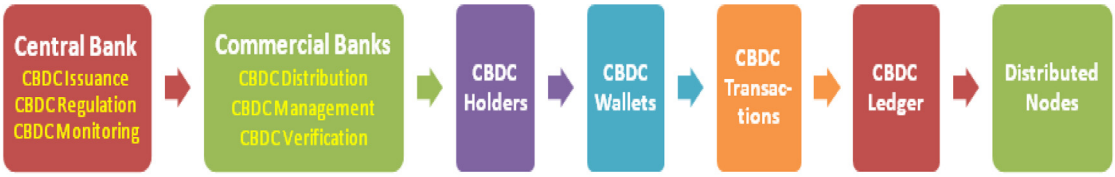


Fig. 6. CBDC process flow diagram.

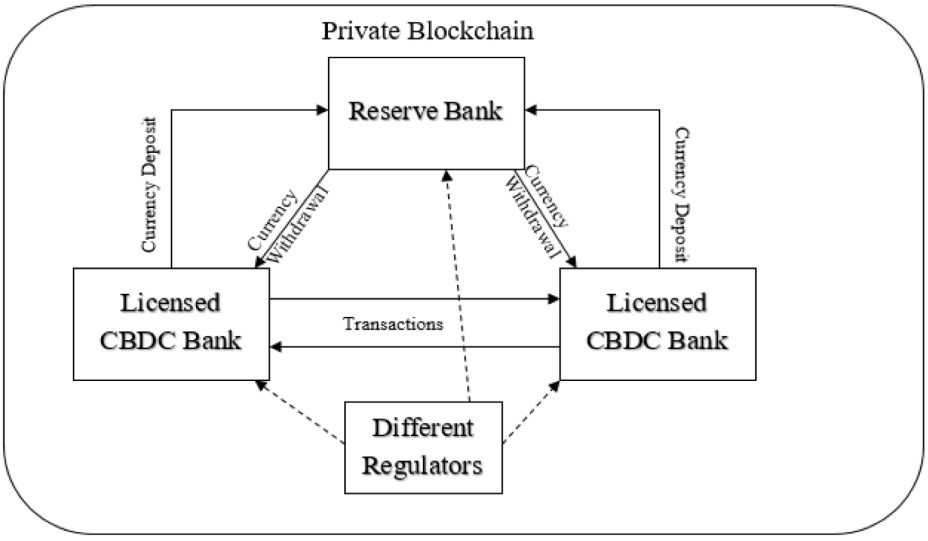


Fig. 7. Private blockchain.

account holders, businesses and licensed banks with or without third-party apps. Private blockchain and consortium blockchain are the two forms of blockchain networks that are intended for private, closed access.

4.1. Private blockchain

While the volume of transactions handled by licensed banks and the reserve bank is expected to be high, a private blockchain network that is more efficient is required. The nodes in a private blockchain are managed by a single company, and transactions can be processed swiftly and effectively. Private blockchain uses a consensus process that can be tuned for speed, as there are fewer nodes to reach consensus with. In the suggested concept, the Regulatory compliance agencies (SEBI, Income Tax Authority, CAB, and Enforcement Directorate) can also be included to a private blockchain for transaction monitoring and auditing. CBDC can be implemented using a private blockchain.

4.2. Consortium blockchain

Consortium Blockchain is a type of blockchain network where multiple organizations or entities come together to form a decentralized network. This type of network is suitable for digital currencies issued by central banks, known as Central Bank Digital Currency (CBDC), because it provides a balance between privacy, security, and scalability. Also, in the Indian economy scenario as there are multiple third party apps, and many licensed banks, creating a consortium blockchain network will provide a secure, privacy-proof network. All the digital applications need to have in-built operation interoperability to perform transactions between CBDC and UPI, NEFT, RTGS or instant money transfer.

The architectures of the private blockchain and consortium blockchain have been shown in Figs. 7 and 8 respectively. Private and consortium blockchain technology offer certain advantages over public blockchain technology, including:

• Privacy: Private and consortium blockchains offer greater privacy than public blockchains, as access to the network is restricted to authorized users. This can be particularly important for organizations that deal with sensitive or confidential data.

• Scalability: Private and consortium blockchains can be more scalable than public blockchains, as they do not require the same level of computational power to maintain the network. This can make them more cost-effective and efficient for certain types of transactions.

• Governance: Private and consortium blockchains offer greater control over the network, as they are governed by a single entity or a group of entities working together. This can make it easier to implement changes and updates to the network, and can also provide greater accountability.

• Flexibility: Private and consortium blockchains can be customized to meet the specific needs of the organization or consortium using them. This can make them more flexible than public blockchains, which may not be able to accommodate certain types of transactions or use cases.

• Compliance: Private and consortium blockchains can be designed to comply with specific regulatory requirements, which can be important for organizations operating in heavily regulated industries such as finance or healthcare.

5. Feasibility analysis

The Reserve Bank of India (RBI) is exploring the possibility of launching a digital version of the Indian Rupee. However, we can still discuss the theoretical and implementation feasibility of digital currencies in India based on available information and general trends.

5.1. Theoretical feasibility

Theoretically, a digital currency in India could provide several benefits, including:

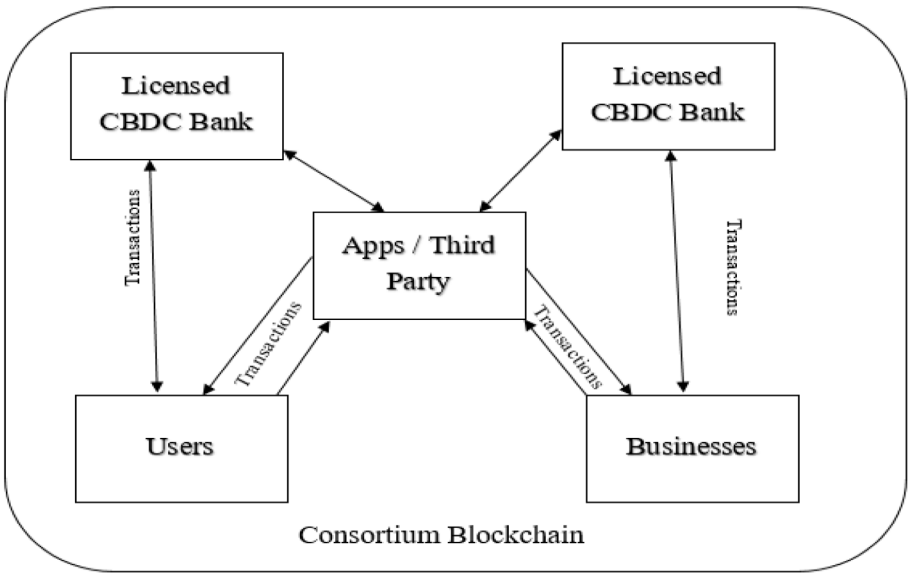


Fig. 8. Consortium blockchain.

- Increased Financial Inclusion: India is a country with a large population of unbanked individuals who do not have access to traditional financial services. A digital currency could provide these individuals with a low-cost and accessible means of transacting value, enabling greater financial inclusion.
- Reduced Transaction Costs: Digital currencies can potentially reduce transaction costs by eliminating the need for intermediaries such as banks and payment processors, which could lead to greater efficiency and lower costs for consumers and businesses.
- Improved Transparency: Digital currencies have the potential to increase transparency in financial transactions, making it easier to track money flows and prevent fraudulent activities such as money laundering.

5.2. Implementation feasibility

The implementation of a digital Rupee system in India will require a significant and reliable technical infrastructure and regulatory support that can handle large volumes of transactions and ensure the stability of the digital currency. Here are some of the key factors that would impact the implementation feasibility:

- Technical Infrastructure: Implementing a digital currency requires robust technical infrastructure that can support the transactional volume and speed required for a large-scale digital currency system. This would require significant investment in hardware and software, as well as cybersecurity measures to ensure the security and integrity of the system.
- Regulatory Framework: Digital currencies operate in a legal and regulatory environment that must be supportive and well-defined. India has a complex regulatory environment for financial services, and any digital currency project would need to operate within this framework while also addressing issues such as data privacy and security.
- Public Acceptance: The success of any digital currency project depends on the acceptance and adoption by the public. India is a country with a diverse population and varying levels of technological literacy, and any digital currency project would need to address these factors to ensure widespread adoption.

In summary, the theoretical benefits of a digital currency in India are clear, but the implementation feasibility will depend on a range of factors, including technical infrastructure, regulatory framework, and public acceptance. It remains to be seen whether the RBI will proceed with a digital currency project, but it is clear that any such project would require significant investment, planning, and regulatory support.

6. Comparison of e₹ and other CBDCs

CBDCs are designed to provide a secure, efficient, and reliable means of transacting digital currency, and to complement or replace physical cash in the economy. Several countries were exploring the possibility of launching central bank digital currencies (CBDCs). The People's Bank of China is one of the furthest along in developing a CBDC, known as the Digital Currency Electronic Payment (DCEP) [15,16]. The Central Bank of the Bahamas has already launched a CBDC in 2020 known as the Sand Dollar [9]. The Riksbank, Sweden's central bank, is exploring the potential launch of an e-krona [17], which is expected to be an account-based system, with users storing their e-krona in a digital wallet. It will be interesting to see how these systems continue to evolve and differ from one another. However, it is difficult to compare the specifics of different CBDCs, as each country will have its own unique set of circumstances, needs, and objectives. Table 3 provides some of the basic differences among e-Rupee and these three CBDCs.

7. Prospects of e₹

Digital Rupee could offer several potential advantages over physical currencies in India, especially in terms of increasing efficiency, promoting financial inclusion, and improving security and transparency in the payment system. e-Rupee is probably simpler, faster, and less expensive and will offer every transaction advantage available with other types of digital currency. It is essentially identical to banknotes. The digital Rupee has the potential to bring significant benefits to the Indian economy and society by reducing the reliance on cash, and modernizing the financial system. The following arguments support the notion that the digital Rupee is the currency of the future:

7.1. Centralized

The Indian government has expressed an interest in promoting the use of digital currencies, and has taken steps to support the development and adoption of e-Rupee. This could help to increase trust in e-Rupee and encourage its use. The government will recognize

Table 3
Comparison among different CBDCs.

CBDC	e-Rupee	Sand Dollar	E-Krona	DCEP
Launch Date	Pilot project launched on 1st December, 2022	October, 2020	Still in the development and testing phase, with no official launch dates yet.	Trial began in April, 2020
Technology	Centralized blockchain based system that follows a hybrid model.	Centralized blockchain based system that works on a two-tier model.	Centralized blockchain based system, two-tier architecture.	Centralized, permissioned blockchain based system, two-tier architecture.
Payment Mechanism	Prepaid digital currency	Digital version of fiat currency	Digital version of fiat currency	Prepaid digital currency
Accessibility	Available to anyone with a smartphone	Available only to residents of the Bahamas	Expected to be accessible only to Swedish residents	Available to anyone with a smartphone
Interoperability	Not interoperable with other digital currencies	Designed to be interoperable with other digital currencies	Expected to be interoperable with other blockchain-based currencies	Not interoperable with other digital currencies
Offline Capabilities	Expected to have an offline feature	Require internet connectivity	Require internet connectivity	Has offline payment feature
Privacy	Still evaluating privacy issues	Uses a privacy-enhancing technology called zero-knowledge proofs	Still evaluating privacy issues	The Chinese government has indicated that DCEP transactions will be traceable, which has raised some privacy concerns
Geographic Coverage	Not known	Within the Bahamas	Limited to Sweden	Intended to be used both domestically and internationally
Status	Still in the testing phase	Fully operational	Still in the testing phase	Rolled out in major cities including Shanghai, Chengdu and Beijing

the digital Rupee as entirely legal tender. The digital Rupee will not be completely decentralized like other cryptocurrencies; instead, the Reserve Bank of India (RBI) will control it. The RBI/Government has access to every transaction taking place on authorized networks.

7.2. Secure

A digital Rupee can potentially offer greater security compared to traditional physical money by leveraging advanced cryptographic protocols, multi-factor authentication, decentralized ledger technology, and reduced susceptibility to physical theft. It may also be integrated with Aadhaar, a biometric identification system used in India, which will enable users to receive payments directly into their bank accounts without the need for physical documents or signatures. In contrast to real currency, a digital currency's lifespan will be infinite because it cannot be physically damaged or lost. A digital currency leaves a digital trail that can be traced and audited more easily compared to physical cash. This can help prevent and detect fraud, money laundering, and other illicit activities. Therefore, the digital Rupee is expected to have a robust security system that includes the use of cryptography and a consensus mechanism that prevents double-spending and other fraudulent activities.

7.3. Ease of use

Digital currency can be used anytime and anywhere, without the need for physical cash or a physical bank. This can make transactions more convenient and efficient, especially for people who live in remote or rural areas. We do not necessarily need a bank account to use e₹. We may still purchase digital Rupees from the bank in the form of tokens. In general, it would be similar to a cash withdrawal from our bank account; but, instead of giving us cash, banks would credit our electronic wallets, allowing us to use it just like regular currency. The transactions via e₹ will provide real-time account settlements.

7.4. Global acceptance

Money transfers across borders and currency exchanges are time-consuming and costly. The fast cross-border money transfer is expected to improve bank cash management and operations with the introduction of the digital Rupee. NRIs who hold digital Rupee can utilize it for international financial transactions without regard to location. It will support the expansion of Indian economic endeavors.

7.5. Positive impact on economy

India has a higher cash inclination than the Nordic nations, including the UK and Australia, at 17% (the ratio of cash withdrawn to GDP) [18]. e-Rupee could help people become less reliant on cash. Naturally, it will reduce the expenses associated with managing, printing, and distributing physical currency. Furthermore, the use of a digital Rupee can also help to reduce the number of illicit transactions and the use of black money, which can increase tax revenue and reduce corruption. Finally, the implementation of a digital Rupee can provide the government with more data and information on spending patterns, which can be used to improve the effectiveness of economic policies and generate more revenue. Overall, the adoption of a digital Rupee can have a significant positive impact on the Indian economy and generate fiscal revenue in the long run.

The development of digital currency is the overarching trend in the modern age of electronic payment [19]. The CBDC is currently being studied by an increasing number of nations due to its numerous advantages [20]. India has also seen a significant increase in the use of digital payment options in recent years, and this trend is expected to continue. This could create demand for e-Rupee as an alternative digital payment option.

8. Challenges

Digital currencies, including e-Rupee, are still a relatively new and complex technology, and there may be regulatory challenges that need to be addressed in order for e-Rupee to be successful. India has not yet developed clear guidelines on the usage of digital currencies, and there have been recommendations to prohibit them outright. The adoption of a digital currency requires adequate digital infrastructure, education, and regulations to ensure its safety, reliability, and usability [21]. It is important to carefully consider the potential risks and challenges associated with a digital currency such as:

8.1. Digital illiteracy

In terms of digital literacy, India ranked 73rd in a list of 120 countries in 2021 [22]. The main reason is that there are many rural areas in India where high speed internet facilities are still not available. Therefore, people of those areas face the problems to avail the facilities of digital revolution. India must solve this issue in order to succeed in its objective to promote digital money.

8.2. Scalability issue

India has a vast population and a digital economy that is expanding rapidly. One major challenge is scalability, as networks can struggle to process large volumes of transactions simultaneously. This issue can be addressed through technological advancements and network upgrades. The architecture must be scalable and capable of handling enormous amounts of transactions and user accounts.

8.3. Privacy and security concern

The RBI keeps a central record of every transaction. Authorities may use centralized data for additional purposes. India is a nation with a high incidence of cyber attacks and a high level of cyber security risk. The introduction of digital currency may result in an increase in cyber attacks and the potential of digital thefts. Therefore, the cyber security threats will always be the major concern. The design must include powerful security features, such as multifactor authentication, encryption, and real-time monitoring and alerting.

8.4. Competition from other payment options

In terms of usability, support system, inventive mechanism and low transaction fees, e-Rupee will face competition from other digital payment options, such as bank-based digital payment systems and existing cryptocurrencies. India has a diversified population that speaks numerous languages. The architecture should handle different languages and provide an intuitive user experience for individuals who may not be skilled in English. India is a price-sensitive market, and excessive transaction fees are likely to dissuade users. To encourage acceptance and usage, the architecture should offer cheap transaction fees. India has a large number of unbanked and underbanked individuals, and incentive mechanisms could be utilized to promote the use of digital currencies. The architecture should include such methods as referral or transaction rewards.

9. Conclusion

India is a largely cash-based society, with a high percentage of transactions being conducted using physical currency. This can be problematic for a number of reasons, including the cost and time required for printing and distributing physical currency, the risk of counterfeiting, and the difficulty of tracking and taxing transactions. The RBI's e₹ – initiative essentially aims to replace traditional currency notes in wallets and may be used to send and receive payments via QR codes or through the respective parties' digital Rupee wallets. It could make it easier for people to make electronic payments and transactions, which could increase financial inclusion and lead to economic growth. e-Rupee will be exchangeable at par with current currencies, accepted as payment, and a secure place to keep wealth. However, there is a challenge of adoption, as the new mode of payment may not be readily accepted by the general public or traditional financial institutions. In order for e-Rupee to be successful, it will need to be accepted by merchants as a form of payment. If merchants are not willing to accept e-Rupee, it will be difficult for it to gain widespread adoption. Education and awareness campaigns may be necessary to overcome these challenges and promote adoption. The architecture should integrate with prominent payment gateways in India, such as digital wallets and UPI, to ease the exchange of digital currency for fiat cash. The government should come up with clear guidelines to boost up confidence among people. It would be important to ensure that the general public is educated about the use and risks of a digital currency, and that the necessary infrastructure is in place to support its use. Security of money would always remain the key obstacles. Efforts must be made to enhance the security of the network through cryptographic protocols and other measures. It is true that there are certain difficulties in employing digital currencies in India. Though many issues may be fixed in a short time, few require a long-term strategy. With the widespread use of digital currencies, India has a great chance to lead the world. e-Rupee could act as a catalyst for innovation, promoting rivalry and payment efficiency. Hopefully, this initiative will open up more discussion on the best course of action.

CRediT authorship contribution statement

Md. Asraful Haque: Equally contributed to the study conception and design and material preparation. **Mohd Shoaib:** Equally contributed to the study conception and design and material preparation.

Acknowledgment

The authors read and approved the final manuscript.

Funding

The authors received no financial support for the research, authorship and/or publication of this article.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Concept Note on Central Bank Digital Currency, FinTech Dept, Reserve Bank of India, 2022, Available online: https://rbidocs.rbi.org.in/rdocs/PublicationReport/Pdfs/CONCEPTNOTEACB531172E0B4DFC9A6E506C2C24FFB6.PDF.

[2] Satoshi Nakamoto, Bitcoin: A peer-to-peer ElectronicCash system, 2009, Available online: https://bitcoin.org/bitcoin.pdf.

[3] K. Verma, A. Jain, What is digital rupee? How is it different from cryptocurrency? in: Forbes Advisor, 2022, Available online: https://www.forbes.com/advisor/in/investing/digital-currency-in-india/.

[4] Future of Cryptocurrency in India Continues to Hang in the Balance, The Hindu, 2021, Available online: https://www.thehindu.com/business/future-of-cryptocurrency-in-india-continues-to-hang-in-the-balance/article34704676.ece.

[5] Embracing Cryptocurrency, The Hindu, 2021, Available online: https://www.thehindu.com/opinion/op-ed/embracing-cryptocurrency/article34824894.ece.

[6] Ministry of Finance, India – One of the Pioneers in Introducing CBDC, Press Information Bureau, 2022, Available online: https://static.pib.gov.in/WriteReadData/specificdocs/documents/2022/dec/doc2022121139201.pdf.

[7] BIS, Central bank digital currencies: foundational principles and core features, Available online: https://www.bis.org/publ/othp33.pdf.

[8] Tao Zhang, Zhigang Huang, Blockchain and central bank digital currency, ICT Express 8 (2) (2022) 264–270.

[9] PROJECT SAND DOLLAR: A Bahamas Payments System Modernisation Initiative, Central Bank of The Bahamas, 2019, Available online: https://www.centralbankbahamas.com/viewPDF/documents/2019-12-25-02-18-11-Project-Sanddollar.pdf.

[10] S.L.N. Alonso, M.A.E. Fernandez, D.S. Bas, J. Kaczmarek, Reasons fostering or discouraging the implementation of central bank-backed digital currency: A review, Economies 8 (2) (2020) 41.

[11] X. Han, Y. Yuan, F.-Y. Wang, A blockchain-based framework for central bank digital currency, in: 2019 IEEE International Conference on Service Operations and Logistics, and Informatics, SOLI, Zhengzhou, China, 2019, pp. 263–268.

[12] A. Tapscott, D. Tapscott, How blockchain is changing finance, Harvard Bus. Rev. 1 (9) (2017) 2_5.

[13] S. Ghimire, H. Selvaraj, A survey on bitcoin cryptocurrency and its mining, in: 2018 26th International Conference on Systems Engineering, ICSEng, 2018, pp. 1–6, http://dx.doi.org/10.1109/ICSENG.2018.8638208.
[14] A. Carstens, Digital currencies and the future of the monetary system, Hoover institution policy seminar, Basel 27 (2021).
[15] Michael Gu, CEP: China's national digital currency overview, 2023, Available online: https://boxmining.com/dcep/.
[16] Tong Zhang, Impacts of digital currency electronic payment (DCEP) on China's banking system, advances in economics, in: Business and Management Research, Vol. 203, pp. 3242–3246, ICEMCI 2021.
[17] Gabriel Soderberg, Behind the Scenes of Central Bank Digital Currency Emerging Trends, Insights, and Policy Lessons, FINTECH NOTE/2022/004.
[18] Sangeeta Ojha, 10 Reasons why digital rupee is the future of money, mint, 2022, Available online: https://www.livemint.com/money/personal-finance/10-reasons-why-digital-rupee-is-the-future-of-money-11667878002029.html.
[19] D.G. Birch, The war over virtual money is real, J. Paym. Strategy Syst. 13 (4) (2020) 300–309.
[20] J. Zhang, et al., A hybrid model for central bank digital currency based on blockchain, IEEE Access 9 (2021) 53589–53601.
[21] E.V. Sinelnikova-Muryleva, Central bank digital currencies: Potential risks and benefits, Voprosy Ekonomiki (4) (2020) 147–159.
[22] Tanushree Basuroy, Internet literacy index in India 2021, by category, 2022, Available online: https://www.statista.com/statistics/1232343/internet-literacy-index-by-category-india/.

BenchCouncil Transactions on Benchmarks, Standards and Evaluations 3 (2023) 100105

Contents lists available at ScienceDirect

BenchCouncil Transactions on Benchmarks, Standards and Evaluations

journal homepage: www.keaipublishing.com/en/journals/benchcouncil-transactions-on-benchmarks-standards-and-evaluations/

Case Report

ChatGPT for healthcare services: An emerging stage for an innovative perspective

Mohd Javaid [a,*], Abid Haleem [a], Ravi Pratap Singh [b]

[a] Department of Mechanical Engineering, Jamia Millia Islamia, New Delhi, India
[b] Department of Mechanical Engineering, National Institute of Technology, Kurukshetra, Haryana, India

ARTICLE INFO

Keywords:
ChatGPT
Applications
Healthcare
Treatment
Education
Learning
Limitations

ABSTRACT

Generative Pretrained Transformer, often known as GPT, is an innovative kind of Artificial Intelligence (AI) which can produce writing that seems to have been written by a person. OpenAI created this AI language model called ChatGPT. It is built using the GPT architecture and is trained on a large corpus of text data to respond to natural language inquiries that resemble a person's requirements. This technology has lots of applications in healthcare. The need for accurate and current data is one of the major obstacles to adopting ChatGPT in healthcare. GPT must have access to precise and up-to-date medical data to provide trustworthy suggestions and treatment options. It might be accomplished by ensuring that the data used by GPT is received from reliable sources and that the data is updated regularly. Since sensitive medical information would be involved, it will also be crucial to consider privacy and security issues while utilising GPT in the healthcare industry. This paper briefs about ChatGPT and its need for healthcare, its significant Work Flow Dimensions and typical features of ChatGPT for the Healthcare domain. Finally, it identified and discussed significant applications of ChatGPT for healthcare. ChatGPT can comprehend the conversational context and provide contextually appropriate replies. Its effectiveness as a conversational AI tool makes it useful for chatbots, virtual assistants, and other applications. However, we see many limitations in medical ethics, data interpretation, accountability and other issues related to the privacy. Regarding specialised tasks like text creation, language translation, text categorisation, text summarisation, and creating conversation systems, ChatGPT has been pre-trained on a large corpus of text data, and somewhat satisfactory results can be expected. Moreover, it can also be utilised for various Natural Language Processing (NLP) activities, including sentiment analysis, part-of-speech tagging, and named entity identification.

1. Introduction

Using the chatbot Artificial Intelligence (AI) language paradigm, Open AI creates the ChatGPT communication tool. Generative Pre-trained Transformer (GPT) human-like alerts are available for various duties, including responding to inquiries, troubleshooting ChatGPT network issues, and producing original content. They can also translate between different languages. ChatGPT can learn from prior discussions and apply that learning to deliver appropriate answers to future questions, becoming a more effective chatbot over time [1–3]. ChatGPT recognises the general context of a query or conversation and creates detailed replies relevant to the subject. ChatGPT may be utilised in healthcare for various goals, from bettering patient experiences and assisting medical personnel in optimising healthcare procedures and revealing valuable information. It can provide a better healthcare solution which is helpful for medical care providers and patients' communication [4,5].

ChatGPT comprehends and reacts to various conversational inputs, such as queries, claims, and directives in healthcare. It can converse with patients naturally and human-likely, which is advantageous for chatbots, customer service agents, and digital assistants. By using Machine Learning (ML) algorithms and natural language processing (NLP) approaches, ChatGPT develops conversational proficiency. The model can analyse and forecast word sequences through word embedding and recurrent neural networks. ML is essential to effective AI, and Data must be continuously fed into chatbot neural networks [6,7]. When identifiable patient data is entered into ChatGPT, it becomes a part of the database that the chatbot will eventually utilise. Generative Transformers can build conversational AI apps like chatbots that can have genuine discussions with consumers and other stakeholders after being educated. ChatGPT can accurately respond to customer inquiries and frequently asked questions by analysing various written materials, including textual and spoken language [8,9].

When there are feeds in the training data, they may also be reflected in response to the appropriate question. ChatGPT is a learning model

* Corresponding author.
E-mail addresses: mjavaid@jmi.ac.in (M. Javaid), ahaleem@jmi.ac.in (A. Haleem), singhrp@nitkkr.ac.in (R.P. Singh).

https://doi.org/10.1016/j.tbench.2023.100105
Received 27 March 2023; Received in revised form 12 April 2023; Accepted 13 April 2023
Available online 20 April 2023

that can only react based on feed data. Due to ChatGPT's excellent scalability and efficiency, businesses of various sizes may use it for a reasonable price. It can provide more thorough dialogue than other technologies because of its capacity to evaluate both written and spoken language. It is accurate, with a low percentage of false positives and negatives. Chatbots, used in various businesses to offer customer support, answer questions, and carry out other duties, are the most effective use of ChatGPT [10–12]. ChatGPT must be adjusted and tested for tasks particular to mental health for applications to be developed successfully. Models may be better prepared to handle behavioural traits with the correct blend of unique data, hyperparameters, architecture, and algorithms. This process of fine-tuning is essential to ensuring that models are used morally and adequately [13,14].

ChatGPT assists in analysing customer data and segmenting customers according to their preferences, needs, and behaviours, enabling the marketing team to conduct more successful targeted marketing campaigns. With the help of its sophisticated machine-learning algorithms, it can overcome language barriers, significantly improving consumer experiences and expanding our worldwide reach. It may help companies strengthen their online presence by being accessible around the clock on websites and social media platforms to respond quickly to consumers' frequently asked questions [15–17]. With the help of its sophisticated algorithms, ChatGPT may increase sales and marketing efforts, automate repetitive processes, and boost customer service while also increasing productivity. Observing how ChatGPT affects the salesforce platform and spurs industry innovation as technology develops further will be fascinating. In healthcare, this is helpful in various applications because they can comprehend natural language inputs and provide human-like answers [18,19]. The main aim of this paper is to discuss the significant applications of ChatGPT in healthcare.

2. ChatGPT

Generative Pretrained Transformers (GPT) refer to systems that can comprehend and generate long strings of complicated concepts. ChatGPT is a natural language processing (NLP) model created by OpenAI that enables real-time discussions with an AI chatbot similar to a person's. It is built on the GPT architecture, a language model that uses unsupervised learning to produce writing that resembles a human's. ChatGPT gathers data from every source it can find, feeds it into a transformer model, maps the connections between the various pieces of data, and makes educated estimates about what text should be used in which circumstances [20,21]. ChatGPT and comparable technologies may be trained on organisational data to alter the industry as technology develops. They also provide excellent starting points for producing software and content, managing knowledge, enhancing consumer interactions, and improving employee experiences. Event planners, tutors, and virtual personal assistants are all potential future developments [22,23].

ChatGPT has the potential to be a game-changing innovation that would significantly enhance the way humans communicate with machines. It remains to be seen whether it will be utilised for good or harm, but one thing is for sure: it will probably significantly influence our lives in the years to come. It is crucial to consider the technology's potential advantages and threats as it continues to advance and develop and to guarantee that it is utilised morally and responsibly. The replies produced by Chat GPT are based on the data it was trained on. ChatGPT is one of the chatbot applications, a very potent language model that uses ML algorithms to simulate human–machine interface. The ChatGPT has various uses, including content creation, translation, and summarisation. It can produce human-like text in healthcare for better treatment and patient disease awareness [24–26].

3. Need for ChatGPT in healthcare

Although many elements of healthcare need connection with patients, it is only sometimes necessary for optimal treatment. By enhancing adherence to treatment regimens and offering more practical and accessible care, ChatGPT may enhance the care given by a human healthcare provider and improve patient outcomes [27,28]. ChatGPT would need to have its prediction powers restricted for use in healthcare. A transformer model will detect patterns in the training data and apply that knowledge during inference. Transformer models may hallucinate predictions in a medical summary because they are rewarded by identifying patterns and generating predictions based on them. Patients who reside in underserved or rural regions could struggle to meet a certified diabetes educator or other healthcare experts physically. These patients may be able to use ChatGPT to get help and knowledge from a dependable source, even if they cannot physically visit a healthcare centre [29,30].

Patients may experience anxiety and confusion as they adjust to their new diabetes diagnosis. Some patients could find ChatGPT a handy and approachable method to get information and assistance while figuring out how to manage their disease. For certain people to adequately manage their diabetes, more frequent or intensive assistance may be needed. These patients could access extra information and assistance via ChatGPT, which would help them better manage their conditions. It answers patient questions, enhancing happiness and lowering the need for human care. ChatGPT can generate interesting and relevant content using natural language processing, depending on the input and user preferences. It facilitates communication between patients, insurance providers, and healthcare professionals. ChatGPT may help provide timely access to pertinent healthcare information to the appropriate parties [31–33].

4. Research objectives

ChatGPT assists healthcare personnel with routine chores like report generation and transcription of medical records. Healthcare providers might save time and concentrate on other crucial duties, including patient care, by automating these procedures using ChatGPT. For instance, a GPT system may be taught to reliably and quickly transcribe patient medical records, freeing up medical staff to spend more time dealing with patients and delivering treatment. It may lessen the possibility of medical record inaccuracies, which might harm patients. To make medical reports and other documents like clinical trials more understandable for patients and healthcare professionals, ChatGPT may be used to summarise them. ChatGPT may translate medical texts from one language to another, facilitating communication and aiding in comprehending crucial information between patients and healthcare providers [34–36]. The primary research objectives of this review-based article are as under:

RO1: - To brief about ChatGPT and its need for healthcare;

RO2: - to study the significant Work Flow dimensions of ChatGPT for the healthcare sector;

RO3: - to brief the typical features of ChatGPT towards the Healthcare domain;

RO4: - to identify significant applications and limitations of ChatGPT for healthcare.

5. Significant work flow dimensions of ChatGPT for the medical sector

The several distinguished dimensions related to chatGPT structure towards the solicitations in the medical domain are highlighted in Fig. 1. It further reflects the patient-related criteria, services and facts, database traits, workflow progress stages, etc. To process the ChatGPT working structure, a streamlined flow of information and knowledge is a must. Fig. 1 exemplifies the different working and progressive steps of the ChatGPT system for supporting the routine needs of the social structure. Further, the database gets sampled, and the process gets concluded by determining the reward model and updating dates the

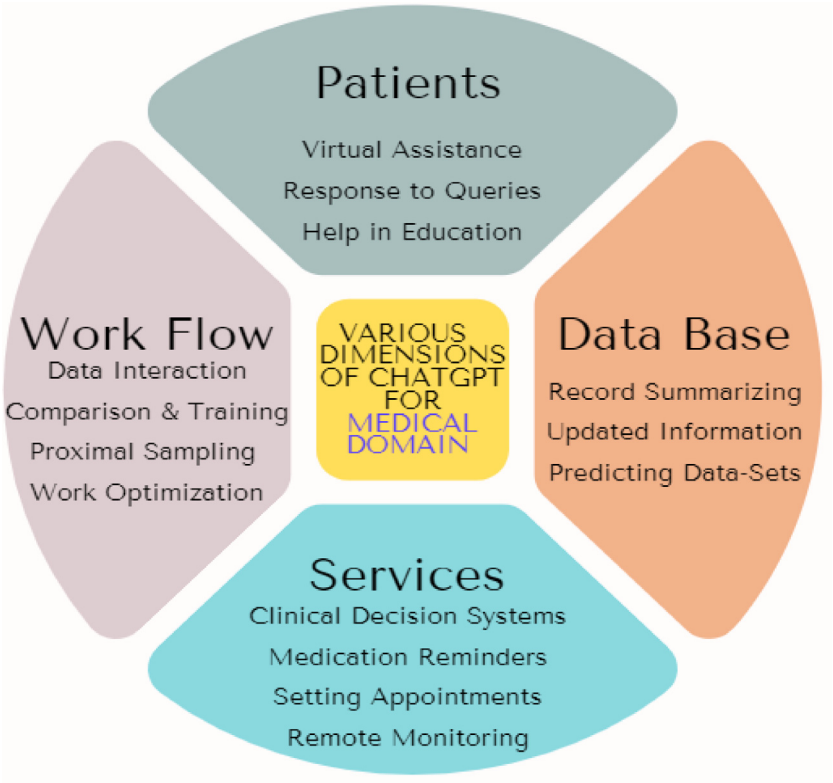


Fig. 1. Associated dimensions of ChatGPT framework for medical domain.

same in the cloud data set [37–39]. Various four associated dimensions are highlighted and discussed with the help of Fig. 1.

Making use of ChatGPT for the creation of clinical decision support systems is one possible use for healthcare. These programmes may review patient information and provide suggestions for treating pain and other ailments. For instance, ChatGPT could examine a patient's medical background, vital signs, and other information to recommend the best anaesthesia or dose. Ensuring they get the best treatment may enhance patient safety and results. Delivering pre-operative instruction is another possible use of ChatGPT in anaesthesia. ChatGPT may be used to provide individualised, evidence-based information to patients who may have questions or concerns regarding their scheduled operation. The treatment of post-operative pain and other symptoms may also be aided with ChatGPT. Personalised pain management advice, for instance, might be given via ChatGPT based on a patient's medical background, level of pain tolerance, and other variables. This could ensure that patients get the best treatment for their unique needs [40–42].

Medical education is another area that is anticipated to see a substantial influence. ChatGPT is a potent tool for promoting learning since it is an AI language model that can comprehend and react to spoken language. Students may get a more profound comprehension of challenging ideas using ChatGPT's interactive learning environment. Students may get immediate feedback, ask questions, and conduct more exciting and individualised subject exploration utilising conversational AI. ChatGPT's advancement in AI has been intriguing and has the potential to change how we interact with technology [43,44] entirely. ChatGPT is a highly accurate AI-powered chatbot that can precisely comprehend natural language and respond to user questions. One of its main advantages is that this technology can provide replies to cues it has yet to be exceptionally trained on. It makes a helpful tool for building chatbots that can have natural conversations with people since it can handle a variety of subjects and circumstances [45,46].

Chatbots for customer support are one possible use for ChatGPT. A ChatGPT-powered chatbot might respond to ranging consumer questions thoroughly and accurately, freeing human customer care professionals to address more complex problems. A chatbot built to deliver information about a specific medical condition or treatment might utilise ChatGPT to create replies in addition to its text creation capabilities. It might also provide replies for a computer programme that aids patients in managing their treatment, such as a virtual assistant who reminds them to take their prescription or gives them details about their health state. In essence, we advise users to utilise the tool as a source of creative inspiration, to generate ideas, and to act as a springboard for later-on, mostly human works. In other words, ChatGPT may provide many suggestions that combine existing concepts and may lead us to a place we would never have considered on our own [47,48].

ChatGPT is a valuable tool for connecting with persons who may not speak the same language or have the same communication preferences since it employs Natural Language Processing, which is taught to interpret and comprehend natural language. ChatGPT has certain clear benefits over conventional customer service methods, including the ability to provide real-time assistance around-the-clock, allowing users to get assistance more quickly. It can provide clients with a more personalised experience by promptly replying to queries, processing several conversations simultaneously, and saving time by understanding linguistic complexity. ChatGPT delivers the precise answers to their questions in detail so that they may know everything about their query, and it is affordable and presents correct answers to our inquiries [49,50]. By employing ChatGPT to gather user input, Open AI aims to make AI systems more natural and safer to engage with Students, professors, and scientists may utilise ChatGPT for writing since it can respond to inquiries and create a document on a subject based on the compilation of material accessible online.

The purpose of ChatGPT is to make our life simpler. ChatGPT has been a ground-breaking innovation that supports these conversations. With this AI chatbot's robust model and research-based learning capabilities provided by the ChatGPT, people can even converse with a bot and get a humanised response back. ChatGPT can completely change how we produce, distribute, and use instructional information in the context of eLearning. Every topic may be researched and learned about by lone learners using ChatGPT. Self-directed learning of this kind may

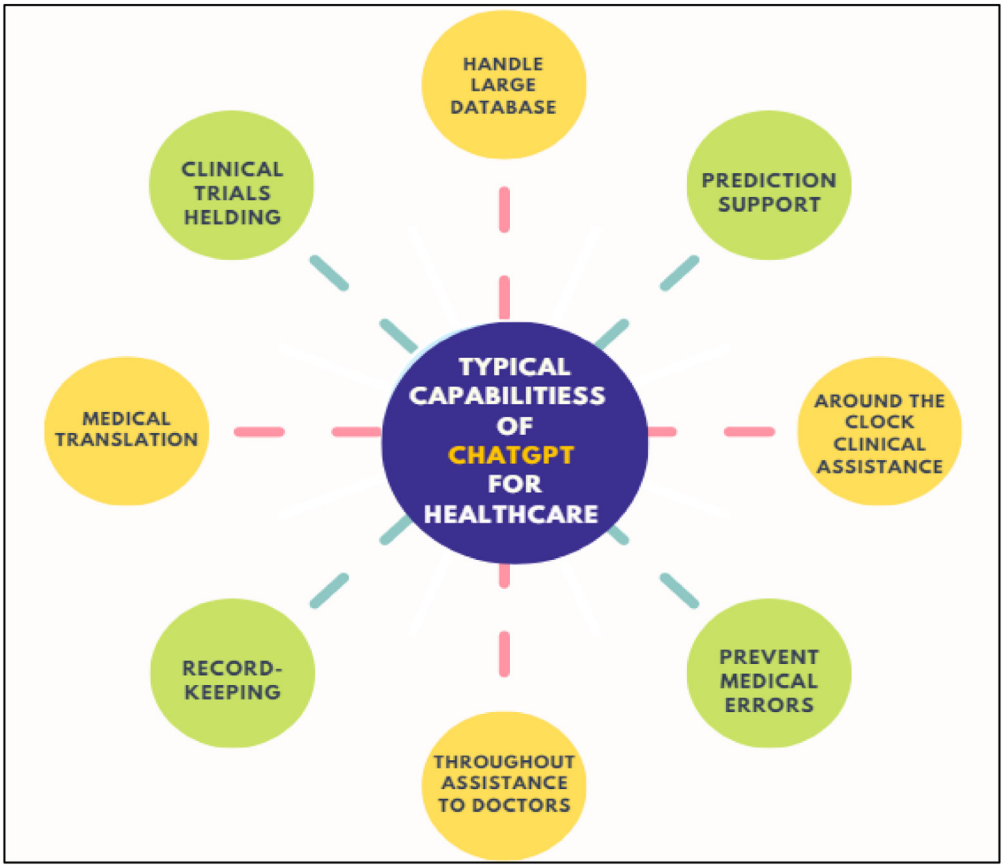


Fig. 2. Typical capabilities of ChatGPT for the healthcare sector.

be very beneficial for enhancing one's knowledge and abilities. Using ChatGPT, medical students may ask questions, get rapid feedback, and get solutions suited to their requirements. The use of ChatGPT may aid students in understanding and remembering what they are learning [51–53].

ChatGPT technology can completely transform eLearning and the process of developing educational materials. It allows training companies to create, evaluate, and change material to ensure it is instructional valid and satisfies business and learning goals. It also enables learners to interact directly with the model to improve their knowledge and abilities. It may help treat post-operative pain and other symptoms and create clinical decision support systems and pre-operative education. While ChatGPT can potentially enhance patient care, it is crucial to consider its limits and utilise it as an addition to, rather than a substitute for, human knowledge and discretion [54,55].

6. Typical features of ChatGPT towards healthcare domain

Fig. 2 explores the various associated typical capabilities, features, and applications of ChatGPT support for the healthcare sector. It includes the features like remembering aspects, prediction support, medical translations, etc. Apart from these different features and capabilities, various limitations have been observed, such as; the sometimes generation of incorrect information that may arise with biased content, etc. [56,57]. In addition to this, several associated other characteristics and classical perspectives of ChatGPT are further represented and elaborated in Fig. 2.

The development of virtual assistants for patients is one instance of how ChatGPT might be used in medicine. These assistants may provide individualised suggestions and counsel based on the patient's medical history, present symptoms, and other pertinent information. For instance, a virtual assistant may advise on managing a chronic ailment like diabetes or advise over-the-counter drugs or home cures for a patient suffering from the flu or cold. Many platforms, including websites, smartphone applications, and voice assistants, may be used to access these virtual assistants. As people may get individualised suggestions and guidance without seeing a healthcare practitioner physically, this can be constructive for patients residing in remote regions or needing help obtaining healthcare. In order to find novel drug targets and develop fresh ideas, ChatGPT may be used to evaluate a lot of scientific material, including research articles and patents. For drug development, this technology is used to train the model on a large body of scientific literature before using the model to provide fresh hypotheses or recommendations for more studies [58,59].

In specific ways, ChatGPT as a language model is a ground-breaking innovation in healthcare, especially in terms of its capacity to comprehend and produce text on various subjects with excellent accuracy. ChatGPT has the potential to be used in a variety of healthcare applications, including automating time-consuming tasks like report generation summaries and note taking, which can save time and improve efficiency; helping patients with symptom-checking, medication management, and appointment scheduling; and supporting patient education, compliance, and self-management of chronic conditions. ChatGPT can be employed in the healthcare industry for several activities, including patient contacts, clinical trial analysis, medication research, and medical recordkeeping. In some ways, ChatGPT's evolution is similar to that of the web browser [60,61]. While the internet existed long before the web browser, it was made more widely accessible by the latter. Similarly, ChatGPT makes AI more approachable by offering a straightforward and user-friendly conversational interface.

The ability to be linked to a variety of platforms, including websites, chatbots, and mobile applications, is another benefit of ChatGPT. This enables a user-friendly user experience and a more seamless technology integration into current systems. For instance, a chatbot coupled

with ChatGPT may respond to customer inquiries, suggest products, or help customers complete transactions. In addition to customer service, ChatGPT has the potential to be employed in several other sectors, such as education and finance. For instance, ChatGPT might be used in the healthcare industry to provide people with individualised information and assistance, such as responding to inquiries about their symptoms or helping them locate a doctor. ChatGPT might be used in the classroom to provide students with individualised coaching and assistance, enabling them to study more interestingly and dynamically. ChatGPT's potential to take the role of human workers in various fields, including journalism and customer service, contributes to its current popularity. Humans can teach the model to carry out previously performed jobs by humans, and they can do so much more quickly and accurately [62–64].

ChatGPT is profoundly influencing the writing sector, as many authors and content producers use the tool to either spark new ideas or enhance their existing works. It is more likely to be used to support and supplement human authors' writing and content-creating efforts. It may give employees more time to concentrate on higher-value jobs that require greater creativity and problem-solving if accepted and utilised correctly. Practical usage of ChatGPT requires unique expertise. Instead of being concerned about how it will affect our work, we must comprehend it and educate ourselves fully. The applications for ChatGPT are many and diverse, ranging from language translation software to chatbots for customer support [65,66]. ChatGPT may also help with content sourcing and curation from various sources, aiding businesses in creating a unified and successful content marketing strategy.

ChatGPT is a natural language processing technology with AI that enables conversational chatbots. Asking the language model for assistance with tasks like composing emails, articles, and code is possible. The basis for medical guidance and treatment is high-quality evidence. In the age of democratic healthcare, patients and clinicians utilise a variety of channels to obtain data that influences their choices. However, at this stage of its development, ChatGPT may need to be sufficiently resourced or set up to provide accurate and objective information. Developers may work on chatbots and voice-based apps using ChatGPT. Developers may evaluate the responsiveness and correctness of their apps in real-time by simulating user interactions. Companies might use ChatGPT to propose products or to provide details about impending deals or promotions. Sales may be boosted, and consumer involvement increased in this way. The ChatGPT programme may also be a personal assistant to assist users with various chores. For instance, a chatbot may generate emails, schedule appointments, alert users about impending deadlines or key events, or even assist with housework [67,68].

7. Typical ChatGPT applications for healthcare

ChatGPT offers support for healthcare providers, which can help reduce wait times and improve patient satisfaction. This can include patient inquiries regarding insurance, billing, and appointments to provide them with the necessary information. Healthcare practitioners needing help in making informed patient care choices might utilise ChatGPT as a clinical decision-support tool [69,70], but with caution. Healthcare workers may use this technology to learn about treatments, drugs, and diagnostic techniques and get help on what to do next. This technology can automate specific processes, increase efficiency in the healthcare industry, and even replace some jobs. Remembering that AI may improve human skills in the healthcare industry and open up new career prospects is also crucial. By using AI in healthcare, professionals may spend less time on routine chores and more on more challenging and valuable duties like patient care, counselling, and cooperation with other healthcare practitioners. ChatGPT may help raise patient safety, lower mistakes, and enhance treatment quality. ChatGPT can be a game-changer because of its extraordinary fluency and inventiveness [71–74]. The significant applications of ChatGPT for the healthcare sector are identified and briefly presented in Table 1.

ChatGPT is anticipated to provide support in many ways to improve healthcare services' sustainability and error-free nature. AI can completely transform healthcare management by simplifying interactions, automating time-consuming chores, and creating more precise and effective procedures for patients and doctors. This helps carry out operations that require human intellect, such as comprehending and interpreting linguistic signals, forming judgements, or learning from data. An AI system with a language model is created primarily to process and comprehend natural language. Language models may create text or reply to user input in a manner that sounds natural and human-like because they are trained on massive datasets of text and utilise ML techniques to understand the patterns and structures of language. ChatGPT can write creative sonnets and tales in their distinctive style, narrate historical events in a famous person's voice, and correct computer programming instructions [75–77].

8. Discussion

ChatGPT can reply to various themes and discussions since it was trained on various text materials, including books, papers, and web pages. ChatGPT may be used as a language model for various things, such as chatbots, question-and-answer platforms, and language translation. The capacity of Chat GPT to produce superior, human-like replies to text prompts garnered it great recognition and appeal, even though it is simply one of several AI technologies created by OpenAI. It has been used in several settings, such as social networking, educational applications, and customer service. Several physicians have tested ChatGPT to determine whether the AI-based chatbot can assist with doctors' routine chores. ChatGPT may provide patients with round-the-clock assistance with lesser human involvement. This technology can carry out many challenging activities for managing complex medical and clinical data. It might aid in describing and quantifying human–AI interactions and standardising experimental techniques.

ChatGPT may soon be widely used in clinical practice, with various applications in almost all medical specialities, such as patient communication and clinical decision assistance. Clinicians began experimenting with ChatGPT as a result of its outstanding success. Customer support, marketing, education, and entertainment are used for ChatGPT. They may be used to provide individualised customer service, such as making product suggestions specific to the consumer's needs or replying to their enquiries. Moreover, they might be used for marketing activities like producing material on social media or offering automated customer service. ChatGPTs may be used in the classroom to support student learning by giving them individualised feedback or direction. Compared to conventional rule-based chatbots, it has been trained on a vast corpus of text data, enabling it to provide more correct and coherent responses.

Several enquiries may be handled simultaneously, and rapid and effective solutions can increase user experience and patient satisfaction. ChatGPT might be a cheaper alternative to employing human customer support agents for businesses trying to enhance customer service. It has access to a wealth of knowledge and information on various subjects since it has been trained on a significant quantity of text data. It can create content, such as summaries, articles, and product descriptions, which increases productivity by obviating the requirement for human content generation. It is a helpful tool for corporations and organisations with a worldwide presence since it can translate text across languages. ChatGPT may be used to deliver information and answers on various subjects. It may be used to assess and classify the sentiment of text data, giving companies and organisations valuable insights [78].

The development and use of Chat GPT are still in their early phases, and over the following years, it is anticipated to continue to advance quickly. Adoption will rise as more people learn about its possibilities and use it, resulting in more remarkable advancements and inventions. ChatGPT and other technologies promise to boost productivity, communication, and efficiency at work and elsewhere. By delivering prompt

Table 1
ChatGPT applications for the healthcare domain.

S. No	Applications	Description
1.	Educate patients	• Patients may utilise ChatGPT to educate themselves on their health and problems, giving them the knowledge, they need to decide on their treatment. • This may respond to patients' queries and provide them with details on procedures, drugs, and dietary adjustments; rule-based expert systems and knowledge graphs are utilised for activities including diagnosis, therapy, planning, and drug development. • In specialities, including radiology, pathology, and ophthalmology, AI-powered medical imaging systems are employed for tasks like picture categorisation, segmentation, and diagnosis. • AI-powered robots are employed for procedures, including surgery, physical therapy, and patient monitoring. • ChatGPT in healthcare can increase the efficacy and efficiency of medical treatment. • One of the possible issues with using AI in healthcare is that it can dehumanise it, putting more emphasis on data and algorithms than the needs and values of patients and healthcare professionals. • The tool's capacity to produce lines of code is another characteristic that software professionals will undoubtedly value greatly. • If the users accurately articulate the issue statement, ChatGPT can produce a code snippet. • ChatGPT's bot can convert complex technical topics into clear and understandable words for everyone.
2.	Clinical studies	• ChatGPT may be utilised in clinical studies to assist with data gathering • The chatbot can help people to provide information about clinical trials. • A team of specialists, comprising data scientists, engineers, healthcare workers, and ethicists, are needed to develop and deploy AI models in the healthcare industry, which might create new jobs. • It is crucial to remember that any possible job displacement brought on by medical AI should be minimised by offering chances for healthcare employees to get new training and skills and establishing new jobs in fields like data science and AI research and implementation. • By offering several methods of articulating the exact words, ChatGPT may aid in interpreting patients' open remarks. • In order to establish a conversational mechanism for patients to communicate their symptoms in English, we could utilise the public version of ChatGPT to learn terms that potentially map to symptoms.
3.	Monitor patients remotely	• Medical personnel may remotely monitor patients using ChatGPT to maintain tabs on their health. • Patients may be reminded to check their vital signs by the chatbot, and they can alert medical personnel if anything changes or causes them to worry. • Those searching for help with routine business chores like composing emails, producing reports, and establishing other communications find ChatGPT increasingly popular. • In order to provide appropriate output, it uses AI to learn from prior encounters and sample texts. Because of this, creating material can be done quickly and easily. • The creation of conversational and interactive chatbot experiences is also possible with ChatGPT. • Anyone may build chatbots with an understandable conversation flow that responds to natural language questions by utilising ChatGPT. • This is accomplished by combining domain-specific training with OpenAI's GPT language model.
4.	Accessing information regarding health	• In order to obtain instant access to information about their health and problems, patients may utilise ChatGPT as a virtual assistant. • The chatbot will provide accurate and up-to-date replies to queries from patients about their symptoms, medical procedures, and prescription medications in different languages. • Text in various forms, including articles, tales, and social media postings, may be generated using ChatGPT. • ChatGPT may be used to summarise content in many languages and translate text from one language to another. • ChatGPT can classify and categorise text by determining the tone of a social network post. • It may pull the most critical details from a lengthy text and show them abbreviated. • Thus, to create chatbots that can comprehend and react to natural language input from patients and carers, ChatGPT may be utilised. • These chatbots can provide symptom checking, triage, scheduling, and other fundamental healthcare information. • ChatGPT is also for quizzes, language translation, paragraph production, etc.
5.	Medical suggestions and counselling	• ChatGPT can provide patients with medical suggestions and counselling based on their symptoms and medical background. • The chatbot can help patients make educated choices regarding their health and provide them with peace of mind. • A chatbot that employs ChatGPT to summarise patient medical data for healthcare professionals, including information on diagnosis, treatment, and progress. • For healthcare workers, patients, and researchers, a chatbot that employs ChatGPT for medical language translation may translate medical terminology and phrases from one language to another. • A chatbot that leverages ChatGPT to help people and medical professionals by reporting adverse events associated with medications and other items. • The chatbot may help users complete required forms and documents, walk them through the reporting process, answer questions regarding the event and the reporting requirements, and provide general information about it. • Using ChatGPT, the clinical safety chatbot can help medical practitioners to report adverse events connected to clinical trials, such as adverse responses, significant adverse events, and unforeseen issues.
6.	Schedule appointments	• ChatGPT allows patients to schedule appointments with doctors, making getting the treatment they need at a convenient time easier. • Patients may ask the chatbot for a list of available appointment slots, which might help them choose a time that works for them. • The ChatGPT model must be trained and tuned by professionals knowledgeable in ML methods and methodologies. • For the chatbot to utilise an extensive collection of medical data and queries to answer user input, data scientists or analysts with expertise in data analysis methods, including data cleansing, data visualisation, and statistical analysis, are required. • The chatbot must be implemented, integrated with other systems, and optimised for performance by software developers familiar with programming languages. • For the chatbot to have a user-friendly interface, such as a website or mobile app, that enables users to engage with the chatbot effortlessly, user interface designers with expertise in user interface design concepts are required.

(*continued on next page*)

Table 1 (*continued*).

S. No	Applications	Description
7.	Help to identify patient symptoms	• ChatGPT helps patients identify their symptoms and decide on the best action by acting as a symptom checker. • The chatbot may ask patients about their symptoms and provide them with a list of possible diagnoses and instructions on what to do next. • ChatGPT is a significant advancement in AI technology and has the potential to enhance how humans communicate with machines. • It will be crucial for academics and industry leaders to work together as the technology develops and matures to ensure it is used for the betterment of society. • ChatGPT undergoes rigorous training on vast volumes of data to understand linguistic patterns. • The procedure guarantees accuracy while speculating the following words in a string of words, where supervised and reinforcement learning techniques are used to train and improve ChatGPT. • Human specialists train the computer to make judgements that benefit people by providing plausible and moral replies. • The conversation capabilities of ChatGPT can be developed to answer follow-up inquiries, confess its errors, dispute faulty premises, and reject unsuitable requests.
8.	Medication reminders	• Patients may use ChatGPT to get medication reminders, encouraging them to take their prescriptions as directed. • Patients may get reminders on when to take their drugs and information about side effects and possible drug interactions from the chatbot. • It is possible to produce content rapidly and effectively using ChatGPT. Also, it may be utilised to provide distinctive, captivating content customised to specific audiences. • ChatGPT may improve personalisation; thus, examining user data may provide tailored suggestions for goods, services, and information. • Moreover, it may be utilised to develop virtual assistants that are catered to user requirements. ChatGPT's rapid access to knowledge and insights may revolutionise how organisations run. • Real-time analytics and suggestions may be provided with this technology and used in conjunction with business apps to assist organisations in making better choices and remaining competitive.
9.	Developing patient-specific treatment programmes	• Developing patient-specific treatment programmes is another possible use for GPT in the medical field. • A GPT-powered system may provide a personalised treatment plan by looking into a patient's medical history, present symptoms, and other characteristics, as well as the patient's particular wants and preferences. • Patients with unusual or complicated diseases that need specialist treatment may find this extremely helpful. • For instance, depending on a patient's medical history and other criteria, a GPT system may suggest a particular mix of drugs or treatments most likely successful for that patient. • This might lower the possibility of adverse responses or other issues and guarantee that patients get the best treatment possible for their circumstances. • It opens up new avenues for creativity and artistic expression by making it possible to produce creative material like music, poetry, and visual art. • It has several potential uses in fields including customer service, virtual assistants, and more because of its massive size and human-like replies. • Deep learning, reinforcement learning, natural language processing, and computer vision are just a few of the cutting-edge research specialities of OpenAI.
10.	Medical terminology and ideas	• To increase ChatGPT's comprehension of medical terminology and ideas, which may be used to extract structured data from unstructured materials like electronic health records (EHRs) and clinical notes, ChatGPT can be fine-tuned on medical texts. • A vast corpus of text data is used to train the Chat GPT using an unsupervised learning approach. • Pre-training involves teaching the model to recognise correlations and patterns in text data by foretelling the next word in a string of words. • When pre-training is finished, the model may be fine-tuned to perform a particular job, such as producing text or responding to inquiries. • In order to fine-tune a model, a smaller dataset relevant to the current job must be used. • Based on its training data, Chat GPT predicts the most probable set of words that will come after a prompt to produce a response. • Longer paragraphs of text and shorter replies may be produced using the approach. • Chat GPT combines attention techniques and language modelling to produce replies. • The attention mechanisms allow the model to concentrate on various sections of the input sequence while creating its answer.
11.	Digital assistant for doctors	• ChatGPT might be trained to serve as a digital assistant for doctors by using AI and ML. The system would collect crucial data from patient records, classifying information including test findings, family history, symptoms, and current medications. • Physicians can evaluate patient requirements more quickly now that this information is easily accessible by using AI. This feature enables a sharper focus on the crucial components of patient care. • Training ChatGPT to function as a virtual assistant for doctors is possible. • From patient records, it may extract essential data that can be used to populate reports with information such as family history, symptoms, present medicines, allergies, test results, etc. • By succinctly putting the information at their fingertips, doctors would have more time to visit patients and concentrate more on patient care. • Chat GPT might be tailored for specific sectors or use cases to serve those customers' requirements better. • The ability to handle other languages would increase Chat GPT's usability for users all across the globe. • Chat GPT might be improved better to comprehend a conversation's or task's context, resulting in more detailed and pertinent replies. • For the demands of organisations of various sizes, from tiny companies to massive corporations, Chat GPT may be scaled up or down. • There are several chances for innovation and cooperation in the future for ChatGPT.

(*continued on next page*)

Table 1 (*continued*).

S. No	Applications	Description
12.	Enhance communication	• The capacity of ChatGPT to enhance communication, where the natural language processing tool can comprehend and react to patient enquiries and concerns, is one of the key advantages of utilising ChatGPT in healthcare. • This lessens the burden on healthcare personnel. • As healthcare organisations ramp up their digitalisation, ChatGPT will be a resource for everything from patient communication to automating back-office tasks. • The business uses ChatGPT to create and analyse customer surveys to learn more about consumer requirements and preferences. • The extensive survey data could be analysed using ChatGPT, which provided insightful information on consumer behaviour and industry trends. • Businesses are employing ChatGPT to examine competitor data and provide insights into their market positioning and business plans, as a result, they could keep one step ahead of the competition and confidently decide on their marketing tactics. • To manage client enquiries and provide real-time information on the shipment status, delivery dates, and other relevant factors, ChatGPT may be coupled with chatbots or messaging services.
13.	Assistance to diabetic patients	• ChatGPT may be used to provide rapid and individualised assistance to diabetic patients. • A trained diabetes educator is often the ideal person to answer these concerns or questions since many persons with diabetes may have them. • It might be challenging for educators to be accessible at all times to respond to these queries. By giving patients a chance to get assistance and knowledge even when their educator is not physically there, ChatGPT may help close this gap. • Moreover, ChatGPT may create individualised meal plans and locate recipe suggestions based on a patient's unique health requirements and preferences. • It may be used to assist in the resolution of specific issues or to motivate patients to adopt a healthy diet and lifestyle. • AI-powered chatbots can provide round-the-clock customer service and quickly respond to client enquiries. • Chatbots may help compile insightful data about prospective diabetic patients and provide data-driven insights to help patients, thereby allowing them to customise doctors' strategies.
14.	Rapid access to medical information	• ChatGPT's rapid access to medical information has the potential to enhance healthcare. • Also, physicians and nurses can access the most recent findings and treatment suggestions by incorporating the language model into healthcare apps. • With ChatGPT, developers may create code more quickly and effectively using code snippets generated based on specific programming languages and paradigms. • It is crucial to remember that this technology cannot replace physical work, soft skills, or the development and maintenance of relationships. • ChatGPT can respond to follow-up queries, acknowledge errors, refute false assumptions, and reject unsuitable requests, but the final word has to be from the doctor.
15.	Help doctors reply to insurance claims.	• ChatGPT might help doctors reply to insurance claims much more quickly. • A doctor may utilise an AI tool to create a response and rapidly update it before submitting it, saving them time and effort. • Early adopters have been utilising ChatGPT to help with tedious activities, including drafting sick notes, patient letters, and letters requesting payment from medical insurance for specific pricey patient prescriptions. • Chat GPT is developed using a vast amount of online content. • It tries to mimic human writing and may serve several functions in medical care and scientific study. • This necessitates that data be continuously fed back into chatbot neural networks. • When identifiable patient data is entered into ChatGPT, it becomes a part of the database that the chatbot will eventually utilise. • Doctors and their employees may spend less time negotiating insurance approvals, creating recommendation letters, and dealing with claim rejections. • By providing a template and a set of instructions, ChatGPT can produce code rapidly and precisely.
16.	Individualised health advice and suggestions for patient care	• AI-powered chatbots may provide individualised health advice and assistance in identifying medical issues. • Moreover, AI-driven chatbots may help patients have a better healthcare experience by automating administrative activities like appointment scheduling and medication refills. • ChatGPT's sophisticated language processing capabilities make it the best tool for automating routine operations. • With data from tests, laboratories, vital signs, and symptoms available, this technology may be taught to match the information and provide doctors with suggestions for patient care and treatment options.
17.	Keep patients informed	• Doctors may keep patients informed throughout therapy by transmitting information to patients and utilising the ChatGPT bot to translate complicated medical records into understandable English. • Due to their hectic schedules, physicians are sometimes difficult to reach throughout the workday. • Answers to commonly asked questions about diagnosis and disease management are available on ChatGPT. • AI development has revolutionised computer science and changed how people interact with technology. • The ChatGPT, which can communicate with people in a nearly human-like way, is one of the most extraordinary instances of AI. • ChatGPT can produce human-like text in response to instructions given to a web browser in seconds, and this writing may be in various formats.
18.	Mental health-related contexts	• ChatGPT may be used in several mental health-related contexts, such as diagnosing mental diseases or supporting treatment sessions. • Without the need for expert therapists, ChatGPT may provide individualised replies catered to people seeking therapy. • Also, physicians and counsellors might diagnose their patients' illnesses more accurately by recognising a person's conversational patterns and constructing specialised therapies over time. • By using ChatGPT to its total capacity, we may reimagine how we connect with our interior thoughts and discover sympathetic paths to recovery. • ChatGPT has the potential to be a powerful tool for improving mental health and well-being. • When utilised properly, it may improve communication, help individuals comprehend their ideas and feelings, and teach them coping mechanisms for challenging circumstances.

(*continued on next page*)

Table 1 (*continued*).

S. No	Applications	Description
19.	Revolutionise the digital health sector	• ChatGPT has the power to revolutionise the digital health sector and lessen physician burnout. • By automating time-consuming activities, freeing up time to perform essential responsibilities, and boosting patient communication, ChatGPT may enhance the entire experience for both doctors and patients. • The use of ChatGPT in healthcare is only the start of a new age of AI-assisted medicine, but for now, it is imperative to maintain a human eye on the results it produces. • AI is used in a plethora of different ways. Hence, its applications appear limitless and full of potential in many fields, widening the range of options and indicating a promising future. • ChatGPT will continue to improve its capacity to discern human language and provide more sensible and appropriate responses. • The model may get even "smarter" and better at responding to a broader variety of questions and requests by applying more current ML methods and training on brand-new datasets. • ChatGPT has a wide range of possible future developments, and it is fascinating to see the app's continuous evolution and improvement. • In reality, ChatGPT elaborates its responses using sentences that are likely to be recognised as such by a person.

and correct answers to frequent questions and actions, including making plans or gathering information, Chat GPT may help individuals save time and effort. It may increase client happiness and loyalty by giving prompt, individualised solutions to client concerns. Internet search is one of ChatGPT's most often cited benefits. A straightforward question using ChatGPT often yields a straightforward and unambiguous response, which Google, despite all of its algorithms, seldom matches. ChatGPT was trained on a large corpus of conversational text and used the GPT paradigm of operation. As a result, it can answer questions and deliver information in a way that is practically identical to a human discussion.

Doctors may use this technology to instruct medical students and provide answers to their inquiries excitingly. The production of tests or quizzes is one possible educational use for ChatGPT. This network is built from several transformer blocks that analyse the input text and provide predictions. The network has self-attention mechanisms that allow one to evaluate the significance of various words and phrases about one another and the conversation as a whole. ChatGPT is an excellent tool for developers who want to create a new feature or enhance an existing one since it can comprehend and evaluate complicated technological ideas. A novice programmer may learn general programming concepts and computer science concepts with the aid of ChatGPT. It may provide knowledge of the advantages and disadvantages of different coding languages, the most efficient ways to deal with coding issues, and the best practices for software development. ChatGPT's advantages for companies are a game-changer that will help them better their minor operations and continually remain one step ahead of the competition.

In order to help with medical coding and billing, ChatGPT may provide recommendations for the better treatment of the patients as per the patient's symptoms and medical history. It automates clinical encounter reports and discharge summaries, saving patient care time. This is helpful for real-time, evidence-based clinical decision assistance for healthcare practitioners.

9. Limitations of ChatGPT in healthcare

With technological development, there is potential to influence society and the economy significantly. Thus, this AI tool has to be well thought out before any application. It could need help comprehending complicated or abstract ideas, which might result in errors or misinterpretations. The large quantity of data that ChatGPT has access to poses ethical and privacy issues. Examples include the risk of misuse or illegal access to private data. Some of the significant limitations we foresee are as follows.

- One of the significant limitations is that it may be used to propagate false healthcare information or fake news since the model might need to be able to tell which sources are trustworthy and which are not.
- This might promote the spread of inaccurate information and inflict damage on certain people or groups, which could have detrimental effects on society.
- ChatGPT could automate employment by providing chatbots and customer support personnel, which lead to healthcare job losses and a decline in the need for human involvement.
- A large number of ethical issues exist with using this smart chatbot in patient care and medical research.
- There are various issues, such as, predominantly medical ethics, privacy, consent, medical care standards, dependability, and equality.
- The implementation of ChatGPT in healthcare and research has ethical ramifications, although they still need to be better understood.
- ChatGPT puts users in danger of violating their privacy. The foundation of the doctor–patient relationship's trust is the confidentiality of patient information. This privacy is in danger from ChatGPT, a concern that weaker patients cannot fully comprehend.
- Patients may need to comprehend the implications of their agreement fully. Some people may still need to be asked for permission. As a result, medical professionals and organisations risk facing legal action.
- ChatGPT cannot even provide personalised advice to diabetic patients based on their blood sugar and medication data.
- As a language model, ChatGPT cannot prescribe treatments to patients or access their medical records.
- Responses from the ChatGPT are created automatically and can only sometimes match the subtlety or tone of a human answer, which might make the user experience less individualised.
- ChatGPT could not comprehend the context of a job or discussion as effectively as a person, which might result in less accurate or pertinent replies. Like with any automated system, there is a chance that ChatGPT replies may include mistakes or inaccuracies, which may cause confusion or annoyance for the user. If sensitive data is exchanged or kept inside the system, using Chat GPT may give rise to worries about data privacy and security.
- ChatGPT may provide inaccurate information or biased replies that can have a detrimental effect on the overall calibre of the code created is one of the main drawbacks for developers. The code may need to be fixed, usable, or unnecessarily complicated.
- The critical thinking and problem-solving skills humans are absent in ChatGPT. This might cause issues in several applications where ChatGPT could respond incorrectly because it needs help to grasp the subtleties of a query. The inability of ChatGPT to develop code that needs many contexts is another one of its drawbacks. To reach the intended outcome, developers would have to provide the code with every piece of information possible, which could be more practical.

- ChatGPT can produce straightforward code in various programming languages when given instructions, but it cannot handle complex issues. ChatGPT has yet to be widely used by companies to create programmes. Since the technology is still in its infancy, handling sophisticated coding jobs will take much work.
- ChatGPT must improve its capacity to process sophisticated questions or provide information tailored to a given environment. Additionally, since it cannot comprehend emotions or provide emotional support, the software may be unable to cope with emotional reactions or understand users. This may be a drawback when consumers seek emotional support, such as in mental health or counselling services.
- ChatGPT can provide human-like replies and has access to much information, but it needs a higher level of common sense. This technology could provide meaningful or correct answers to specific queries or circumstances for health-related issues, but not to all.
- ChatGPT can access much information, but it can only access some of the knowledge that doctors possess. It may not be able to respond to inquiries regarding very specialised or narrow subjects, and it might not be up to date on current advancements or changes in particular disciplines.

10. Future scope

The technology behind ChatGPT is still being developed and improved, and OpenAI has already published multiple versions of GPT with progressively better speed and features. In the upcoming days, this technology will be one of the most important in future healthcare development. It will be used in healthcare to get precise outcomes. The ChatGPT AI system does not yet compare its findings to real-world data, but real-world data integration is something anticipate in the future. Recruiters will use ChatGPT to automate several steps in the hiring process, improving efficiency and reducing time and costs. ChatGPT will help travellers quickly and conveniently book flights, hotels, and other forms of transportation. It may provide real-time information on weather, local events, and flight status, making it more straightforward for passengers to plan their travels and remain informed. Unlike any other technology, it can comprehend and produce text resembling human speech, making it adaptable and helpful for many healthcare purposes. ChatGPT is sure to impress and please, whether we use it for customer support, content production, or pleasure. It is thus a highly beneficial and effective tool for everyone. ChatGPT can be the future of healthcare eLearning, from creating text-based information to offering individualised learning experiences. Training companies must collaborate with a solution provider that provides integrated learning platform solutions driven by generative AI and GPT to fully realise these technologies' potential. ChatGPT is improving the state of all AI technologies. Deep learning's capabilities are being pushed to their limits to pave the path for future developments in AI technology.

11. Conclusion

ChatGPT is an effective tool for producing human-like text replies to questions. Its capacity to produce well-organised and educational text responses has made it a popular option for various applications. ChatGPT was developed using a vast amount of online content. ChatGPT offers several applications in the healthcare industry and benefits both patients and medical personnel. It tries to mimic human writing and may serve several functions in healthcare. The healthcare industry, which is constantly evolving to meet patients' increasing requirements, is one of the industries with the quickest rate of development globally. Because of technology improvements, ChatGPT is becoming a crucial tool for healthcare providers, offering a range of benefits to patients and healthcare professionals. The healthcare sector can use ChatGPT as it is considered cutting-edge conversational AI due to its tremendous training experience and exceptional natural language comprehension. The basis for medical guidance and treatment is high-quality evidence. In healthcare, patients and clinicians utilise a variety of channels to obtain data that influences their choices. However, at this stage of its development, ChatGPT may need to be sufficiently resourced or set up to provide accurate and objective information. Based on input data, ChatGPT may automatically produce medical reports, including radiology reports, pathology reports, and discharge summaries. Medical research articles may be analysed using ChatGPT to spot significant ideas and patterns and aid in the hunt for novel approaches. The data from adverse event reporting may be fine-tuned in ChatGPT to find patterns and trends that can be utilised to improve patient safety. Using the dataset, the ChatGPT model may be trained to comprehend user input naturally and accurately for healthcare. However, this AI tool cannot replace a doctor. There are several limitations pertaining to responsibility, medical ethics, legal framework, interpretation of data and variations in the human anatomy and responses.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M. Cascella, J. Montomoli, V. Bellini, E. Bignami, Evaluating the feasibility of ChatGPT in healthcare: An analysis of multiple clinical and research scenarios, J. Med. Syst. 47 (1) (2023) 1–5.

[2] B. Ram, P.V. Pratima Verma, Artificial intelligence AI-based Chatbot study of ChatGPT, google AI bard and baidu AI, World J. Adv. Eng. Technol. Sci. 8 (01) (2023) 258–261.

[3] Ö. Aydın, E. Karaarslan, OpenAI ChatGPT generated literature review: Digital twin in healthcare, 2022, Available at SSRN 4308687.

[4] S.B. Patel, K. Lam, ChatGPT: The future of discharge summaries? Lancet Dig. Health 5 (3) (2023) e107–e108.

[5] Y. Shen, L. Heacock, J. Elias, K.D. Hentel, B. Reig, G. Shih, L. Moy, ChatGPT and other large language models are double-edged swords, Radiology (2023) 230163.

[6] M.H. Temsah, A. Jamal, J.A. Al-Tawfiq, Reflection with ChatGPT about the excess death after the COVID-19 pandemic, New Microbes New Infect (2023).

[7] R.J.M. Ventayen, OpenAI ChatGPT generated results: Similarity index of artificial intelligence-based contents, 2023, Available at SSRN 4332664.

[8] A.M. DiGiorgio, J.M. Ehrenfeld, Artificial intelligence in medicine & ChatGPT: De-tether the physician, J. Med. Syst. 47 (1) (2023) 32.

[9] S.B. Johnson, A.J. King, E.L. Warner, S. Aneja, B.H. Kann, C.L. Bylund, Using ChatGPT to evaluate cancer myths and misconceptions: artificial intelligence and cancer information, JNCI Cancer Spectr 7 (2) (2023) pkad015.

[10] A. Grünebaum, J. Chervenak, S.L. Pollet, A. Katz, F.A. Chervenak, The exciting potential for ChatGPT in obstetrics and gynecology, Am. J. Obstet. Gynecol. (2023).

[11] A.H. Kumar, Analysis of ChatGPT tool to assess the potential of its utility for academic writing in biomedical domain, Biol. Eng. Med. Sci. Rep. 9 (1) (2023) 24–30.

[12] M. Aljanabi, ChatGPT: Future directions and open possibilities, Mesop. J. CyberSecur. 2023 (2023) 16–17.

[13] D. Singh, ChatGPT: A new approach to revolutionise organisations, Int. J. New Media Stud. (IJNMS) 10 (1) (2023) 57–63.

[14] S.S. Biswas, Role of chat GPT in public health, Ann. Biomed. Eng. (2023) 1–2.

[15] M. Abdullah, A. Madain, Y. Jararweh, ChatGPT: Fundamentals, applications and social impacts, in: 2022 Ninth International Conference on Social Networks Analysis, Management and Security, SNAMS, IEEE, 2022, pp. 1–8.

[16] T.H. Kung, M. Cheatham, A. Medenilla, C. Sillos, L. De Leon, C. Elepaño …, V. Tseng, Performance of ChatGPT on USMLE: Potential for AI-assisted medical education using large language models, PLoS Digit. Health 2 (2) (2023) e0000198.

[17] M. Mijwil, M. Aljanabi, A.H. Ali, ChatGPT: Exploring the role of cybersecurity in the protection of medical information, Mesop. J. Cybersecur. 2023 (2023) 18–21.

[18] M. Sallam, ChatGPT utility in health care education, research, and practice: Systematic review on the promising perspectives and valid concerns, Healthcare 2023 (11) (2023) 887.

[19] F.C. Kitamura, ChatGPT is shaping the future of medical writing but still requires human judgment, Radiology (2023) 230171.
[20] J. Gunawan, Exploring the future of nursing: Insights from the ChatGPT model, Belitung Nurs. J. 9 (1) (2023) 1–5.
[21] S. Biswas, ChatGPT and the future of medical writing, Radiology (2023) 223312.
[22] K. Alhasan, J. Al-Tawfiq, F. Aljamaan, A. Jamal, A. Al-Eyadhy, M.H. Temsah, J.A. Al-Tawfiq, Mitigating the burden of severe pediatric respiratory viruses in the post-COVID-19 era: Chatgpt insights and recommendations, Cureus 15 (3) (2023).
[23] L. De Angelis, F. Baglivo, G. Arzilli, G.P. Privitera, P. Ferragina, A.E. Tozzi, C. Rizzo, ChatGPT and the rise of large language models: The new AI-driven infodemic threat in public health, 2023, Available at SSRN 4352931.
[24] A. Lecler, L. Duron, P. Soyer, Revolutionising radiology with GPT-based models: Current applications, future possibilities and limitations of ChatGPT, Diagn. Interv. Imaging (2023).
[25] A. Haleem, M. Javaid, R.P. Singh, An era of ChatGPT as a significant futuristic support tool: A study on features, abilities, and challenges, BenchCouncil Trans. Benchmarks Stand. Eval. (2023) 100089.
[26] A. Arora, A. Arora, The promise of large language models in health care, Lancet 401 (10377) (2023) 641.
[27] A.B. Mbakwe, I. Lourentzou, L.A. Celi, O.J. Mechanic, A. Dagan, ChatGPT passing USMLE shines a spotlight on the flaws of medical education, PLoS Digit. Health 2 (2) (2023) e0000205.
[28] J. Homolak, Opportunities and risks of ChatGPT in medicine, science, and academic publishing: a modern Promethean dilemma, Croat. Med. J. 64 (1) (2023) 1–3.
[29] D.L. Mann, Artificial intelligence discusses the role of artificial intelligence in translational medicine: A JACC: Basic to translational science interview with ChatGPT, Basic Transl. Sci. (2023).
[30] L. Iftikhar, DocGPT: Impact of ChatGPT-3 on health services as a virtual doctor, EC Paediatr. (2023) 12, 45–55.
[31] H. Lee, The rise of ChatGPT: Exploring its potential in medical education, Anatom. Sci. Educ. (2023).
[32] G. van Schalkwyk, Artificial intelligence in pediatric behavioral health, Child Adoles. Psychiatry Mental Health 17 (1) (2023) 1–2.
[33] V.W. Xue, P. Lei, W.C. Cho, The potential impact of ChatGPT in clinical and translational medicine, Clin. Transl. Med. 13 (3) (2023).
[34] B. Gordijn, H.T. Have, ChatGPT: evolution or revolution? Med. Health Care Philos. (2023) 1–2.
[35] M.J. Ali, A. Djalilian, Readership awareness series–paper 4: Chatbots and ChatGPT-ethical considerations in scientific publications, in: Seminars in Ophthalmology, Taylor & Francis, 2023, pp. 1–2.
[36] G. Eysenbach, The role of ChatGPT, generative language models, and artificial intelligence in medical education: a conversation with ChatGPT and a call for papers, JMIR Med. Educ. 9 (1) (2023) e46885.
[37] S.R. Ali, T.D. Dobbs, H.A. Hutchings, I.S. Whitaker, Using ChatGPT to write patient clinic letters, Lancet Digit. Health (2023).
[38] A.S. George, A.H. George, A review of ChatGPT AI's impact on several business sectors, Partners Univ. Int. Innov. J. 1 (1) (2023) 9–23.
[39] J. Dahmen, M. Kayaalp, M. Ollivier, A. Pareek, M.T. Hirschmann, J. Karlsson, P.W. Winkler, Artificial intelligence bot ChatGPT in medical research: the potential game changer as a double-edged sword, Knee Surg. Sports Traumatol. Arthrosc. (2023) 1–3.
[40] R.S. D'Amico, T.G. White, H.A. Shah, D.J. Langer, I asked a ChatGPT to write an editorial about how we can incorporate chatbots into neurosurgical research and patient care., Neurosurgery (2022) 10–1227.
[41] O.P. Singh, Artificial intelligence in the era of ChatGPT-opportunities and challenges in mental health care, Indian J. Psychiatry 65 (3) (2023) 297–298.
[42] R.K. Sinha, A.D. Roy, N. Kumar, H. Mondal, R. Sinha, Applicability of ChatGPT in assisting to solve higher order problems in pathology, Cureus 15 (2) (2023).
[43] M.R. King, P.T. chatG, A conversation on artificial intelligence, chatbots, and plagiarism in higher education, Cell. Mol. Bioeng. (2023) 1–2.
[44] F. Ufuk, The role and limitations of large language models such as ChatGPT in clinical settings and medical journalism, Radiology (2023) 230276.
[45] A.M. Hopkins, J.M. Logan, G. Kichenadasse, M.J. Sorich, Artificial intelligence chatbots will revolutionise how cancer patients access information: ChatGPT represents a paradigm shift, JNCI Cancer Spect. 7 (2) (2023) pkad010.
[46] G.H. Sun, S.H. Hoelscher, The ChatGPT storm and what faculty can do, Nurse Educ. (2023) 10–1097.
[47] T.B. Arif, U. Munaf, I. Ul-Haque, The future of medical education and research: Is ChatGPT a blessing or blight in disguise? Med. Educ. Online 28 (1) (2023) 2181052.
[48] C. Ahn, Exploring ChatGPT for information of cardiopulmonary resuscitation, Resuscitation (2023) 185.
[49] G.G.R. Sng, J.Y.M. Tung, D.Y.Z. Lim, Y.M. Bee, Potential and pitfalls of ChatGPT and natural-language artificial intelligence models for diabetes education, Diabetes Care (2023) dc230197.
[50] D. Baidoo-Anu, L. Owusu Ansah, Education in the era of generative artificial intelligence (AI): Understanding the potential benefits of ChatGPT in promoting teaching and learning, 2023, Available at SSRN 4337484.
[51] J.H. Lubowitz, ChatGPT, an artificial intelligence chatbot, is impacting medical literature, Arthroscopy (2023).
[52] M. Ollivier, A. Pareek, J. Dahmen, M. Kayaalp, P.W. Winkler, M.T. Hirschmann, J. Karlsson, A deeper dive into ChatGPT: History, use and future perspectives for orthopaedic research, Knee Surg. Sports Traumatol. Arthrosc. (2023) 1–3.
[53] M. Balas, E.B. Ing, Conversational AI models for ophthalmic diagnosis: Comparison of ChatGPT and the isabel pro differential diagnosis generator, JFO Open Ophthalmol. (2023) 100005.
[54] A. Juhi, N. Pipil, S. Santra, S. Mondal, J.K. Behera, H. Mondal ., J.K. Behera IV, The capability of ChatGPT in predicting and explaining common drug-drug interactions, Cureus 15 (3) (2023).
[55] L. Zhu, W. Mou, R. Chen, Can the ChatGPT and other large language models with internet-connected database solve the questions and concerns of patients with prostate cancer?, 2023, medRxiv, 2023-03.
[56] M.R. Chavez, T.S. Butler, P. Rekawek, H. Heo, W.L. Kinzler, ChatGPT (generative pre-trained transformer): Why we should embrace this technology, Am. J. Obstet. Gynecol. (2023).
[57] A.T. Gabrielson, A.Y. Odisho, D. Canes, Harnessing generative artificial intelligence to improve efficiency among urologists: Welcome ChatGPT, J. Urol. (2023) 10–1097.
[58] T.J. Chen, ChatGPT and other artificial intelligence applications speed up scientific writing, J. Chin. Med. Assoc. 1 (2023) 0–1097.
[59] V. Taecharungroj, What can ChatGPT do? Analysing early reactions to the innovative AI chatbot on Twitter, Big Data Cogn. Comput. 7 (1) (2023) 35.
[60] B. Rathore, Future of textile: Sustainable manufacturing & prediction via ChatGPT, Eduzone: Int. Peer Rev./Ref. Multidiscip. J. 12 (1) (2023) 52–62.
[61] M.R. King, The future of AI in medicine: A perspective from a chatbot, Ann. Biomed. Eng. (2022) 1–5.
[62] I. Dergaa, K. Chamari, P. Zmijewski, H.B. Saad, From human writing to artificial intelligence generated text: examining the prospects and potential threats of ChatGPT in academic writing, Biol. Sport 40 (2) (2023) 615–622.
[63] O. Oviedo-Trespalacios, A.E. Peden, T. Cole-Hunter, A. Costantini, M. Haghani, S. Kelly ., G. Reniers, The risks of using ChatGPT to obtain common safety-related information and advice, 2023, Available at SSRN 4346827.
[64] S. Sok, K. Heng, ChatGPT for education and research: A review of benefits and risks, 2023, Available at SSRN 4378735.
[65] V. Milan-Ortiz, A.R. Damughatla, A.M. Qazi, S. Kamatham, S. Oli, P. Koleti ., A. Qazi, Neutropenic enterocolitis following autologous stem cell transplantation: A compelling clinical case report written with the assistance of ChatGPT, Cureus 15 (3) (2023).
[66] N. Kurian, J.M. Cherian, N.A. Sudharson, K.G. Varghese, S. Wadhwa, AI is now everywhere, Br. Dent. J. 234 (2) (2023) 72.
[67] I. Munir, Artificial intelligence ChatGPT in medicine. Can it be the friend you are looking for? J. Bangladesh Med. Assoc. North Am. (BMANA) BMANA J. (2023) 01–04.
[68] A.J. Nastasi, K.R. Courtright, S.D. Halpern, G.E. Weissman, Does ChatGPT provide appropriate and equitable medical advice?: A vignette-based, clinical evaluation across care contexts, 2023, medRxiv, 2023-02.
[69] R.A. Khan, M. Jawaid, A.R. Khan, M. Sajjad, ChatGPT-reshaping medical education and clinical management, Pak. J. Med. Sci. 39 (2) (2023).
[70] A. Scerri, K.H. Morin, Using chatbots like ChatGPT to support nursing practice, J. Clin. Nurs. (2023).
[71] G. Sebastian, Do ChatGPT and other AI chatbots pose a cybersecurity risk?: An exploratory study, Int. J. Secur. Priv. Perv. Comput. (IJSPPC) 15 (1) (2023) 1–11.
[72] K. Uludag, Testing creativity of ChatGPT in psychology: Interview with ChatGPT, 2023, Available at SSRN 4390872.
[73] P. Gandhi, V. Talwar, Artificial intelligence and ChatGPT in the legal context, Indian J. Med. Sci. 75 (1) (2023) 1.
[74] B. Rathore, Future of AI & generation alpha: Chatgpt beyond boundaries, Eduzone: Int. Peer Rev./Ref. Multidiscip. J. 12 (1) (2023) 63–68.
[75] M.D. Xames, J. Shefa, ChatGPT for research and publication: Opportunities and challenges, 2023, Available at SSRN 4381803.
[76] U.K. Hisan, M.M. Amri, ChatGPT and medical education: A double-edged sword, J. Pedagog. Educ. Sci. 2 (01) (2023).
[77] J. Deng, Y. Lin, The benefits and challenges of ChatGPT: An overview, Front. Comput. Intell. Syst. 2 (2) (2022) 81–83.
[78] H. Alkaissi, S.I. McFarlane, Artificial hallucinations in ChatGPT: implications in scientific writing, Cureus 15 (2) (2023).

TBench Editorial Board

More information about this series at https://www.benchcouncil.org/tbench/

TBench Call For Papers

BenchCouncil Transactions on Benchmarks, Standards and Evaluations (TBench)
ISSN:2772-4859

Aims and Scopes

BenchCouncil Transactions on Benchmarks, Standards, and Evaluations (TBench) publishes position articles that open new research areas, research articles that address new problems, methodologies, tools, survey articles that build up comprehensive knowledge, and comments articles that argue the published articles. The submissions should deal with the benchmarks, standards, and evaluation research areas. Particular areas of interest include, but are not limited to:

- 1. Generalized benchmark science and engineering (see https://www.sciencedirect.com/science/article/pii/S2772485921000120), including but not limited to
  - measurement standards
  - standardized data sets with defined properties
  - representative workloads
  - representative data sets
  - best practices
- 2. Benchmark and standard specifications, implementations, and validations of:
  - Big Data
  - AI
  - HPC
  - Machine learning
  - Big scientific data
  - Datacenter
  - Cloud
  - Warehouse-scale computing
  - Mobile robotics
  - Edge and fog computing
  - IoT
  - Chain block
  - Data management and storage
  - Financial domains
  - Education domains
  - Medical domains
  - Other application domains
- 3. Data sets
  - Detailed descriptions of research or industry datasets, including the methods used to collect the data and technical analyses supporting the quality of the measurements.
  - Analyses or meta-analyses of existing data and original articles on systems, technologies, and techniques that advance data sharing and reuse to support reproducible research.
  - Evaluating the rigor and quality of the experiments used to generate the data and the completeness of the data description.
  - Tools generating large-scale data while preserving their original characteristics.
- 4. Workload characterization, quantitative measurement, design, and evaluation studies of:
  - Computer and communication networks, protocols, and algorithms
  - Wireless, mobile, ad-hoc and sensor networks, IoT applications
  - Computer architectures, hardware accelerators, multi-core processors, memory systems, and storage networks
  - High-Performance Computing
  - Operating systems, file systems, and databases

- Virtualization, data centers, distributed and cloud computing, fog, and edge computing
- Mobile and personal computing systems
- Energy-efficient computing systems
- Real-time and fault-tolerant systems
- Security and privacy of computing and networked systems
- Software systems and services, and enterprise applications
- Social networks, multimedia systems, Web services
- Cyber-physical systems, including the smart grid

- 5. Methodologies, metrics, abstractions, algorithms, and tools for:
  - Analytical modeling techniques and model validation
  - Workload characterization and benchmarking
  - Performance, scalability, power, and reliability analysis
  - Sustainability analysis and power management
  - System measurement, performance monitoring, and forecasting
  - Anomaly detection, problem diagnosis, and troubleshooting
  - Capacity planning, resource allocation, run time management, and scheduling
  - Experimental design, statistical analysis, simulation
- 6. Measurement and evaluation
  - Evaluation methodology and metric
  - Testbed methodologies and systems
  - Instrumentation, sampling, tracing, and profiling of Large-scale real-world applications and systems
  - Collection and analysis of measurement data that yield new insights
  - Measurement-based modeling (e.g., workloads, scaling behavior, assessment of performance bottlenecks)
  - Methods and tools to monitor and visualize measurement and evaluation data
  - Systems and algorithms that build on measurement-based findings
  - Advances in data collection, analysis, and storage (e.g., anonymization, querying, sharing)
  - Reappraisal of previous empirical measurements and measurement-based conclusions
  - Descriptions of challenges and future directions the measurement and evaluation community should pursue

Bench 2023 CALL FOR PAPERS

The 15th BenchCouncil International Symposium on Benchmarking, Measuring and Optimizing (Bench 2023)

In conjunction with Federated Intelligent Computing and Chip Conference (FICC 2023)

https://www.benchcouncil.org/bench2023/index.html

Full Papers: July 31, 2023, at 11:59 PM AoE
Notification: September 30, 2023, at 11:59 PM AoE
Final Papers Due: October 31, 2023, at 11:59 PM AoE
Conference Date: December 3–5, 2023
Venue: Sanya, China.

Please note that citizens from up to 59 nations can visit Sanya without a Visa from the Chinese Government. Sanya is a beautiful seaside city, well known as Hawaii in China.

Submission website: https://bench2023.hotcrp.com/

Introduction

Evolving from nine BPOE/SDBA workshops in conjunction with ASPLOS, VLDB, and ICS, Bench is an international multidisciplinary conference on benchmarks, standards, data sets, evaluation, and optimization. Bench 2023 is the fifteenth edition. The Bench conference encompasses a wide range of topics in benchmarks, datasets, metrics, indexes, measurement, evaluation, optimization, supporting methods and tools, and industry best practices in computer science, AI, medicine, finance, education, management, etc. Benchâs multidisciplinary and interdisciplinary emphasis provides an ideal environment for developers and researchers from different areas and communities to discuss practical and theoretical work.

Bench 2023 invites manuscripts describing original work in the above areas and topics (Call for Papers). All accepted papers will be presented at the Bench 2023 conference and published by Springer LNCS (Pending, Indexed by EI). At least one of the authors of the TBench articles published last year is requested to present their work at the Bench conference.

Regularly, the Bench conference will present the BenchCouncil Achievement Award ($3000), the BenchCouncil Rising Star Award ($1000), the BenchCouncil Best Paper Award ($1000), and the BenchCouncil Distinguished Doctoral Dissertation Awards in Computer Architecture ($1000) and in other areas ($1000). This year, the BenchCouncil Distinguished Doctoral Dissertation Award includes two tracks: computer architecture and other areas. Among the submissions of each track, four candidates will be selected as finalists. They will be invited to give a 30-minute presentation at the Bench 2023 Conference and contribute research articles to BenchCouncil Transactions on Benchmarks, Standards and Evaluation. Finally, for each track, one among the four will receive the award for each track, which carries a $1,000 honorarium.

With generous support from BenchCouncil, Bench 2023 will offer travel grants for students to defray a portion of their travel cost. The size and number of these grants will vary depending on funding availability, the number of student applicants, and their respective priority. Grant awards will be made before the early registration deadline; expenses will be reimbursed after the conference; grant recipients will be asked to submit original receipts to verify their expenditures as well as a 1-page summary of their involvement during the conference. While we encourage all in need of a travel grant to apply, the selection process will give higher priority to students who would otherwise not be able to attend the conference. We strongly encourage applications from students that belong to under-represented groups.

Organization

General Co-Chairs
Rakesh Agrawal, Data Insights Laboratories, San Jose, CA, USA
Aoying Zhou, East China Normal University

Program Co-Chairs
Weining Qian, East China Normal University
Sascha Hunold, TU Wien, Austria

Program Vice-Chairs
Biwei Xie, Institute of Computing Technology, CAS
Kai Shu, Illinois Institute of Technology

Web Chair
Jiahui Dai, BenchCouncil

Technical Program Committee (continuously updated):
Bin Ren, William & Mary
Guangli Li, Institute of Computing Technology, Chinese Academy of Sciences
Gwangsun Kim, POSTECH
Khaled Ibrahim, Lawrence Berkeley National Laboratory
Mario Marino, Leeds Beckett University
Miaoqing Huang, University of Arkansas
Murali Emani, Argonne National Laboratory
Vladimir Getov, University of Westminster
Woongki Baek, UNIST
Xiaoyi Lu, University of California, Merced
Zhen Jia, Amazon
Steven Farrell, Lawrence Berkeley National Laboratory

Award Committees
2023 BenchCouncil Achievement Award Committee:
Prof. D. K. Panda, the Ohio State University
Prof. Lizy Kurian John, the University of Texas at Austin
Prof. Geoffrey Fox, Indiana University
Prof. Jianfeng Zhan, University of Chinese Academy of Sciences
Prof. Tony Hey, Rutherford Appleton Laboratory STFC (Since 2020)
Prof. David J. Lilja, University of Minnesota, Minneapolis (Since 2021)
Prof. Jack J. Dongarra, University of Tennessee (Since 2022)
John L. Henning, Oracle (Since 2023)

2023 BenchCouncil Rising Star Award Committees:
Prof. D. K. Panda, the Ohio State University
Prof. Lizy Kurian John, the University of Texas at Austin
Prof. Geoffrey Fox, Indiana University
Prof. Jianfeng Zhan, University of Chinese Academy of Sciences
Prof. Torsten Hoefler, ETH Zürich (Since 2021)
Prof. Vijay Janapa Reddi, Harvard University (Since 2022)
Dr. Peter Mattson, Google, USA (Since 2022)
Dr. Wanling Gao , ICT, Chinese Academy of Sciences (pending)
Dr. Douwe Kiela, Stanford University (Since 2023)

BenchCouncil Distinguished Doctoral Dissertation Award Committee in Other Areas:
Prof. Jack Dongarra, University of Tennessee
Dr. Xiaoyi Lu, The University of California, Merced
Dr. Jeyan Thiyagalingam, STFC-RAL

Dr. Lei Wang, ICT, Chinese Academy of Sciences
Dr. Spyros Blanas, The Ohio State University

BenchCouncil Distinguished Doctoral Dissertation Award Committee in Computer Architecture:
Prof. Resit Sendag, University of Rhode Island, USA
Dr. Peter Mattson, Google
Dr. Vijay Janapa Reddi, Harvard University
Dr. Wanling Gao, Chinese Academy of Sciences

Bench Steering Committees
Prof. Dr. Jack Dongarra, University of Tennessee
Prof. Dr. Geoffrey Fox, Indiana University
Prof. Dr. D. K. Panda, The Ohio State University
Prof. Dr. Felix, Wolf, TU Darmstadt.
JProf. Dr. Xiaoyi Lu, University of California, Merced
Prof. Dr. Resit Sendag, University of Rhode Island, USA
Dr. Wanling Gao, ICT, Chinese Academy of Sciences & UCAS
Prof. Dr. Jianfeng Zhan, BenchCouncil

Call for papers

The Bench conference encompasses a wide range of topics in benchmarks, datasets, metrics, indexes, measurement, evaluation, optimization, supporting methods and tools, and other best practices in computer science, medicine, finance, education, management, etc. Benchâs multidisciplinary and interdisciplinary emphasis provides an ideal environment for developers and researchers from different areas and communities to discuss practical and theoretical work. The topics of interest include, but are not limited to the following:

– Benchmark science and engineering across multi-disciplines: The formulation of problems or challenges in emerging and future computing; The benchmarks, datasets, and indexes in multidisciplinary applications, e.g., medical, finance, education, management, psychology, etc; Benchmark-based quantitative approaches to tackle multidisciplinary and interdisciplinary challenges; Industry best practices.

– Benchmark and standard specifications, implementations, and validations: Big Data, Artificial intelligence (AI), High performance computing (HPC), Machine learning, Big scientific data, Datacenter, Cloud, Warehouse-scale computing, Mobile robotics, Edge and fog computing, Internet of Things (IoT), Blockchain, Data management and storage, Financial, Education, Medical or other application domains.

– Dataset: Detailed descriptions of research or industry datasets, including the methods used to collect the data and technical analyses supporting the quality of the measurements; Analyses or meta-analyses of existing data and original articles on systems, technologies, and techniques that advance data sharing and reuse to support reproducible research; Evaluating the rigor and quality of the experiments used to generate the data and the completeness of the data description; Tools that can generate large-scale data while preserving their original characteristics.

– Workload characterization, quantitative measurement, design, and evaluation studies: Computer and communication networks, protocols and algorithms; Wireless, mobile, ad-hoc and sensor networks, IoT applications; Computer architectures, hardware accelerators, multi-core processors, memory systems and storage networks; HPC systems; Operating systems, file systems and databases; Virtualization, data centers, distributed and cloud computing, fog and edge computing; Mobile and personal computing systems; Energy-efficient computing systems; Real-time and fault-tolerant systems; Security and privacy of computing and networked systems; Software systems and services, and enterprise applications; Social networks, multimedia systems, web services; Cyber-physical systems, including the smart grid.

– Methodologies, metrics, abstractions, algorithms, and tools: Analytical modeling techniques and model validation; Workload characterization and benchmarking; Performance, scalability, power and reliability analysis; Sustainability analysis and power management; System measurement, performance monitoring and forecasting; Anomaly detection,

problem diagnosis and troubleshooting; Capacity planning, resource allocation, run time management and scheduling; Experimental design, statistical analysis, and simulation.

– Measurement and evaluation: Measurement standards; Evaluation methodologies and metrics; Testbed methodologies and systems; Instrumentation, sampling, tracing and profiling of large-scale, real-world applications and systems; Collection and analysis of measurement data that yield new insights; Measurement-based modeling (e.g., workloads, scaling behavior, assessment of performance bottlenecks); Methods and tools to monitor and visualize measurement and evaluation data; Systems and algorithms that build on measurement-based findings; Advances in data collection, analysis and storage (e.g., anonymization, querying, sharing); Reappraisal of previous empirical measurements and measurement-based conclusions; Descriptions of challenges and future directions that the measurement and evaluation community should pursue.

Paper Submission

Papers must be submitted in PDF. For a full paper, the page limit is 15 pages in the LNCS format, not including references. For a short paper, the page limit is 8 pages in the LNCS format, not including references. The review process follows a strict double-blind policy per the established Bench conference norms. The submissions will be judged based on the merit of the ideas rather than the length. After the conference, the proceedings will be published by Springer LNCS (Pending, Indexed by EI). Please note that the LNCS format is the final one for publishing.

At least one author must pre-register for the symposium, and at least one author must attend the symposium to present the paper. Papers for which no author is pre-registered will be removed from the proceedings.

Formatting Instructions

Please make sure your submission satisfies ALL of the following requirements:

- All authors and affiliation information must be anonymized.
- Paper must be submitted in printable PDF format.
- Please number the pages of your submission.
- The submission must be formatted for black-and-white printers. Please make sure your figures are readable when printed in black and white.
- The submission must describe unpublished work that is not currently under review of any other conference or journal venues.

Submission site: https://bench2023.hotcrp.com/
LNCS latex template: https://www.benchcouncil.org/file/llncs2e.zip

Awards

BenchCouncil Achievement Award ($3,000)

- This award recognizes a senior member who has made long-term contributions to benchmarking, measuring, and optimizing. The winner is eligible for the status of a BenchCouncil Fellow.

BenchCouncil Rising Star Award ($1,000)

- This award recognizes a junior member who demonstrates outstanding potential for research and practice in benchmarking, measuring, and optimizing.

BenchCouncil Best Paper Award ($1,000)

- This award recognizes a paper presented at the Bench conferences, which demonstrates potential impact on research and practice in benchmarking, measuring, and optimizing.

BenchCouncil Distinguished Doctoral Dissertation Award ($2000)

- This award recognizes and encourages superior research and writing by doctoral candidates in the broad field of benchmarks, data, standards, evaluations, and optimizations community. This year, the award includes two tracks, including the BenchCouncil Distinguished Doctoral Dissertation Award in Computer Architecture ($1000) and BenchCouncil Distinguished Doctoral Dissertation Award in other areas ($1000).