IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 1 December 2024, accepted 5 January 2025, date of publication 20 January 2025, date of current version 27 January 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3531391

RESEARCH ARTICLE

Performance of Quantum Annealing Machine Learning Classification Models on ADMET Datasets

HADI SALLOUM[1,2,3], KAMIL SABBAGH[1,3], VLADISLAV SAVCHUK[1], RUSLAN LUKIN[1], OSAMA ORABI[2,3], MARAT ISANGULOV[1], AND MANUEL MAZZARA[2]

[1]Research Center of the Artificial Intelligence Institute, Innopolis University, 420500 Innopolis, Russia
[2]Department of Computer Science and Engineering, Innopolis University, 420500 Innopolis, Russia
[3]Q Deep, 420502 Innopolis, Russia

Corresponding author: Hadi Salloum (h.salloum@innopolis.ru)

All authors were supported by the Research Center of the Artificial Intelligence Institute of Innopolis University.

ABSTRACT The Quantum Annealer built by D-Wave, known as Advantage, is currently the largest quantum computer in the world, featuring a topology called "Pegasus." This groundbreaking system opens new possibilities for solving highly complex problems. The advancement of quantum annealers has spurred experimental demonstrations and intensified research interest, particularly in quantum machine learning. However, the application of quantum annealing in machine learning remains limited due to the lack of conclusive performance evaluations on real-world datasets. There is still no clear consensus on the efficacy of these models in practical scenarios. This work focuses on experimentally evaluating quantum annealing machine learning (QAML) classification methods, specifically Quantum Support Vector Machines (QSVM) and QBoost, on ADMET datasets—one of the most important datasets in the drug discovery domain. We compare QAML with classical machine learning to evaluate their relative performance. This study seeks to address this gap by rigorously comparing the performance of QBoost and QSVM models using ADMET datasets, employing D-Wave's Quantum Annealer. This work provides a comprehensive analysis of the potential and limitations of quantum annealers in quantum machine learning, with a focus on their application to real-world data in the ADMET domain. The findings offer critical insights into the nuanced advantages and challenges of quantum annealers in advancing machine learning methodologies.

INDEX TERMS Quantum computing, quantum annealing, Pegasus topology, quantum support vector machines (QSVM), QBoost, ADMET datasets, drug discovery, classical machine learning, real-world data.

I. INTRODUCTION

Quantum Computing (QC) stands at the forefront of technological advancements, promising Innovative capabilities that could redefine computational power across various domains. Rooted in the foundational principles of quantum mechanics, QC utilizes concepts specifically quantum superposition and entanglement to enable computations that surpass the capabilities of classical computers [1], [2].

Richard Feynman, a well-known physicist, laid down the conceptual groundwork for the development of quantum computing in his 1959 lecture [3]. In a series of foundational papers [4], [5], [6], Feynman imagined how a quantum computer might work and how quantum algorithms could be successfully designed, while pointing out their potential in simulating physical systems. Although Feynman did not explicitly formulate important notions for quantum computing such as qubits or gates, his seminal insights influenced subsequent developments in the field. Building on Feynman's insights and earlier works by Poplavskiĭ [7] and Ingarden [8], Paul Benioff [9], [10] described the first

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Huang.

quantum mechanical model of a computer in the early '80s. This was further expanded upon by Deutsch at Oxford [11], who became the first researcher in 1985 to formally propose the concept of a quantum Turing machine, thus laying the foundation for a universal quantum computer.

The development of quantum computing has been marked by significant milestones over the years. Igeta and Yamamoto [12] made a pivotal contribution by describing the first physical realization of a quantum computer, while Ekert [13] introduced the first entanglement-based secure communication protocol. However, it was not until the late 1990s that the term ''qubit'' was formally coined [14], marking a watershed moment in the field. This period also witnessed the first experimental demonstration of a quantum algorithm [15], as well as the establishment of The Quantum Information Science and Technology Roadmapping Project in 2002, followed by the creation of the first operational pure-state NMR quantum computer in 2004 [16], [17].

Since then, the field of gate-based quantum computing has advanced at a rapid pace, with various physical systems being employed to represent qubits. These systems manipulate qubits by applying quantum gates, which enable increasingly complex computations. Over the past two decades, numerous technological paradigms have emerged, each contributing uniquely to the progress of quantum computing:

- **Superconducting Qubits:** This architecture uses superconducting circuits to represent and manipulate qubits. IBM's Quantum systems [18] and Google's quantum computing initiatives [19] are at the forefront of developments in this area.
- **Trapped Ion Quantum Computers:** In this system, individual ions trapped in electromagnetic fields serve as qubits. Companies like IonQ [20] and Honeywell [21] are advancing this approach, with notable progress in scalability and error correction.
- **Topological Quantum Computers:** This approach leverages topological qubits, which offer robustness against certain types of errors. Microsoft's StationQ project [22] is actively researching this promising technology.
- **Photonic Quantum Computers:** Using photons, or light particles, to represent qubits, photonic quantum computing offers potential advantages in terms of speed and fault tolerance. Companies like Xanadu and PsiQuantum are at the leading edge of research in this domain [23], [24].
- **Spin Qubits in Semiconductors:** By using the spin state of electrons in semiconductor materials to represent qubits, this approach is being pioneered by companies such as Intel and QuTech [25], [26], [27].
- **Neutral Atom Qubits:** Neutral atoms trapped by lasers serve as qubits in this emerging technology. ColdQuanta and Atom Computing are spearheading efforts in neutral atom qubit research [28], [29], [30].
- **Nitrogen-Vacancy (NV) Centers in Diamond:** Defects in diamond structures are used to store and process quantum information in this type of quantum computing, with research being conducted at leading institutions such as Harvard and MIT [31], [32], [33].
- **Quantum Dots:** These quantum computers employ quantum dots—nanoscale semiconductor structures—to confine electrons and represent qubits. Toshiba and IBM are among the organizations advancing research in this field [34], [35], [36].

While gate-based quantum computing has seen remarkable advancements across diverse technological architectures, quantum annealers represent a different approach. Quantum annealers are designed to solve combinatorial optimization problems by executing the quantum annealing process, as highlighted by Heng [37]. This form of quantum computing is distinct from gate-based systems, as it focuses on specific problem-solving techniques rather than general-purpose computation. As such, quantum annealers have found applications in areas requiring highly specialized optimization techniques, complementing the broader progress in quantum computing research.

Quantum annealing is a heuristic approach to solving problems in combinatorial optimization, initially developed for implementation on classical machines [38], [39], [40]. It can be viewed as a variant of the well-known simulated annealing (SA) metaheuristic [41]. The concept of integrating a model of quantum annealing into a heuristic optimization framework has been independently proposed by several researchers, including Apolloni [42], Ray et al. [43], and Kadowaki and Nishimori [44]. Tadashi Kadowaki and Hidetoshi Nishimori first proposed quantum annealing [44]. They introduced quantum fluctuations into the simulated annealing process for optimization problems, intending to achieve faster convergence to the optimal state. Quantum fluctuations induce transitions between states, playing a role analogous to thermal fluctuations in conventional approaches. This idea was tested using the transverse Ising model, where the transverse field varies over time, similar to how temperature changes in classical simulated annealing.

The objective is to find the ground state of the diagonal part of the Hamiltonian with high accuracy as quickly as possible. The time-dependent Schrödinger equation was numerically solved for small systems with various exchange interactions. A comparison with the results from classical thermal methods reveals that quantum annealing leads to the ground state with a much higher probability in almost all cases, assuming the same annealing schedule is followed, as shown in Fig. 1.

Quantum annealing operates on the principle of quantum tunneling, driven by quantum fluctuations, which facilitates state transitions toward the system's ground state. The final ground state is reached through the controlled application of quantum fluctuations. Initially, large quantum fluctuations are applied, which are then progressively reduced, allowing the system to evolve into the ground state. Figure 2 presents a schematic illustration of the quantum annealing process [45].

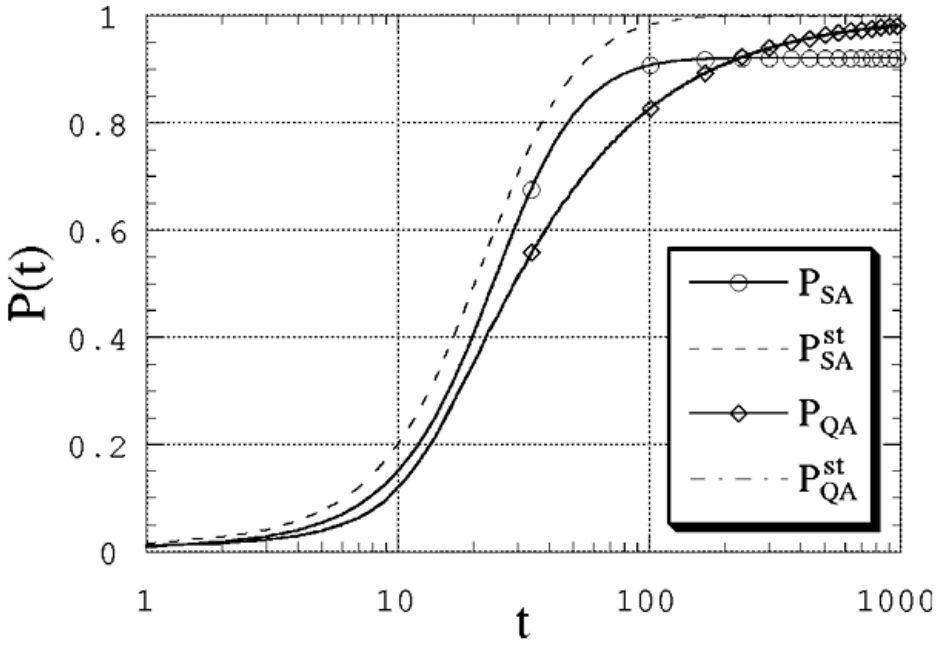


FIGURE 1. Time dependence of the overlaps of the ferromagnetic model. The quantum method clearly gives better convergence to the ground state, while the classical counterpart gets stuck in a local minimum with a non-negligible probability. Adapted from [44].

Mathematically, quantum annealing aims to find the minimum of a cost function, often represented by a Hamiltonian:

$$H = \sum_i \omega_i \sigma_i^x + \sum_{i<j} J_{ij} \sigma_i^z \sigma_j^z, \tag{1}$$

where σ_i^x and σ_i^z are Pauli matrices representing the spins of qubits, ω_i are biases, and J_{ij} are couplings between qubits. The ground state of this Hamiltonian corresponds to the solution of the optimization problem.

In quantum annealing, the system's evolution is governed by the time-dependent Schrödinger equation:

$$i\hbar \frac{\partial}{\partial t} |\psi(t)\rangle = H(t) |\psi(t)\rangle \tag{2}$$

where $|\psi(t)\rangle$ is the state of the quantum system at time t, and $H(t)$ is the time-dependent Hamiltonian. During the annealing process, the Hamiltonian is gradually changed from an initial Hamiltonian H_0, whose ground state is easy to prepare, to the final problem Hamiltonian H_F, whose ground state encodes the solution to the optimization problem. The time-dependent Hamiltonian is often expressed as:

$$H(t) = (1 - s(t))H_0 + s(t)H_F \tag{3}$$

where $s(t)$ is a schedule function that varies from 0 to 1 throughout the annealing process. A common choice for the schedule function is a linear ramp, i.e., $s(t) = t/T$, where T is the total annealing time.

Quantum entanglement is a fundamental phenomenon that significantly influences the evolution of qubit systems during the quantum annealing process. As demonstrated in the work cited in [46], both two-qubit and eight-qubit systems exhibit a gradual transition of the ground state as the system parameters are varied. The system is initialized in the ground state of H_0, which is typically a simple Hamiltonian with a known ground state. As the annealing progresses ($s(t)$ increases), the system evolves towards the ground state of H_F, ideally finding the optimal solution to the optimization problem encoded in H_F.

Initially, when the parameter s is small, the system manifests a superposition, wherein each qubit has an equal probability of being in the "up" or "down" state. As s increases, the energy gap between the ground and excited states diminishes, leading to the formation of entangled states that represent a superposition of all qubits being "up" or all being "down." Eventually, as the energy gap becomes smaller than the thermal energy, the system transitions into a mixed state, resulting in a loss of entanglement. Upon the completion of the quantum annealing process (when $s = 1$), the system stabilizes into two localized configurations, where all qubits align in the same direction—either all "up" or all "down." This entire process is illustrated in Figure 3.

Quantum annealing algorithms can be analyzed using various techniques, including adiabatic theorem analysis, quantum Monte Carlo simulations, and theoretical investigations of the system's energy landscape. These analyses help understand the behavior of quantum annealing algorithms and their effectiveness in solving optimization problems. Additionally, it's important to note that the optimization problem can be formulated as a Quadratic Unconstrained Binary Optimization (QUBO) problem, allowing for the conversion of classical optimization problems into a form suitable for quantum annealing. The QUBO formulation is

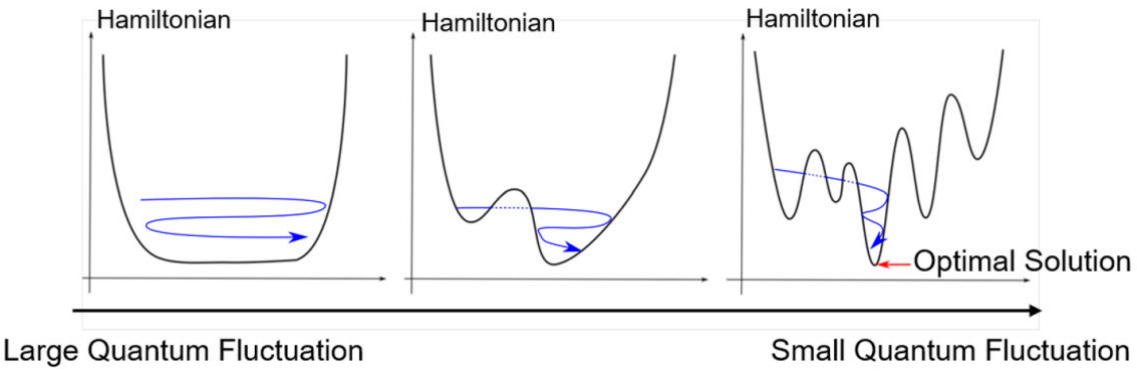


FIGURE 2. Schematic of the system of quantum annealing. Adapted from [45].

typically represented as:

$$\text{minimize } f(x) = x^T Q x \tag{4}$$

where x is a binary vector representing the variables of the optimization problem, and Q is the QUBO matrix encoding the problem's objective function and constraints.

These systems are offered by companies such as D-Wave Systems with their D-Wave Quantum Annealers [47]. Private companies have also played a significant role, directing efforts toward fully connected quantum computers, software libraries, and leveraging quantum computing for enhancing security services [48], [49], [50], [51], [52].

In recent years, researchers have increasingly recognized the transformative potential of quantum computing in advancing machine learning. This growing interest is reflected in numerous studies highlighting the promise of quantum algorithms in enhancing both the performance and efficiency of learning models [53], [54], [55], [56]. Building upon this foundation, our research delves into the capabilities of D-Wave's Advantage Quantum Annealer—the largest quantum computer in existence, boasting 5000 qubits [57]. The core objective of this paper is to assess the quantum advantages of this annealer model when applied to machine learning classification tasks across diverse datasets. Through a thorough and methodical analysis, we aim to uncover the potential impacts of quantum annealing in shaping the future of machine learning.

A. AIM AND NOVELTY

The principal aim of this work is to rigorously evaluate the performance of quantum annealing machine learning (QAML) methods, with a particular focus on Quantum Support Vector Machines (QSVM) and QBoost, using ADMET datasets. It is important to emphasize that this study is solely concerned with quantum annealing approaches, which differ significantly from gate-based quantum methods. ADMET datasets, which are crucial in drug discovery, provide an ideal test case for assessing the practical applicability of quantum annealing machine learning techniques. By comparing the performance of QAML models against classical machine learning methods, this study aims to fill the gap in performance evaluations of quantum annealing in real-world applications, focusing exclusively on its potential within this domain.

The novelty of this research is multi-faceted, as outlined below:

1) **Comprehensive Experimental Comparison**: This study provides an in-depth experimental comparison between quantum annealing-based machine learning models (QBoost and QSVM) and traditional models. By employing real-world datasets, we offer a meaningful analysis of the strengths and weaknesses of each approach.
2) **Exploration of ADMET Datasets**: The application of QAML methods to ADMET datasets, which are pivotal in the drug discovery process, represents a novel contribution to the field of quantum machine learning. This area remains largely unexplored, making this work particularly innovative.
3) **Advanced Quantum Hardware Integration**: Utilizing D-Wave's Hybrid Quantum Processing Unit (HQPU) and the Fast Anneal feature, this study not only evaluates the potential of quantum annealers but also provides critical insights into their limitations, offering a nuanced understanding of how these technologies may be harnessed to advance machine learning.

B. PAPER STRUCTURE

The paper is organized as follows:

First, the foundational concepts of D-Wave Quantum Processing Units (QPUs) are examined, focusing on their architecture and fast annealing capabilities. This includes a detailed overview of the hybrid quantum-classical solver (HQPU) and its role in quantum computing. Next, the research problem is defined, emphasizing the complexities of processing SMILES representations from ADMET datasets and the challenge of feature extraction for machine learning applications.

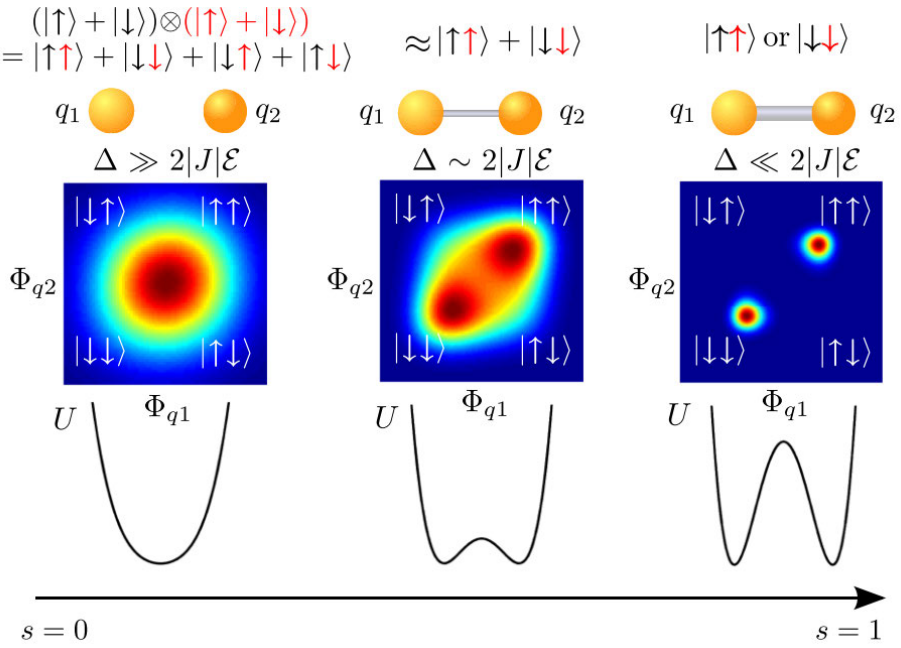


FIGURE 3. Entanglement between two qubits during quantum annealing with $h_i = 0$ and $J < 0$. Adapted from [46].

Following this, the methodology for standardizing SMILES strings and performing feature extraction is described, which forms the basis for both classical and quantum machine learning models. Subsequently, a classical machine learning model, LightAutoML, is implemented as a baseline, and its performance is evaluated using the preprocessed ADMET datasets. The paper then explores quantum machine learning approaches, specifically Quantum Support Vector Machines (QSVM) and QBoost, which utilize the quantum annealing capabilities of D-Wave systems. These methods are applied to benchmarking datasets, providing a comparison to classical techniques. An experimental section presents the setup and outcomes, highlighting the performance metrics used to evaluate both classical and quantum approaches. The advantages and limitations of each method are analyzed.

This is followed by a discussion of the results, exploring the broader implications of quantum computing for machine learning, along with potential future research directions. The paper concludes by summarizing the key findings and reflecting on the performance of the D-Wave Advantage 2 system, particularly its impact on advancing machine learning methodologies through quantum annealing.

II. D-WAVE QUANTUM MACHINES

The development of D-Wave quantum computers represents a series of groundbreaking advancements in quantum computing architecture, with key milestones marking the evolution of their systems. The first commercially produced D-Wave processor was a programmable, superconducting integrated circuit consisting of up to 128 pair-wise coupled superconducting flux qubits [58], [59]. Figure 4 illustrates the architecture of superconducting flux qubits, which allowed for the superposition of quantum states, laying the foundation for subsequent developments in quantum annealing technology.

The conceptual foundations of D-Wave's approach are rooted in pioneering experiments in condensed matter physics, specifically in the study of quantum annealing in magnetic systems. This line of research, led by Gabriel Aeppli, Thomas Felix Rosenbaum, and their collaborators, provided a crucial basis for later developments in quantum computation [61]. These early insights were further refined and applied to quantum computing by physicists Edward Farhi, Seth Lloyd, Terry Orlando, and Bill Kaminsky at the Massachusetts Institute of Technology (MIT). Their work ultimately culminated in the design of D-Wave's quantum annealing architecture, which is built around superconducting flux qubits [62], [63], [64], [65].

In 2007, D-Wave publicly demonstrated its Orion prototype, the first quantum computer unveiled to the public [66]. The subsequent release of the D-Wave One, equipped with a 128-qubit processor, represented a significant leap forward in quantum computational capabilities. Despite early skepticism and debates regarding the practical realization of quantum speedup, collaborations with key partners, such as Lockheed Martin, and research breakthroughs, such as applications in protein structure determination, demonstrated the practical potential of D-Wave's quantum processors [67].

The launch of the D-Wave Two in 2013, featuring a 512-qubit processor, further cemented D-Wave's place in the quantum computing landscape. Comparative studies between

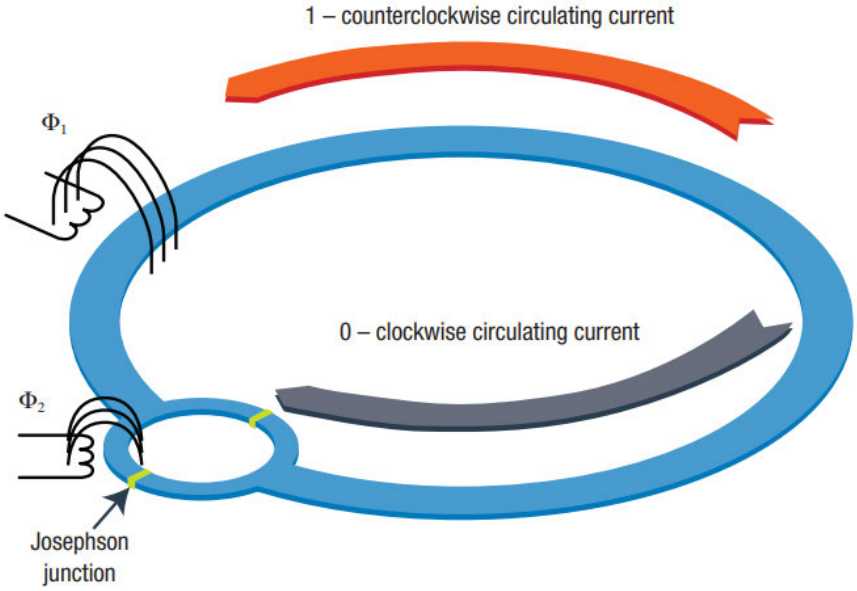


FIGURE 4. Superconducting flux qubit showing the superposition of two quantum states with different magnetic flux. Adapted from [60].

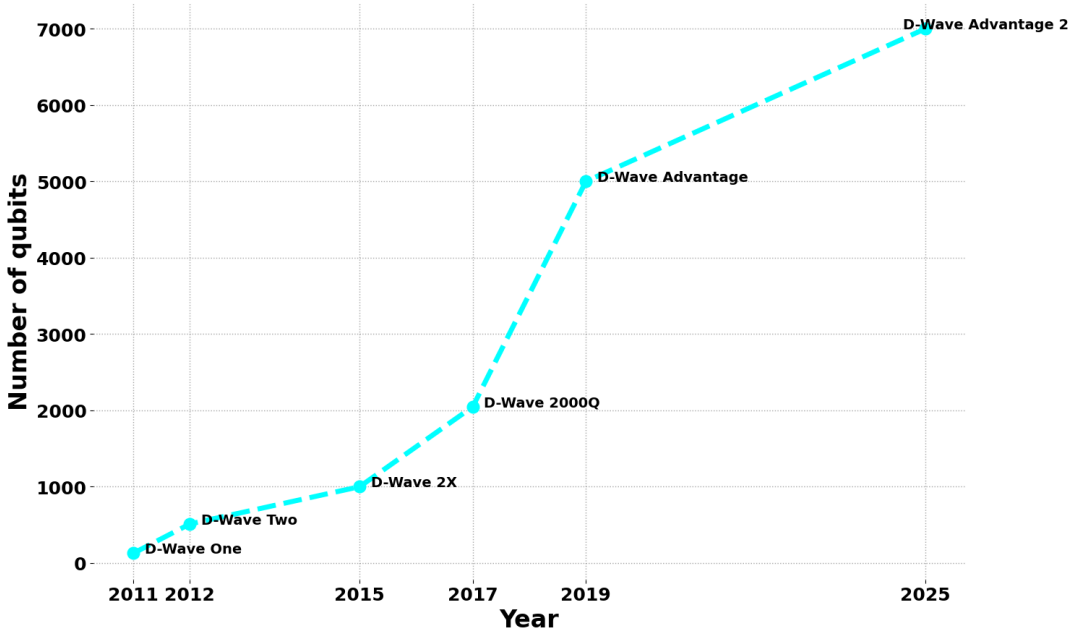


FIGURE 5. Evolution of D-Wave Quantum Computers: Number of Qubits Over Time.

D-Wave's technology and classical algorithms revealed significant performance improvements in solving complex optimization problems. Additionally, collaborations with institutions like NASA and Google underscored the practical applications of D-Wave's quantum solutions, especially in fields such as machine learning and large-scale optimization [68], [69], [70].

In 2015, D-Wave introduced the D-Wave 2X system, which featured a 1000-qubit processor based on the Chimera graph architecture [71]. The Chimera qubits within this architecture are defined by a nominal length of 4, where each qubit is connected to four orthogonal qubits via internal couplers, and a degree of 6, indicating that each qubit is coupled to six distinct qubits [72], [73]. Figure 6 illustrates this Chimera graph structure. The D-Wave 2X system achieved notable performance improvements, particularly constant-factor gains over classical hardware, highlighting the advantages of D-Wave's quantum annealing methodology

for specific computational challenges. In 2017, D-Wave advanced this design further with the release of the D-Wave 2000Q system, maintaining the Chimera graph architecture while expanding the qubit count to 2048 and enhancing its quantum annealing capabilities [74], [75].

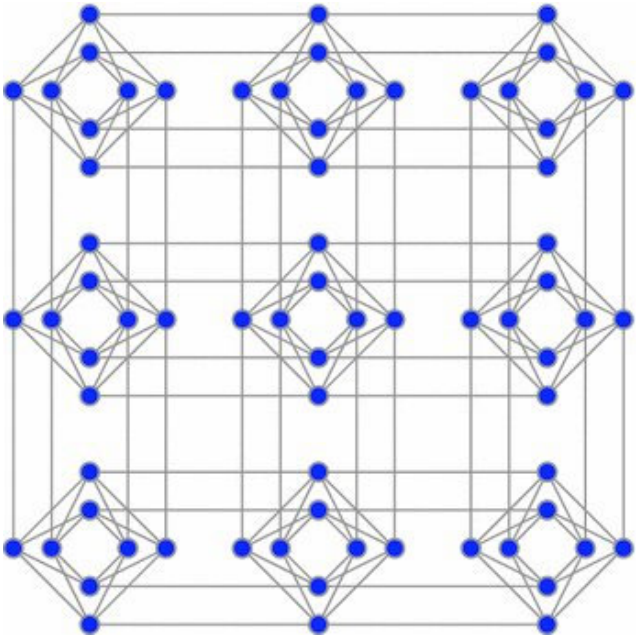


FIGURE 6. Chimera Topology in D-Wave Quantum Annealers. Adapted from [76].

D-Wave's most advanced architecture, the Advantage system, features over 5000 qubits and marks a transition to the Pegasus graph topology. Figure 5 illustrates the historical growth in the number of qubits in D-Wave quantum processors, with projections indicating that the forthcoming Advantage 2, anticipated for release in 2025, will feature more than 7000 qubits [77]. The Pegasus graph topology, introduced with the Advantage architecture, represents a major innovation in quantum hardware design. Unlike its predecessor, Chimera, which exhibited limitations in qubit connectivity and subgraph structures, the Pegasus topology offers superior connectivity and more intricate native subgraphs, making it better suited to a broader range of optimization and machine learning problems.

A key feature of the Pegasus topology is its enhanced connectivity compared to the Chimera topology [79]. In Chimera, qubits were organized into unit cells with limited interconnections [80]. Pegasus improves on this by adding extra couplers, facilitating more complex interactions between qubits (see Figures 7 and 8). This increased connectivity allows Pegasus-based quantum processors to tackle larger and more intricate problem sets. A notable aspect of the Pegasus topology is its native subgraphs, particularly the K_4 and $K_{6,6}$ structures. The K_4 subgraph represents a fully connected cluster of four qubits, while the $K_{6,6}$ subgraph corresponds to a fully connected set of twelve qubits. These

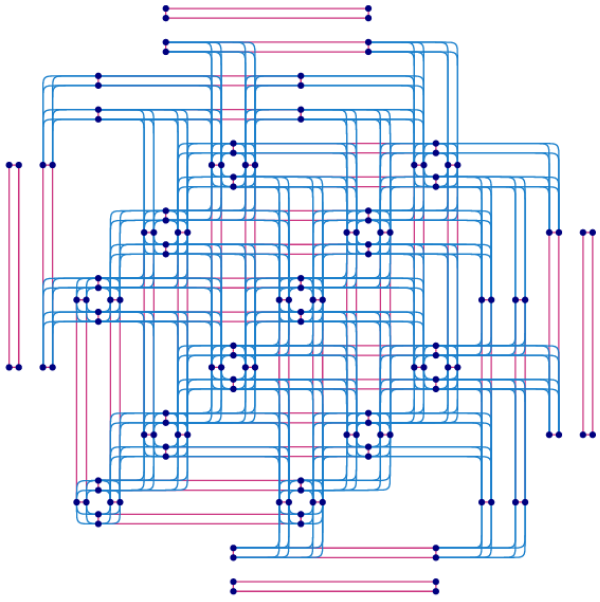


FIGURE 7. A P3-sized Pegasus(0) processor, showing internal couplers (blue curved lines), external couplers (long red lines), and odd couplers (short red lines).

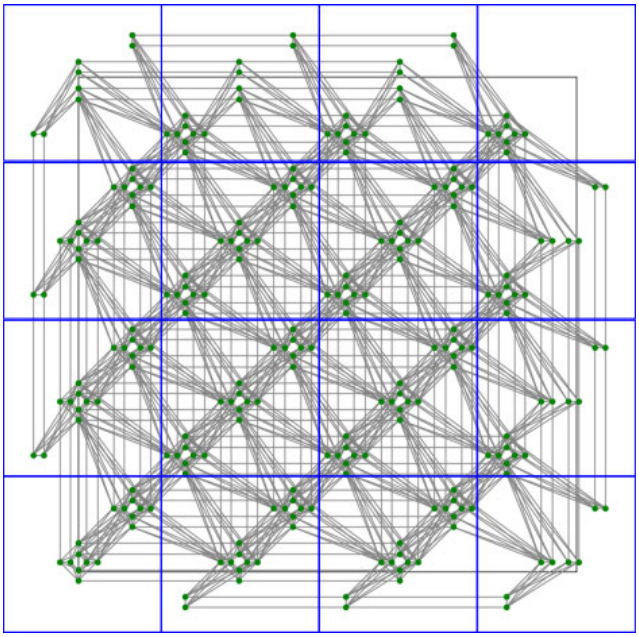


FIGURE 8. Pegasus unit cells in a P4 graph, with qubits represented as green dots and couplers as gray lines. Adapted from [78].

subgraphs are advantageous for algorithms that can exploit such configurations, boosting computational performance.

The Pegasus topology introduced by D-Wave represents a significant advancement in quantum computing architecture. One of its key innovations is the introduction of "odd couplers," which connect similarly aligned qubits, adding a new dimension of flexibility to quantum processing. These odd couplers enhance the system's capacity to perform

more sophisticated calculations and optimizations, thereby enabling quantum processors to tackle a wider range of complex optimization problems with greater efficiency. The transition to the Pegasus graph topology, with its increased connectivity and advanced native subgraphs, marks a critical leap in D-Wave's quantum computing capabilities, significantly pushing the boundaries of performance and computational efficiency.

In conjunction with the Pegasus topology, D-Wave has also introduced fast annealing, a critical feature that has further expanded the scope of quantum annealing. Fast annealing has played a pivotal role in D-Wave's research achievements [57], [81]. These publications highlight the advantages of quantum annealing over classical algorithms, particularly in solving complex optimization problems. Fast annealing refers to the reduction of annealing time scales to a point where thermal excitations are rendered negligible. This ensures that the quantum system remains in a coherent state throughout the process, enhancing the precision and accuracy of the solutions obtained.

As illustrated in Figure 9, fast annealing provides more granular control over quantum systems by extending the duration over which quantum coherence is maintained. This, in turn, prevents interference from thermal dynamics and allows quantum processors to maintain a higher degree of accuracy during computations. Such advancements are essential for achieving quantum supremacy, where quantum devices decisively outperform classical computing systems on specific tasks.

The combination of the Pegasus topology's enhanced qubit connectivity and odd couplers, alongside the integration of fast annealing, represents a monumental step forward in the trajectory of quantum computing innovation. Together, these features not only improve the performance and efficiency of D-Wave's quantum processors but also enable the exploration of new frontiers in quantum optimization, simulation, and supremacy.

A. KEY BENEFITS OF FAST ANNEALING

- **Negligible Thermal Excitation**: The fast annealing process operates within time scales that minimize thermal effects, thereby making thermal excitations negligible. This improvement allows the system to stay closer to its ground state, ensuring more accurate results in solving complex optimization problems.
- **Quantum Supremacy Applications**: Fast annealing enhances the performance of quantum systems, providing users the capability to build applications that leverage quantum supremacy. Quantum supremacy refers to solving specific problems that are infeasible for classical computers.
- **Enhanced Quantum Simulations**: The introduction of fast annealing allows researchers to develop increasingly sophisticated quantum simulations. These simulations can address more intricate problems in fields such as materials science, cryptography, and artificial intelligence.
- **Exploration of Coherent Dynamics**: The improved time scales provided by fast annealing also enable the exploration of coherent quantum dynamics in core performance, which plays a crucial role in the future scalability of quantum systems.

This development marks a significant leap in quantum computing technology, providing users with the tools necessary to advance their research in quantum supremacy and quantum dynamics.

III. PROBLEM STATEMENT

In drug development and research, a critical task is the prediction of drug permeability and activity across various biological barriers and systems, such as PAMPA (parallel artificial membrane permeability assay), the blood-brain barrier (BBB), and human intestinal absorption (HIA). Additionally, predicting drug interactions with proteins like P-glycoprotein (Pgp) and various cytochrome P450 enzymes (CYP2C9, CYP3A4, etc.) is essential for evaluating drug safety, efficacy, and potential toxicological effects.

This project aims to address several binary classification tasks:

- Predicting the permeability of compounds in the PAMPA assay.
- Determining the activity of drugs against the blood-brain barrier.
- Predicting human intestinal absorption based on chemical structure.
- Identifying P-glycoprotein inhibition and interactions with cytochrome P450 enzymes (CYP2C9, CYP3A4).
- Assessing the mutagenicity, carcinogenicity, and liver injury potential of drugs.

These classification tasks involve a wide variety of datasets, each with specific biological or chemical endpoints, and accurate models are needed to predict these properties from chemical descriptors such as SMILES strings.

We will solve all these classification tasks using classical and quantum annealing machine learning methods. The datasets considered include: PAMPA NCATS, HIA Hou, Pgp Broccatelli, Bioavailability Ma, BBB Martins, CYP2C9 Substrate CarbonMangels, CYP3A4 Substrate CarbonMangels, hERG, DILI, and Carcinogens Lagunin. For detailed information on each dataset, refer to A.

IV. STANDARDIZATION AND FEATURE EXTRACTION OF SMILES IN ADMET DATASETS

In cheminformatics, rigorous preprocessing and standardization of molecular representations are indispensable for ensuring the integrity and robustness of computational analyses. This study systematically addresses the standardization and feature extraction of SMILES (Simplified Molecular Input Line Entry System) strings derived from ADMET datasets [82], [83], [84]. Advanced chemical informatics

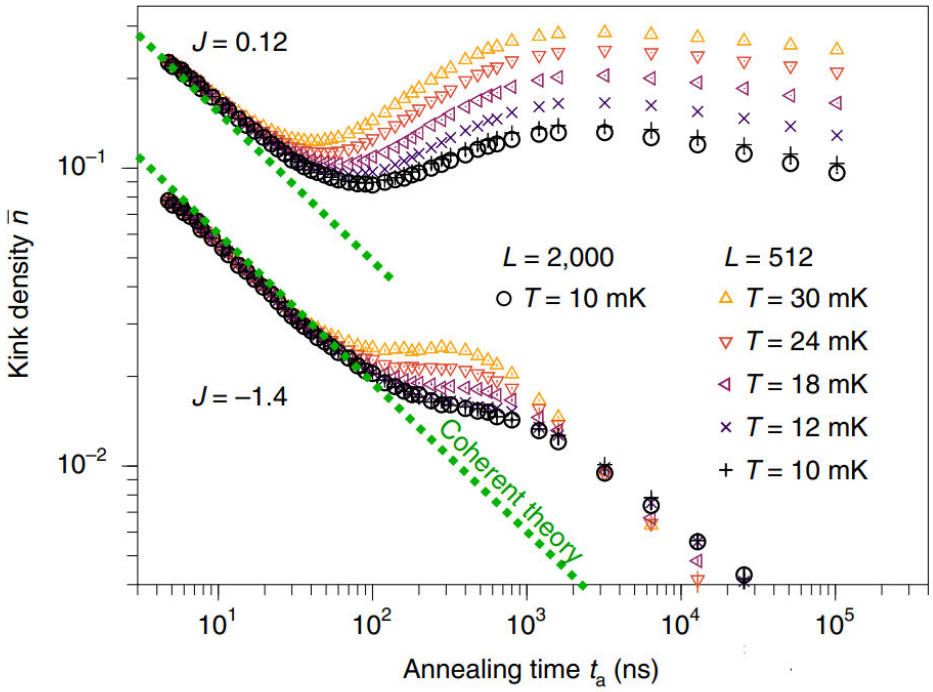


FIGURE 9. Advancements in fast annealing within quantum computing, highlighting its effectiveness in suppressing thermal excitations and its role in enhancing capabilities towards achieving quantum supremacy. Adapted from [81].

tools and mathematical frameworks were employed to enhance the representational accuracy and feature richness of the molecular data.

A. SMILES STANDARDIZATION

The standardization of SMILES strings involves transforming the raw molecular data into a canonical form that is both chemically valid and structurally consistent across the dataset. This process was carried out using the `datamol` and `RDKit` libraries, which provide a suite of functions for correcting and sanitizing SMILES strings [85], [86], [87].

1) **Error Correction**: For each molecule M_i represented by a SMILES string $S(M_i)$, we define an error correction function $E : S(M_i) \rightarrow S'(M_i)$, where $S'(M_i)$ is the corrected SMILES string. This function detects and resolves syntactical and chemical errors by enforcing chemical valence rules and correcting atom and bond annotations.
2) **Salt Removal**: SMILES may contain cations or anions that do not contribute to the chemical structure of the molecule, but increase molecular weight and therefore need to be stripped. The salt removal process involves decomposing a molecule M_i into its constituent fragments $\{F_{i1}, F_{i2}, \ldots, F_{in}\}$ and isolating the primary fragment F_{ip} that represents the core neutral molecule. This is formalized by a function $R : S(M_i) \rightarrow S(F_{ip})$, where R removes extraneous fragments based on their molecular weight and charge distribution.
3) **Sanitization**: Sanitization is the process of transforming each molecule M_i into a chemically valid form $M_i^{\text{sanitized}}$ by applying a series of rules $\mathcal{S} = \{s_1, s_2, \ldots, s_m\}$ that ensure correct valence and stereochemistry, try to catch potential aromaticity errors, fix aromatic nitrogen and remove dummy atoms. This can be mathematically represented as:

$$M_i^{\text{sanitized}} = \mathcal{S}(M_i) = \left(\prod_{j=1}^{m} s_j \right) M_i \tag{5}$$

where $\prod_{j=1}^{m} s_j$ denotes the sequential application of sanitization rules.
4) **Standardization**: To ensure a unique representation, each molecule M_i is mapped to a canonical form $C(M_i)$ through a canonicalization function $C : M_i \rightarrow \mathcal{C}(M_i)$. The canonical form $\mathcal{C}(M_i)$ is invariant under isomorphisms of the molecular graph $G(M_i)$, thus eliminating redundancy in molecular representations. It consists of disconnecting metals from non-metals, correcting functional groups, recombining charges, neutralizing the molecule to ensure the net charge of 0 and adding stereochemical information.

B. FEATURE EXTRACTION

Following the standardization, molecular features were extracted using a comprehensive array of algorithms from the `molfeat` library. The extracted features encompass both classical molecular descriptors and advanced graph-

based embeddings, enabling a thorough characterization of molecular structures. The feature extraction process involves mapping each molecule M_i to a high-dimensional feature space $\mathbb{R}^k$, where k is the dimensionality of the feature vector.

- **desc2D**: The 2D molecular descriptors are computed by mapping each molecule M_i to a vector $\mathbf{D}_{2D}(M_i) \in \mathbb{R}^n$, where n represents the number of calculated descriptors. Each descriptor $D_{2D,j}(M_i)$ is a function of the molecular graph $G(M_i)$ and is defined as:

$$D_{2D,j}(M_i) = f_j(G(M_i)) \tag{6}$$

  where f_j is a descriptor function corresponding to properties such as molecular weight or topological indices [88].
- **GIN (Graph Isomorphism Network) Supervised Models**:
  - **ContextPred**: This model extracts features by maximizing a contextual prediction objective. For each node v_i in the molecular graph $G(M_i)$, the feature $h_i^{(k)}$ at layer k is updated by an aggregation function:

$$h_i^{(k)} = \sigma\left(W^{(k)} \sum_{j \in \mathcal{N}(i)} h_j^{(k-1)} + b^{(k)}\right) \tag{7}$$

    where $\mathcal{N}(i)$ denotes the neighbors of node v_i, $W^{(k)}$ and $b^{(k)}$ are trainable parameters, and σ is an activation function [89].
  - **EdgePred**: The edge prediction task involves estimating the probability $P(E(v_i, v_j) = 1 | G(M_i))$ for an edge $E(v_i, v_j)$ between nodes v_i and v_j [90]. The extracted features are optimized to maximize the log-likelihood:

$$\mathcal{L}_{\text{edge}} = \sum_{(v_i,v_j)\in E} \log P(E(v_i, v_j) = 1) + \sum_{(v_i,v_j)\notin E} \log\left(1 - P(E(v_i, v_j) = 1)\right) \tag{8}$$

  - **Infomax**: The Infomax [91] model extracts features by maximizing the mutual information $I(G; H)$ between global graph-level representations H and local node-level representations h_i. This is achieved by optimizing the contrastive loss:

$$\mathcal{L}_{\text{info}} = -\sum_{i=1}^{n} \log \frac{\exp(\phi(h_i, H))}{\sum_{j=1}^{n} \exp(\phi(h_j, H))} \tag{9}$$

    where $\phi(h_i, H)$ is a scoring function measuring the agreement between h_i and H.
  - **Masking**: In this model, random nodes or edges in the graph $G(M_i)$ are masked, and the model is trained to reconstruct the masked components. The loss function $\mathcal{L}_{\text{mask}}$ penalizes incorrect reconstructions, promoting features that capture the underlying graph structure [89].
- **Mordred**: The Mordred descriptors span a wide range of molecular properties, represented by a high-dimensional vector $\mathbf{F}_{\text{Mordred}}(M_i) \in \mathbb{R}^m$, where m is the number of descriptors. Each descriptor is a functional f_k applied to the graph $G(M_i)$:

$$F_{\text{Mordred},k}(M_i) = f_k(G(M_i)) \tag{10}$$

- **pcqm4mv2_graphormer_base**: This model leverages a transformer architecture with graph-specific encodings. The attention mechanism computes pairwise dependencies between nodes, and the features $\mathbf{F}_{\text{Graphormer}}(M_i)$ are derived from the attention-weighted sum of node embeddings:

$$\mathbf{F}_{\text{Graphormer}}(M_i) = \sum_{i=1}^{n} \alpha_i \mathbf{h}_i \tag{11}$$

  where α_i are attention coefficients, and $\mathbf{h}_i$ are node embeddings [92].
- **RDKit**: Traditional molecular descriptors from RDKit map each molecule M_i to a feature vector $\mathbf{F}_{\text{RDKit}}(M_i)$. These descriptors are defined by explicit chemical rules and function g_j:

$$F_{\text{RDKit},j}(M_i) = g_j(G(M_i)) \tag{12}$$

- **SECFP (Self-Attention Contextualized Fingerprints)**: SECFP features are derived by applying a self-attention mechanism $\mathcal{A}$ to molecular fingerprints [93]. The attention mechanism dynamically weights substructure contributions, resulting in a contextually enriched feature vector $\mathbf{F}_{\text{SECFP}}(M_i)$:

$$\mathbf{F}_{\text{SECFP}}(M_i) = \mathcal{A}(\text{fingerprint}(M_i)) \tag{13}$$

These feature extraction techniques, grounded in both classical and contemporary mathematical models, provide a robust and high-dimensional representation of molecular structures. The combination of standardized SMILES and advanced feature extraction enables the development of predictive models that are both accurate and computationally efficient, facilitating the exploration of ADMET properties.

V. CLASSICAL APPROACH: LIGHTAUTOML

The classical machine learning approach utilizes `LightAutoML` [94], [95], [96], an automated machine learning (AutoML) framework designed to handle model training, tuning, and blending systematically and efficiently. The process begins with a k-fold cross-validation scheme, where the dataset $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, with n samples, is partitioned into k disjoint subsets, denoted as $\{D_1, D_2, \ldots, D_k\}$, such that:

$$D = \bigcup_{i=1}^{k} D_i, \quad D_i \cap D_j = \emptyset \text{ for } i \neq j \tag{14}$$

For each fold i, the model is trained on the training set $D_{\text{train}} = D \setminus D_i$ and validated on the holdout set D_i. Let $\mathcal{M}_m$ represent the m-th model in the ensemble. Each model

$\mathcal{M}_m$ minimizes a loss function $L_m(\hat{y}, y)$, which depends on the task (e.g., $L_m(\hat{y}, y) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$ for regression or cross-entropy loss for classification).

A. MODEL BLENDING AND ENSEMBLE FORMATION

The blending process involves combining predictions from a set of models $\{\mathcal{M}_1, \ldots, \mathcal{M}_M\}$. Let $\hat{y}_i^m = \mathcal{M}_m(x_i)$ denote the out-of-fold prediction of the m-th model on the i-th sample from fold D_i. The final blended prediction $\hat{y}_i$ is computed as a weighted sum of the individual models' predictions:

$$\hat{y}_i = \sum_{m=1}^{M} w_m \hat{y}_i^m \tag{15}$$

where $w_m \geq 0$ is the weight assigned to the m-th model, with the constraint:

$$\sum_{m=1}^{M} w_m = 1 \tag{16}$$

These weights w_m are typically determined by minimizing the out-of-fold validation error across all models. The optimization problem can be formulated as:

$$\min_{\{w_m\}} \frac{1}{n}\sum_{i=1}^{n} L\left(\sum_{m=1}^{M} w_m \mathcal{M}_m(x_i), y_i\right) \tag{17}$$

subject to the constraints:

$$\sum_{m=1}^{M} w_m = 1 \quad \text{and} \quad w_m \geq 0 \tag{18}$$

This forms a convex optimization problem, which ensures that the blended model minimizes the expected loss across the validation folds.

B. MODEL STACKING WITH META-LEARNERS

In more advanced settings, `LightAutoML` may also employ model stacking, where the predictions $\hat{y}_i^m$ from each base model $\mathcal{M}_m$ are used as inputs to a meta-learner model $\mathcal{M}_{\text{meta}}$. The meta-learner's task is to learn the optimal combination of the base models' outputs. Let $\hat{\mathbf{y}}_i = [\hat{y}_i^1, \hat{y}_i^2, \ldots, \hat{y}_i^M]$ be the vector of predictions from the base models for sample x_i. The meta-learner $\mathcal{M}_{\text{meta}}$ makes the final prediction as:

$$\hat{y}_i = \mathcal{M}_{\text{meta}}(\hat{\mathbf{y}}_i) \tag{19}$$

The meta-learner is trained by minimizing a loss function L_{meta} on the out-of-fold predictions of the base models:

$$\min_{\mathcal{M}_{\text{meta}}} \frac{1}{n}\sum_{i=1}^{n} L_{\text{meta}}(\mathcal{M}_{\text{meta}}(\hat{\mathbf{y}}_i), y_i) \tag{20}$$

This meta-learning process allows for a non-linear combination of the base models' predictions, potentially capturing more complex interactions between models and improving overall performance.

C. REGULARIZATION AND OVERFITTING PREVENTION

To prevent overfitting, `LightAutoML` typically incorporates regularization into the model blending or stacking process. A common approach is to apply L_2 regularization on the model weights:

$$\min_{\{w_m\}} \frac{1}{n}\sum_{i=1}^{n} L\left(\sum_{m=1}^{M} w_m \mathcal{M}_m(x_i), y_i\right) + \lambda \sum_{m=1}^{M} w_m^2 \tag{21}$$

where $\lambda > 0$ is a regularization parameter that controls the tradeoff between minimizing the loss and keeping the model weights small, thereby avoiding overfitting to the validation folds.

This method allows `LightAutoML` to construct a flexible, powerful, and robust model ensemble, leveraging the strengths of various algorithms such as linear models, `CatBoost`, and `LightGBM`, and optimizing them for superior performance across a wide range of datasets.

D. RATIONALE FOR CHOOSING LIGHTAUTOML

The decision to use `LightAutoML` in this project is motivated by several key factors. First, `LightAutoML` excels in automating the entire machine learning pipeline, including data preprocessing, model selection, hyperparameter tuning, and model evaluation [94]. This automation reduces the need for manual intervention, thereby accelerating the model development process and ensuring a streamlined workflow. For large-scale datasets, this framework's efficient handling of both computational resources and time is invaluable [97].

Additionally, one of the major strengths of `LightAutoML` is its ability to create highly robust model ensembles. By employing advanced techniques like model blending and stacking, as described in the previous sections, the framework is capable of capturing complex relationships in the data and producing superior predictive performance compared to single models. The regularization techniques incorporated in the blending process further mitigate overfitting, ensuring that the resulting models generalize well to unseen data [98].

`LightAutoML` also seamlessly integrates a wide variety of machine learning algorithms, including linear models, tree-based models (such as `CatBoost` and `LightGBM`), and neural networks. This flexibility allows the framework to tailor its solutions to the nature of the problem and dataset, ensuring that the most appropriate and performant models are selected. Another significant advantage is the scalability of `LightAutoML`, which is essential when working with large datasets or complex models. The framework's ability to parallelize operations across multiple folds of cross-validation, as well as its efficient use of computing resources, makes it an ideal choice for tasks requiring high computational power. Moreover, by utilizing k-fold cross-validation, `LightAutoML` ensures that model performance is rigorously evaluated across multiple splits of the data. This prevents the model from being overly dependent on

any particular portion of the dataset, thus providing a more reliable estimate of its generalization capabilities.

Finally, the modularity of `LightAutoML` allows it to be easily integrated with existing workflows, enabling researchers and practitioners to leverage its capabilities without having to overhaul their entire system. This flexibility makes it a highly practical choice for a variety of machine learning tasks, particularly in environments where multiple models and configurations need to be tested efficiently.

In conclusion, `LightAutoML` was chosen for this project due to its comprehensive, automated approach to machine learning, robust performance through ensemble methods, flexibility in model selection, and scalable architecture. These characteristics make it a powerful tool for addressing the challenges posed by complex datasets while minimizing manual effort and computational overhead.

VI. QUANTUM ANNEALING APPROACHES: QUANTUM SUPPORT VECTOR MACHINES (QSVM) AND QBOOST

A. QSVM

The optimization process for Support Vector Machines (SVM) commences with the primary objective of maximizing the margin between classes [99], [100], [101]. Given a set of input vectors $\mathbf{x}_n$ and corresponding class labels t_n for $n = 1, 2, \ldots, N$, the goal is to identify a hyperplane that effectively separates the data while maximizing the margin. The equation of the hyperplane in a linear SVM is expressed as:

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{22}$$

Here, $\mathbf{w}$ represents the weight vector, $\mathbf{x}$ is the input vector, and b is the bias term.

The margin (γ) is defined as the perpendicular distance from the hyperplane to the nearest data point. Mathematically, the margin is given by:

$$\gamma = \frac{1}{\|\mathbf{w}\|} \tag{23}$$

The primary objective is to maximize this margin (γ).

To achieve this, the problem is formulated as a constrained optimization problem, where the margin is maximized under the constraint that all data points are correctly classified. This is expressed as the minimization of $\frac{1}{2}\|\mathbf{w}\|^2$ under the constraint:

$$t_n(\mathbf{w} \cdot \mathbf{x}_n + b) \geq 1 \quad \text{for } n = 1, 2, \ldots, N \tag{24}$$

To convert the inequality constraints into equality constraints, Lagrange multipliers $\alpha_n \geq 0$ are introduced for each data point:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} \alpha_n [t_n(\mathbf{w} \cdot \mathbf{x}_n + b) - 1] \tag{25}$$

Here, $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_N]$.

For the stationarity conditions, the partial derivatives of L with respect to $\mathbf{w}$ and b are set to zero to find the optimal values:

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{n=1}^{N} \alpha_n t_n \mathbf{x}_n = 0 \tag{26}$$

$$\frac{\partial L}{\partial b} = -\sum_{n=1}^{N} \alpha_n t_n = 0 \tag{27}$$

By solving for $\mathbf{w}$ and b in terms of the Lagrange multipliers α, and substituting the optimal values back into L, the dual form is obtained:

$$L(\alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m t_n t_m K(\mathbf{x}_n, \mathbf{x}_m) \tag{28}$$

If the data is not linearly separable, a kernel function $K(\mathbf{x}_n, \mathbf{x}_m)$ is introduced to compute dot products in a higher-dimensional space. Mercer's theorem substantiates the validity of this transformation, asserting that $K(\mathbf{x}_n, \mathbf{x}_m) = \phi(\mathbf{x}_n) \cdot \phi(\mathbf{x}_m)$, where $\phi(\mathbf{x})$ signifies the transformation function. This obviates the need for explicit computation of transformations, thereby enhancing computational efficiency.

Integral to the SVM optimization process is the inherent convexity of the objective function and constraints. The dual representation of the SVM problem is cast as a convex optimization problem. The objective function $L(\alpha)$ is concave concerning the Lagrange multipliers, while the constraints form convex sets. This ensures the existence of a unique and globally optimal solution for the SVM model.

The dual problem arises from maximizing the Lagrangian:

$$\text{Maximize } L(\alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m t_n t_m K(\mathbf{x}_n, \mathbf{x}_m) \tag{29}$$

Subject to:

$$0 \leq \alpha_n \leq C \quad \text{and} \quad \sum_{n=1}^{N} \alpha_n t_n = 0 \tag{30}$$

The optimality conditions, guided by the Karush-Kuhn-Tucker (KKT) framework, include:

1) **Stationarity of the gradient:**

$$\nabla L(\alpha) = \mathbf{t} - \sum_{m=1}^{N} \alpha_m t_m K(\mathbf{x}_n, \mathbf{x}_m) = 0. \tag{31}$$

2) **Complementary slackness:**

$$\alpha_n \big(t_n(\mathbf{w} \cdot \mathbf{x}_n + b) - 1\big) = 0. \tag{32}$$

These conditions guide the numerical optimization process.

The dual form of the SVM problem is cast as a Quadratic Programming (QP) problem:

$$\text{Minimize } E = \frac{1}{2}\alpha^T Q \alpha - \mathbf{1}^T \alpha \tag{33}$$

Subject to:

$$0 \leq \alpha_n \leq C \quad \text{and} \quad \mathbf{T}^T \alpha = 0 \tag{34}$$

Here, $\mathbf{1}$ is a vector of ones, and Q is the matrix of kernel evaluations.

The optimization process for Support Vector Machines (SVM) involves formulating a quadratic programming (QP) problem to find the optimal Lagrange multipliers $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_N]$ that maximize the objective function. However, traditional optimization methods yield real-valued solutions for α, which is incompatible with the discrete, binary nature of solutions produced by quantum annealers, which operate on QUBO problems.

1) QUBO FORMULATION OF SVM

The transformation of the Support Vector Machine (SVM) optimization problem into a QUBO framework involves a meticulous process of encoding continuous variables into binary representations and integrating the original constraints into an unconstrained objective function. This section formalizes this transformation, presenting key results that establish the theoretical foundation for the QUBO formulation of SVM [102], [103], [104], [105].

a: LEMMA AND THEOREM ON BINARY ENCODING AND QUBO TRANSFORMATION

Lemma 1 (Binary Encoding Lemma): Let α_n be a continuous variable constrained by $\alpha_n \in [0, C]$, where C is a constant. There exists a binary encoding $\mathbf{a}_n \in \{0, 1\}^K$ such that:

$$\alpha_n = \sum_{k=0}^{K-1} 2^k a_{n,k}, \quad \text{with} \quad a_{n,k} \in \{0, 1\}, \quad n \in \{1, \ldots, N\}, \tag{35}$$

where K is the number of binary digits required to approximate α_n within a desired precision, which increases exponentially with K.

Proof: Given the bounded variable $\alpha_n \in [0, C]$, it can be decomposed into a finite sum based on its binary representation. The binary vector $\mathbf{a}_n = (a_{n,0}, a_{n,1}, \ldots, a_{n,K-1})$ represents the binary coefficients corresponding to the powers of 2 in the sum, allowing the reconstruction of α_n with a precision governed by the choice of K. The maximum error of this representation is bounded by $\frac{C}{2^K}$, which diminishes as K increases. Hence, the lemma holds. ∎

Theorem 1 (QUBO Formulation Theorem): Consider the SVM primal optimization problem, subject to the constraints $\sum_{n=1}^{N} y_n \alpha_n = 0$ and $0 \leq \alpha_n \leq C$. The equivalent QUBO formulation is given by:

$$\min_{\mathbf{a}} \mathcal{L}(\mathbf{a}) = \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \sum_{k=0}^{K-1} \sum_{j=0}^{K-1} B_{k+j}\, a_{n,k} a_{m,j}\, y_n y_m K(x_n, x_m) - \sum_{n=1}^{N} \sum_{k=0}^{K-1} B_k a_{n,k} + \xi \left(\sum_{n=1}^{N} \sum_{k=0}^{K-1} B_k a_{n,k} y_n \right)^2, \tag{36}$$

where $\mathbf{a} \in \{0, 1\}^{NK}$ represents the binary encoding of the SVM dual variables α_n, and $\xi > 0$ is a penalty factor ensuring the enforcement of the constraint $\sum_{n=1}^{N} y_n \alpha_n = 0$.

Proof: The proof begins by encoding each continuous variable α_n into its binary representation $\mathbf{a}_n$ as described in Lemma 1. The constraint $\sum_{n=1}^{N} y_n \alpha_n = 0$ is then incorporated into the objective function using a quadratic penalty term weighted by ξ. The resulting objective function $\mathcal{L}(\mathbf{a})$ is a quadratic expression in the binary variables $\mathbf{a}$, with terms corresponding to the interaction of encoded variables and the kernel function $K(x_n, x_m)$. The resulting formulation adheres to the structure of a QUBO problem, confirming the theorem. ∎

Corollary 1 (Optimality Condition Corollary): If $\mathbf{a}^* \in \{0, 1\}^{NK}$ is a global minimizer of the QUBO objective function $\mathcal{L}(\mathbf{a})$, then the corresponding $\alpha_n^* = \sum_{k=0}^{K-1} 2^k a_{n,k}^*$ provides the optimal solution to the original SVM problem, satisfying the primal constraints.

Proof: Given that $\mathbf{a}^*$ minimizes $\mathcal{L}(\mathbf{a})$, the quadratic penalty term ensures that the original constraint $\sum_{n=1}^{N} y_n \alpha_n = 0$ is satisfied by the binary-encoded solution $\mathbf{a}^*$. The reconstructed α_n^* lies within the interval $[0, C]$, thereby respecting the original box constraints on α_n. Thus, α_n^* is a valid and optimal solution for the original SVM optimization problem, establishing the corollary. ∎

The formulation of the SVM problem as a QUBO problem is not only a theoretical exercise but also has practical implications for optimization techniques, especially in the emerging field of quantum computing. By transforming the SVM problem into a QUBO form, we enable its solution using quantum annealing and other quantum-inspired methods, which are particularly well-suited for high-dimensional optimization problems characterized by complex, non-linear interactions. This transformation also underscores the versatility of SVMs and their ability to adapt to different computational paradigms, providing a bridge between classical machine learning and cutting-edge quantum technologies. The rigorous formulation presented in this section provides a foundation for future research and development in the optimization of SVMs using QUBO frameworks, potentially leading to more efficient algorithms and deeper insights into machine learning models.

This symmetric matrix $\tilde{Q}$ is transformed into the upper-triangular Q matrix Q, ensuring compatibility with the QUBO formulation required for quantum annealing.

The final step involves the embedding procedure to map the QUBO problem onto the Advantage quantum annealer. This process combines physical qubits to form logical qubits, enhancing the connectivity between qubits and enabling the solution of larger problems on the quantum annealer.

The QSVM construction involves hyperparameters such as the encoding base B, the number of qubits per coefficient K, the multiplier ξ, and the kernel parameter γ. This formulation, denoted as $qSVM(B, K, \xi, \gamma)$, leverages quantum annealing to solve SVM optimization problems efficiently and effectively.

Algorithm 1 Quantum Annealing Support Vector Machine (SVM)

Input:
  B: Base coefficient for binary variables
  K: Number of binary variables per Lagrange multiplier α
  C: Regularization parameter
  γ: Kernel coefficient
  ξ: Offset for kernel function
  N: Number of training data points
  data: Training data set
  t: Target labels
Output:
  α: Optimized Lagrange multipliers
  b: Bias term
Function $\delta(i, j)$:
  Return $1 if\ i = j else 0$
Function `kernel`(x, y, γ):
  if $\gamma = -1$ **then**
    Return $\langle x, y \rangle$ {Linear kernel}
  else
    Return $\exp(-\gamma \|x - y\|^2)$ {Radial Basis Function (RBF) kernel}
Function `predict_class`$(x\_test, \alpha, b, data, t, \gamma)$:
  $f \leftarrow b + \sum_{n=0}^{N-1} \alpha[n] \cdot t[n] \cdot$ `kernel`$(data[n], x\_test, \gamma)$
  Return sign(f) {Predicted class}
Function `train_SVM`$(B, K, C, \gamma, \xi, N, data, t)$:
  Initialize Q matrix of size $(K \cdot N, K \cdot N)$
for $n = 0$ **to** $N - 1$ **do**
  for $m = 0$ **to** $N - 1$ **do**
    for $k = 0$ **to** $K - 1$ **do**
      for $j = 0$ **to** $K - 1$ **do**
        $Q[K \cdot n + k, K \cdot m + j] \leftarrow 0.5 \cdot (B^{k+j}) \cdot t[n] \cdot t[m]$
        $\cdot$(`kernel`$(data[n], data[m], \gamma) + \xi) - (\delta(n, m) \cdot \delta(k, j) \cdot B^k)$
      end for
    end for
  end for
end for
  Optimize α and b using D-Wave's quantum annealer
  Return α and b

B. QBOOST

QBoost [106], [107], [108] operates through an iterative process that combines quantum selection and classical adaptation to refine the ensemble of weak classifiers into a robust strong classifier. This algorithmic framework represents a significant advancement in machine learning methodologies, offering improved performance and efficiency in constructing accurate classifiers.

The algorithm begins by initializing an ensemble of weak classifiers and a labeled training dataset. In the quantum selection phase, Adiabatic Quantum Optimization (AQO) [109] facilitates efficient exploration of the classifier space by leveraging quantum tunneling effects, allowing QBoost to focus on configurations that offer the most significant performance improvements compared to classical selection methods. This quantum-inspired approach allows for efficient exploration of the classifier space, focusing on configurations that offer the most significant performance improvements. The quantum selection process involves encoding the weak classifiers and training data into a quantum representation suitable for optimization. By leveraging quantum annealing or other quantum-inspired techniques, QBoost explores different combinations of weak classifiers to identify those that contribute most effectively to the strong classifier's accuracy. This phase emphasizes the algorithm's ability to leverage quantum computing principles for efficient selection and optimization, contributing to its overall effectiveness.

Following the quantum selection phase, QBoost adapts its dictionary of weak classifiers based on the errors identified by the strong classifier constructed in the previous step. This adaptation process is crucial for fine-tuning the ensemble, incorporating new classifiers that address specific error patterns and enhance overall classification accuracy.

Metrics such as error rates, validation accuracy, and convergence criteria guide the adaptation process, ensuring that weak classifiers contributing positively to error reduction are given higher importance while those causing significant errors are downgraded or replaced. Weak classifiers that contribute positively to reducing errors are given higher importance, while those causing significant errors are downgraded or replaced. This adaptive dictionary ensures that the ensemble evolves dynamically, optimizing its composition to handle complex classification tasks effectively. The inner loop of QBoost involves the dynamic refinement of the strong classifier. Through iterative training and evaluation, the algorithm continuously assesses and updates the composition of the strong classifier based on error analysis and convergence criteria. This dynamic approach ensures that the final classifier achieves optimal performance while minimizing computational overhead.

During the refinement process, QBoost evaluates the performance of the strong classifier on both training and validation datasets. It monitors changes in error rates, convergence of optimization objectives, and the stability of the classifier's predictions. If the validation error decreases or converges within predefined thresholds, the algorithm updates the strong classifier with new weak classifiers or modified weights. Conversely, if the validation error stagnates or increases, indicating potential overfitting or lack of improvement, the algorithm adjusts its strategies, such as freezing the current classifier and exploring alternative configurations. The dynamic refinement phase of QBoost ensures that the strong classifier evolves iteratively, adapting

Algorithm 2 QBoost Algorithm

Require: Training and validation data, dictionary of weak classifiers, Q (maximum number of weak classifiers optimized in one step), T (total number of weak classifiers)

Ensure: Strong classifier $H(x)$

Initialize weight distribution $d(s) = \frac{1}{S}$ for all samples $s = 1, \ldots, S$

Set $T_{\text{strong}} = 0$, $c(x) = 0$, and $K = \emptyset$

repeat

  if $T_{\text{strong}} < T$ **then**

    Select a set K of $\min(Q, T - T_{\text{strong}})$ weak classifiers from the dictionary based on the current weights $d(s)$

  end if

  for each regularization parameter λ from $\lambda_{\min}$ to $\lambda_{\max}$ **do**

    Solve for $w(\lambda)$:

$$w(\lambda) = \arg\min_{w} \sum_{s=1}^{S} \left(c(x_s) + \frac{1}{Q} \sum_{t \in K} w_t h_t(x_s) - y_s \right)^2 + \lambda w_0$$

    Update $T_{\text{new}}(\lambda) = w(\lambda)$

    Construct the classifier $H(x; \lambda)$:

$$H(x; \lambda) = \text{sign}\left(c(x) + \sum_{t \in K} w_t(\lambda) h_t(x) \right)$$

    Compute validation error $\text{Error}_{\text{val}}(\lambda)$ for $H(x; \lambda)$ on the validation set

  end for

  Select $\lambda^* = \arg\min_{\lambda} \text{Error}_{\text{val}}(\lambda)$

  Update $T_{\text{strong}} = T_{\text{strong}} + T_{\text{new}}(\lambda^*)$

  Update $c(x) = c(x) + \sum_{t \in K} w_t(\lambda^*) h_t(x)$

  Update sample weights:

$$d(s) = \left(\frac{1}{T_{\text{strong}}} \sum_{t \in K} h_t(x_s) - y_s \right)^2$$

  Normalize the weights:

$$d(s) = \frac{d(s)}{\sum_{s=1}^{S} d(s)}$$

until Validation error $\text{Error}_{\text{val}}$ stops decreasing

to changing data patterns, minimizing errors, and improving generalization capabilities. The following pseudocode outlines the core steps of QBoost algorithm, highlighting its quantum-inspired selection, dictionary adaptation, and dynamic refinement processes.

This structured pseudocode encapsulates the QBoost, showcasing its quantum-inspired selection mechanisms, adaptive dictionary refinement, and dynamic training regimen.

VII. EXPERIMENT AND RESULTS

Upon extracting all relevant features, we employed the Quantum Advantage Annealer 4.1, utilizing the advanced Pragsus architecture, to carry out quantum annealing. Due to the problem's complexity and the high dimensionality of the matrices—exceeding the 177-variable limit [110]—we opted for the hybrid quantum solver. This approach leverages the solver's ability to partition the overall problem into smaller, more tractable subproblems using state-of-the-art graph partitioning algorithms. These subproblems are then embedded into the quantum hardware for execution, enabling the simultaneous exploitation of both quantum and classical computational resources. This hybrid technique effectively handles large-scale optimization problems, ensuring efficient and scalable solutions refer to B.

A. METRICS SELECTION: WHY USE AUC-ROC?

The primary metric chosen for evaluating the models' performance is the Area Under the Curve - Receiver Operating Characteristic (AUC-ROC). AUC-ROC was selected due to its robustness in assessing the overall performance of classification models, particularly in imbalanced datasets. It offers several advantages over other metrics such as accuracy, precision, or recall, which can be misleading when dealing with imbalanced data distributions. AUC-ROC evaluates a model's ability to distinguish between classes across all possible thresholds, making it ideal for applications where the costs of false positives and false negatives differ. It is also worth noting that AUC-ROC is widely regarded as a standard and effective measure for the performance of classification models in tasks such as those involving ADMET datasets. Given that AUC-ROC is a well-established and commonly used metric in these types of evaluations, we believe that it sufficiently captures the relevant performance aspects for the quantum annealing machine learning methods presented in our work.

Mathematically, the AUC-ROC is computed by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The TPR, also known as recall or sensitivity, is defined as:

$$\text{TPR} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{37}$$

Similarly, the FPR is defined as:

$$\text{FPR} = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}} \tag{38}$$

The ROC curve plots TPR against FPR, and the area under this curve (AUC) provides a single scalar value that summarizes the model's ability to discriminate between positive and negative classes across all thresholds. A perfect model would achieve an AUC of 1, while a model with no discriminative ability would score an AUC of 0.5. By using the AUC-ROC, we can evaluate the trade-off between sensitivity and specificity, ensuring that models are assessed across a wide range of decision boundaries, providing a more holistic performance analysis.

B. PERFORMANCE EVALUATION

To systematically evaluate the performance of quantum annealing-based machine learning methods—specifically Quantum Boosting (QBoost) and Quantum Support Vector Machines (QSVM)—we conducted a series of rigorous experiments utilizing various ADMET datasets. The evaluation metrics employed AUC-ROC, which serves as a robust indicator of a model's discriminative capability between positive and negative instances.

Both quantum annealing-based methods, QBoost and QSVM, were comprehensively assessed against classical machine learning methodologies, utilizing LightAutoML as a benchmark. The AUC-ROC scores for the evaluated datasets are summarized in Table 1. The empirical results indicate that classical machine learning models generally surpassed their quantum counterparts in performance. Notably, QBoost achieved the highest AUC-ROC score of 0.8614 on the Carcinogens_Lagunin dataset, illustrating the potential of quantum annealing techniques to enhance model efficacy. Although QSVM did not consistently outperform classical algorithms, it did demonstrate competitive performance relative to QBoost in specific instances, suggesting that both models warrant further investigation and development within the domain of quantum machine learning.

In addition to performance metrics, we provide an overview of dataset characteristics in Table 2, which includes critical information such as the number of training examples, positive cases, negative cases, and discrepancies in case distributions. This contextual information is essential for understanding the dynamics of the model evaluations.

The results reveal that classical machine learning models, as represented by LightAutoML, generally achieved superior performance compared to quantum methodologies across the evaluated datasets. The dataset *Carcinogens_Lagunin* exhibited the highest AUC-ROC score for QBoost (0.8614), providing compelling evidence of quantum annealing's capacity to yield performance enhancements in specific contexts. Although QSVM did not consistently exceed classical models, it occasionally surpassed QBoost in select datasets, thereby suggesting that both QBoost and QSVM exhibit considerable potential for advancing quantum machine learning techniques.

For example, in the dataset *BBB_Martins*, QSVM achieved an AUC-ROC score of 0.7939, which outperformed QBoost's score of 0.5826. This variability underscores the notion that while quantum approaches may not universally surpass classical methodologies, they can perform competitively under particular circumstances.

In conclusion, these findings provide valuable insights into the performance dynamics of quantum annealing techniques within the machine learning paradigm, particularly concerning their effectiveness on real-world ADMET datasets. The results emphasize the necessity for continued exploration into the potential applications of quantum annealing in machine learning, particularly as advancements in quantum computing technology continue to progress. The subsequent analysis delineates the performance of each method across datasets of varying sizes (see figures 10, 11 12, 13, and 14):

1) LIGHTAUTOML

- **Observation**: The LightAutoML method consistently achieved high AUC-ROC scores across datasets of varying sizes, although exhibiting the highest variance in performancefor compared to the other methods used. - **Key Insight**: The consistent AUC-ROC values indicate that LightAutoML effectively balances the trade-off between true positive and false positive rates, regardless of dataset size. - **Conclusion**: The robustness of this method signifies its reliability for general-purpose classification tasks, with a commendable ability to adapt to variations in training data size.

2) QBOOST

- **Observation**: QBoost exhibited commendable performance on smaller datasets, such as HIA_Hou, its AUC-ROC scores declined significantly for medium-sized, and large-sized datasets, exemplified by Pgp_Broccatelli and BBB_Martins. - **Key Insight**: The decrease in AUC-ROC scores for larger datasets suggests that QBoost may face challenges in scaling, necessitating further optimization or parameter tuning when applied to complex problems. - **Conclusion**: While QBoost shows potential for strong performance in smaller datasets, its generalizability to larger datasets remains limited without additional refinement.

3) QUANTUM SUPPORT VECTOR MACHINE (QSVM)

- **Observation**: QSVM recorded lower AUC-ROC scores compared to the other methodologies, but the performance was slightly more stable compared to the other methods. Notably, a trend emerged where QSVM's performance improved with larger datasets, such as BBB_Martins. - **Key Insight**: QSVM appears to benefit from the availability of larger datasets, as they provide more robust learning opportunities, consequently leading to enhanced classification performance. - **Conclusion**: QSVM seems to perform well regardless of the size of the datasets, showing competitive results in both small and large sized datasets. But it's consistency even in the large-sized datasets provide potential for scalability.

The findings substantiate the need for ongoing research into the practical applications of quantum annealing in machine learning, particularly as the field of quantum computing continues to evolve.

VIII. DISCUSSION AND FUTURE WORK

In this section, we discuss the key aspects of our research, beginning with the question of whether quantum computing is suitable for this particular task.

Quantum computing, especially through the use of quantum annealers, presents considerable potential for addressing optimization problems. Our results demonstrate the promise of leveraging quantum-classical hybrid approaches. In many cases, quantum annealers can outperform classical methods,

TABLE 1. Summary of AUC-ROC scores.

Dataset Name	AUC-ROC (LightAutoML)	AUC-ROC (QBoost)	AUC-ROC (QSVM)
PAMPA_NCATS	**0.8463**	0.7102	0.6316
HIA_Hou	**0.9962**	0.8323	0.8123
Pgp_Broccatelli	**0.9446**	0.7638	0.6102
Bioavailability_Ma	**0.7599**	0.6254	0.5713
BBB_Martins	**0.9310**	0.5826	0.7939
CYP2C9_Substrate_CarbonMangels	**0.6975**	0.6048	0.5347
CYP3A4_Substrate_CarbonMangels	**0.6442**	0.5554	0.5204
hERG	**0.8185**	0.6155	0.6382
DILI	**0.9113**	0.7103	0.6689
Carcinogens_Lagunin	0.8258	**0.8614**	0.7306

TABLE 2. Summary of dataset characteristics.

Dataset	Training Examples	Positive Cases	Negative Cases
PAMPA_NCATS	1424	1225	199
HIA_Hou	404	351	53
Pgp_Broccatelli	852	461	391
Bioavailability_Ma	448	351	97
BBB_Martins	1421	1096	325
CYP2C9_Substrate_CarbonMangels	468	97	371
CYP3A4_Substrate_CarbonMangels	469	249	220
hERG	458	310	148
DILI	332	161	171
Carcinogens_Lagunin	196	42	154

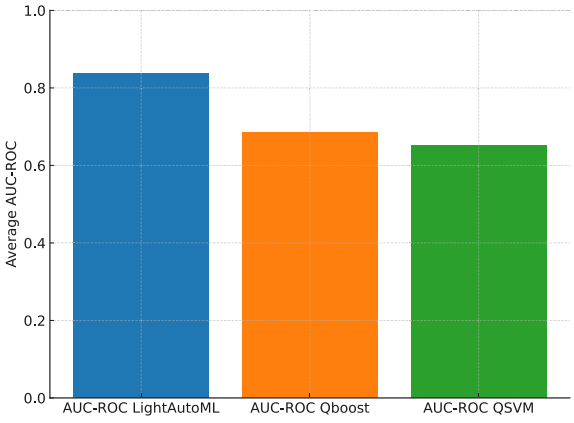


FIGURE 10. Mean AUC-ROC Score Comparison Across Methods.

particularly when dealing with complex combinatorial optimization problems that are computationally challenging for classical algorithms alone.

The theoretical framework behind our results is rooted in the principles of quantum annealing and optimization. Specifically, the performance of quantum annealers can be evaluated by examining the energy landscape of the optimization problem. For problems with complex energy surfaces, quantum annealing provides a nontrivial advantage over classical methods, especially when the problem's energy barrier E_{barrier} is large. The probability $P_{\min}$ of reaching the global minimum is given by:

$$P_{\min} \propto \exp\left(-\frac{E_{\text{barrier}}}{T_{\text{anneal}}}\right) \tag{39}$$

where T_{anneal} represents the annealing time. This implies that quantum annealers can more effectively escape local minima and explore broader solution spaces than classical algorithms in specific scenarios.

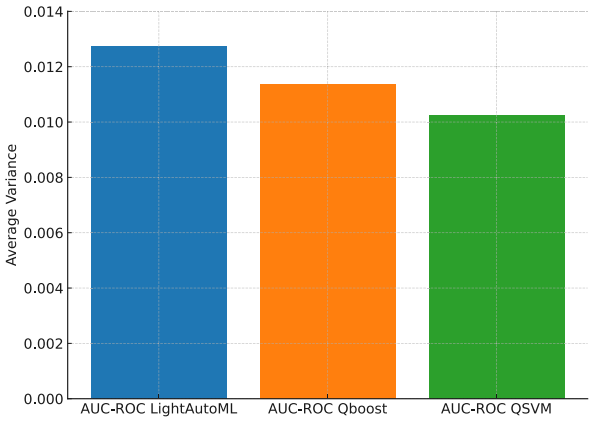


FIGURE 11. Variance in AUC-ROC Scores Comparison Across Methods.

However, it is important to recognize that current quantum hardware has limitations in terms of scalability, precision, and the necessity for problem-specific tuning. Our use of the Quantum Advantage Annealer 4.1 demonstrated notable advancements in solving optimization tasks. Despite these advancements, the performance gap between quantum and classical methods remains variable, depending largely on the size and structure of the problem. As the field progresses, improvements in hardware and refinements in algorithms will likely address these limitations.

One significant advantage we observed was the speed of quantum annealing for specific tasks. For example, during the

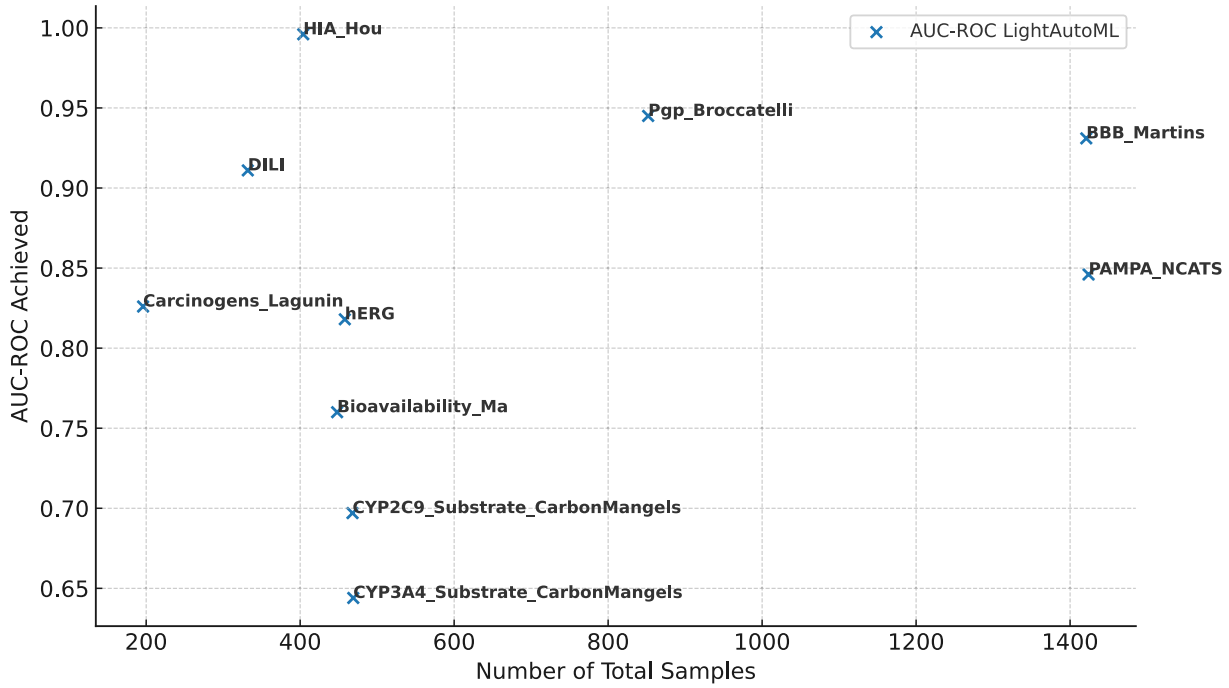


FIGURE 12. AUC-ROC Score vs Dataset Size (LightAutoMl).

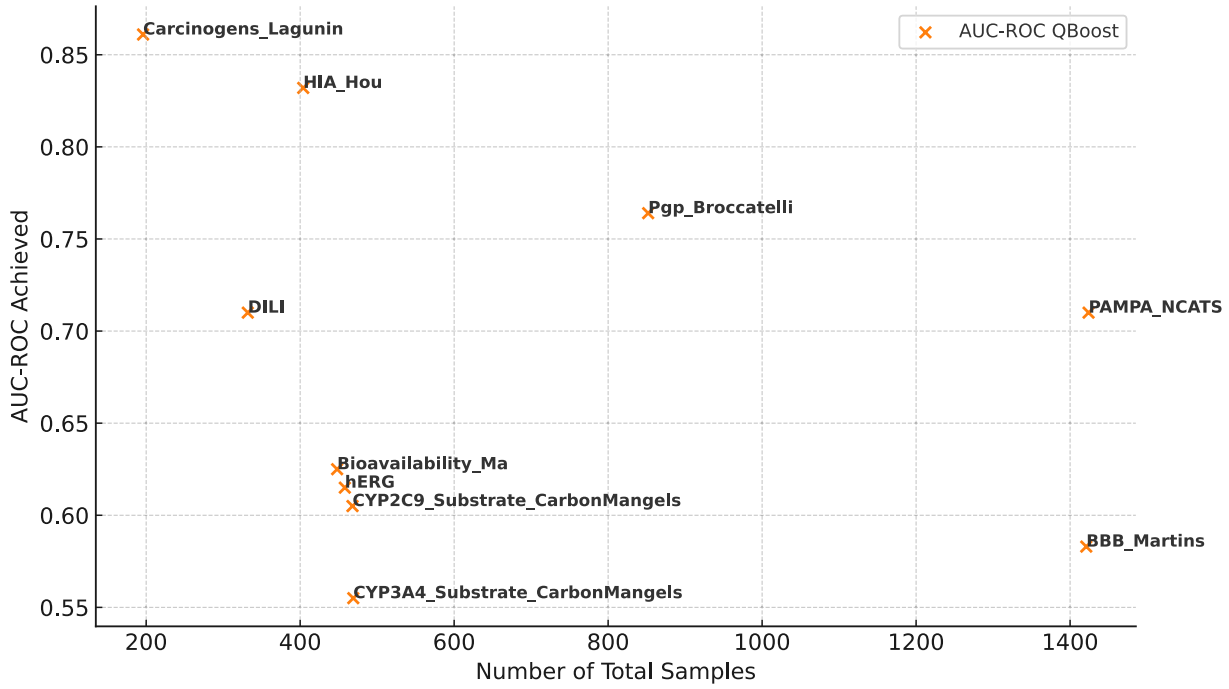


FIGURE 13. AUC-ROC Score vs Dataset Size (Qboost).

training phase, a single iteration using the quantum annealer took approximately 3 seconds, in contrast to the significantly longer training times required by classical methods. This highlights quantum annealing's potential in providing faster solutions for complex optimization problems, particularly where classical methods may struggle with computational efficiency.

Looking ahead, the upcoming Zypher quantum annealing hardware represents a significant step forward in quantum computing capabilities. Zypher is expected to greatly

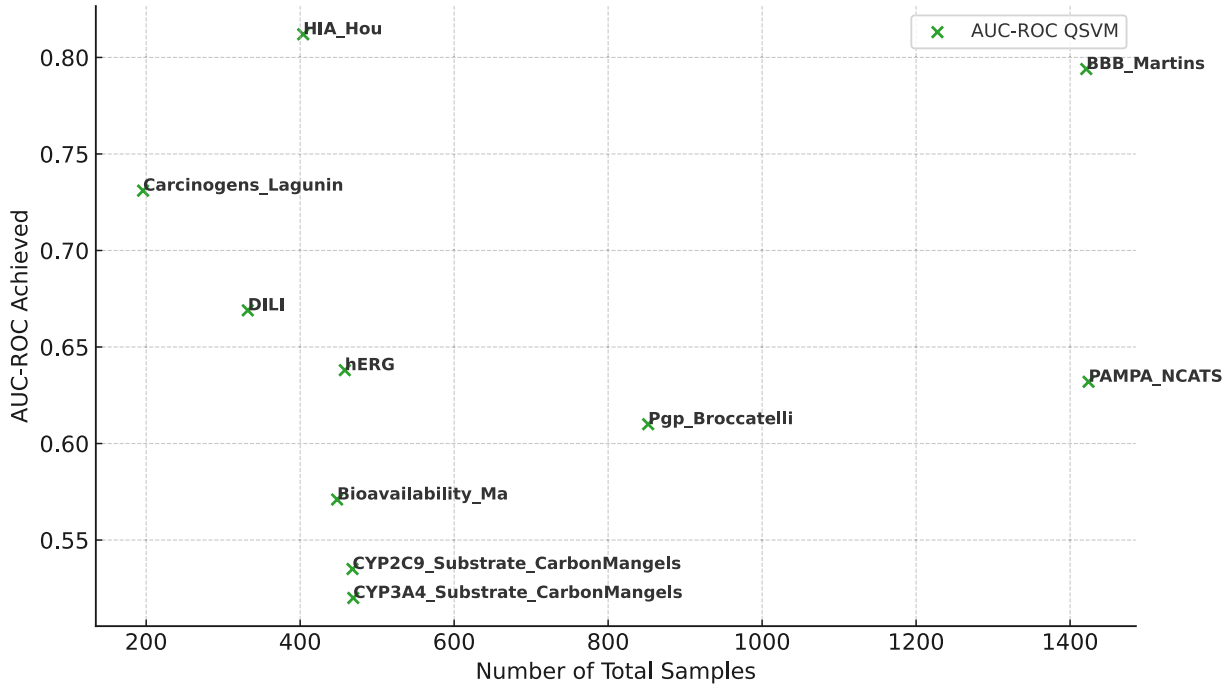


FIGURE 14. AUC-ROC Score vs Dataset Size (QSVM).

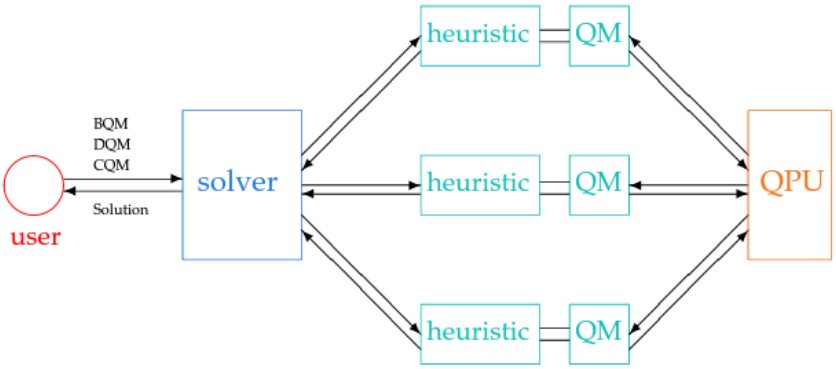


FIGURE 15. Structure of a hybrid solver in D-Wave's hybrid solver service. Adapted from [113].

expand problem-size capacity, allowing for the execution of optimization tasks on much larger datasets with increased precision and efficiency. Unlike its predecessors, Zypher will feature improved qubit coherence, faster annealing cycles, and more effective integration between quantum and classical resources. These enhancements are anticipated to unlock the full potential of quantum computing for real-world applications, particularly in fields requiring large-scale optimization, machine learning, and artificial intelligence.

By incorporating the Zypher hardware into future experiments, we expect to observe substantial improvements in scalability and solution quality. This will enable us to tackle more complex optimization problems, potentially surpassing the limitations of current classical methods. Additionally, with its fast annealing capabilities, Zypher holds the potential to significantly enhance quantum machine learning, positioning it as a crucial tool for advancing both research and industrial applications.

A. GENERAL OBSERVATIONS AND RECOMMENDATIONS

- **AUC-ROC Stability**: The benchmark method demonstrated the highest stability across datasets of varying sizes. The consistently high AUC-ROC scores suggest that this method provides reliable performance across a range of data complexities, making it a strong candidate for tasks that prioritize model robustness. - **QBoost Scalability**: The fluctuation in QBoost's performance indicates potential scalability issues, particularly as dataset size increases. For larger datasets, QBoost may require more sophisticated tuning to achieve competitive results. - **QSVM Sensitivity**: QSVM's sensitivity to dataset size highlights its dependency on large training sets to achieve optimal performance.

This suggests that QSVM is better suited for applications with abundant data and may underperform in data-scarce environments.

In conclusion, while quantum computing is still in its developmental stages, it demonstrates considerable promise, particularly with the emergence of new hardware such as Zypher. The future of quantum computing is likely to witness a reduction in the performance gap between quantum and classical methods. With ongoing advancements in hardware, quantum computing is poised to become a viable solution for solving complex optimization problems and improving machine learning techniques. Zypher's fast annealing capabilities, in particular, offer the potential to make significant contributions to the field of Quantum Machine Learning (QML), opening new opportunities for innovation and research.

IX. CONCLUSION

This study provides a detailed comparative analysis of quantum annealing machine learning methods, specifically Quantum Support Vector Machines (QSVM) and QBoost, applied to ADMET datasets. By leveraging D-Wave's Hybrid Quantum Processing Unit (HQPU) and its Fast Anneal feature, we assessed the performance of these quantum approaches against classical machine learning models, highlighting both their strengths and limitations. The results indicate that while QAML demonstrates potential in specific contexts, particularly with smaller datasets, challenges related to scalability, tuning, and dataset sensitivity remain evident. Quantum computing, particularly through quantum annealers, presents exciting possibilities for solving complex combinatorial optimization problems that are often intractable for classical algorithms. However, the current generation of quantum hardware, including the Quantum Advantage Annealer 4.1, still exhibits limitations in precision and scalability. These limitations suggest that the performance benefits of quantum computing are, for the moment, problem-dependent and subject to improvement as hardware continues to evolve.

Looking to the future, next-generation quantum hardware, such as the forthcoming Zypher, holds significant promise for overcoming these challenges. With enhanced qubit coherence, faster annealing cycles, and improved integration of quantum and classical resources, Zypher is expected to bring about meaningful advancements in the applicability of quantum computing, particularly in the domains of machine learning, optimization, and artificial intelligence. As these technologies mature, we anticipate a more widespread adoption of quantum-classical hybrid models in both research and industry, with quantum computing poised to play a pivotal role in addressing increasingly complex real-world problems.

The findings of this study emphasize the need for continued exploration into the optimization of quantum algorithms, as well as further refinement of quantum hardware to fully realize the potential of quantum computing in practical machine learning applications.

APPENDIX A
DETAILED DESCRIPTION OF ADMET DATASETS

In this study, we evaluate predictive models using a diverse set of datasets from the ADMET Benchmark Group. ADMET properties—Absorption, Distribution, Metabolism, Excretion, and Toxicity—are pivotal in assessing the pharmacokinetics and safety of drugs. Below, we describe each dataset in detail, incorporating advanced mathematical formulations where applicable [111], [112].

A. ABSORPTION DATASETS

The absorption datasets quantify the extent to which a drug is absorbed into the systemic circulation.

PAMPA_NCATS assesses the permeability of compounds through an artificial membrane, simulating intestinal absorption. The permeability coefficient (P_{PAMPA}) is derived from Fick's first law of diffusion, which is expressed as:

$$P_{\text{PAMPA}} = \frac{J \cdot A}{C \cdot t} \tag{40}$$

where J is the steady-state flux of the drug through the membrane, A is the membrane area, C is the concentration of the drug in the donor chamber, and t is the time for which the flux is measured. For a more detailed model considering membrane resistance (R_{mem}), the permeability can be refined as:

$$P_{\text{PAMPA}} = \frac{D_{\text{app}}}{R_{\text{mem}}} \tag{41}$$

where D_{app} is the apparent diffusion coefficient.

The **HIA_Hou** dataset measures human intestinal absorption (HIA) using the ratio of AUC values. The HIA can be described by:

$$\text{HIA} = \frac{\text{AUC}_{\text{oral}}}{\text{AUC}_{\text{iv}}} \times 100 \tag{42}$$

where AUC_{oral} is the area under the plasma concentration-time curve after oral administration and AUC_{iv} is the AUC following intravenous administration.

Pgp_Broccatelli evaluates the interaction of drugs with P-glycoprotein (Pgp), an efflux transporter. The effect of Pgp on drug absorption can be modeled using the following differential equation for the drug concentration $C(t)$ in the presence of Pgp:

$$\frac{dC(t)}{dt} = -k_{\text{efflux}} \cdot C(t) \tag{43}$$

where k_{efflux} is the rate constant for the Pgp-mediated efflux.

The **Bioavailability_Ma** dataset quantifies overall bioavailability (F) of a drug, which is the fraction of the dose that reaches systemic circulation. It can be computed using:

$$F = \frac{AUC_{\text{oral}} \cdot D_{\text{iv}}}{AUC_{\text{iv}} \cdot D_{\text{oral}}} \tag{44}$$

where D_{iv} and D_{oral} are the doses administered intravenously and orally, respectively.

B. DISTRIBUTION DATASETS

Distribution datasets provide insights into drug dispersion and its ability to cross biological barriers.

BBB_Martins measures the permeability of drugs across the blood-brain barrier (BBB). The permeability coefficient (P_{BBB}) can be modeled using the following equation:

$$P_{\text{BBB}} = \frac{D_{\text{BBB}}}{L_{\text{BBB}}} \tag{45}$$

where D_{BBB} is the diffusion coefficient across the BBB, and L_{BBB} is the thickness of the barrier. To account for drug-binding effects, the apparent permeability can be expressed as:

$$P_{\text{BBB}}^{\text{app}} = \frac{D_{\text{BBB}} \cdot (1 - f_{\text{bind}})}{L_{\text{BBB}}} \tag{46}$$

where f_{bind} represents the fraction of the drug bound to plasma proteins.

C. METABOLISM DATASETS

Metabolism datasets evaluate the biochemical modification of drugs by enzymes.

CYP2C19 Veith, **CYP2D6 Veith**, **CYP3A4 Veith**, **CYP1A2 Veith**, and **CYP2C9 Veith** measure enzyme inhibition. The inhibition of these cytochrome P450 enzymes can be described using the following competitive inhibition model:

$$v = \frac{V_{\max} \cdot [S]}{K_{\text{m}} \cdot \left(1 + \frac{[I]}{K_{\text{i}}}\right) + [S]} \tag{47}$$

where v is the reaction velocity, $V_{\max}$ is the maximum reaction rate, $[S]$ is the substrate concentration, K_{m} is the Michaelis-Menten constant, $[I]$ is the inhibitor concentration, and K_{i} is the inhibition constant.

CYP2C9 Substrate CarbonMangels, **CYP2D6 Substrate CarbonMangels**, and **CYP3A4 Substrate CarbonMangels** identify substrates for specific cytochrome P450 enzymes. The substrate affinity can be modeled using the Langmuir adsorption isotherm:

$$\theta = \frac{K_{\text{d}} \cdot [S]}{K_{\text{d}} + [S]} \tag{48}$$

where θ is the fraction of enzyme sites occupied by the substrate, and K_{d} is the dissociation constant.

D. TOXICITY DATASETS

Toxicity datasets assess the potential for harmful effects of drugs.

hERG evaluates the inhibition of the hERG potassium channel, which is linked to cardiac toxicity. The inhibition can be modeled using the following equation:

$$I_{\text{hERG}} = \frac{1}{1 + \left(\frac{[I]}{IC_{50}}\right)^n} \tag{49}$$

where I_{hERG} is the inhibition, $[I]$ is the inhibitor concentration, IC_{50} is the concentration required for 50% inhibition, and n is the Hill coefficient, which characterizes the steepness of the inhibition curve.

hERG_Karim comprises molecular structures labeled as hERG ($< 10\,\mu$M) and non-hERG ($\geq 10\,\mu$M) blockers, encoded as SMILES strings. The dataset integrates data from several sources: DeepHIT, which utilizes advanced neural network architectures to predict hERG inhibition and estimate IC_{50} values; BindingDB, which provides empirical binding affinity data; ChEMBL, which offers comprehensive bioactivity data; and additional literature, which contributes further chemical diversity.

AMES assesses mutagenicity using the Ames test. The mutagenicity score (M) is calculated as:

$$M = \frac{\text{Number of Positive Tests}}{\text{Total Number of Tests}} \tag{50}$$

indicating the proportion of tests showing mutagenic effects. A more detailed analysis involves the calculation of the mutagenicity index, which incorporates both positive and negative controls.

DILI measures drug-induced liver injury, quantifying hepatotoxicity through biomarkers and enzyme activity alterations.

Additional datasets such as **Skin Reaction**, **Carcinogens_Lagunin**, and **ClinTox** provide comprehensive assessments of various toxicity aspects, including allergic reactions, carcinogenic potential, and clinical toxicity profiles.

This detailed exploration of the ADMET datasets underscores their significance in evaluating drug properties and highlights the advanced mathematical models used to interpret the data, enhancing the precision and relevance of predictive modeling in drug discovery.

APPENDIX B
HYBRID SOLVER ARCHITECTURE

The hybrid solver, as depicted in Figure 15, integrates classical heuristic methods with D-Wave's quantum annealer to tackle complex problem-solving tasks efficiently under specific time constraints. The architecture consists of several interconnected components that coordinate to enhance computational performance [113].

At the core of the system is the front-end solver, which receives the problem input and optional time limit T from the user. This solver is responsible for launching multiple heuristic solvers that operate on classical CPUs and GPUs. These heuristic solvers employ traditional algorithms to search for good-quality solutions within the designated solution space. Each heuristic solver embeds a quantum module (QM), which formulates queries for the D-Wave Quantum Processing Unit (QPU).

The hybrid architecture is designed to leverage the unique strengths of both classical and quantum computing. The quantum modules interface with the QPU, where quantum annealing techniques are employed to explore complex solution landscapes. These quantum computations provide

insights and potential solutions that complement the heuristic search efforts.

As the heuristic solvers continue their search, the solutions are refined through iterative feedback from the QPU. The solver's portfolio management component ensures that the solutions are deduplicated, with only a subset of high-quality solutions being presented to the user before the time limit T is reached. The integration of classical and quantum methods, alongside continuous interaction between the heuristic solvers and the QPU, results in an efficient and robust approach to solving complex computational problems.

CODE AND DATA AVAILABILITY

The code and data for this project are available at the following links: GitHub Repository and Data.

ACKNOWLEDGMENT

(Hadi Salloum and Kamil Sabbagh contributed equally to this work.)

REFERENCES

[1] T. D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, and J. L. O'Brien, "Quantum computers," *Nature*, vol. 464, no. 7285, pp. 45–53, Mar. 2010.
[2] H. Salloum, M. Alawir, M. A. Alatasi, S. Asekrea, M. Mazzara, and M. Bahrami, "Quantum advancements in securing networking infrastructures," in *Proc. Int. Conf. Adv. Inf. Netw. Appl.* Cham, Switzerland: Springer, Jan. 2024, pp. 354–363.
[3] J. Preskill, "Quantum computing 40 years later," in *Feynman Lectures on Computation*. Boca Raton, FL, USA: CRC Press, 2023, pp. 193–244.
[4] R. P. Feynman, "There's plenty of room at the bottom," *Resonance*, vol. 16, no. 9, pp. 890–905, Sep. 2011.
[5] R. P. Feynman, "Simulating physics with computers," *Int. J. Theor. Phys.*, vol. 21, nos. 6–7, pp. 467–488, Jun. 1982.
[6] R. P. Feynman, "Quantum mechanical computers," *Opt. News*, vol. 11, no. 2, pp. 11–20, 1985.
[7] R. P. Poplavskiĭ, "Thermodynamic models of information processes," *Sov. Phys. Uspekhi*, vol. 18, no. 3, pp. 222–241, Mar. 1975.
[8] R. S. Ingarden, "Quantum information theory," *Rep. Math. Phys.*, vol. 10, no. 1, pp. 43–72, 1976.
[9] P. Benioff, "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines," *J. Stat. Phys.*, vol. 22, no. 5, pp. 563–591, May 1980.
[10] P. Benioff, "Quantum mechanical Hamiltonian models of Turing machines," *J. Stat. Phys.*, vol. 29, no. 3, pp. 515–546, Nov. 1982.
[11] D. Deutsch, "Quantum theory, the Church–Turing principle and the universal quantum computer," *Proc. Roy. Soc. London A, Math. Phys. Sci.*, vol. 400, no. 1818, pp. 97–117, 1985.
[12] K. Igeta and Y. Yamamoto, "Quantum mechanical computers with single atom and photon fields," in *Proc. Int. Quantum Electron. Conf.* Washington, DC, USA: Optica Publishing Group, Jul. 1988, pp. 1–3, Paper. TuI4.
[13] A. K. Ekert, "Quantum cryptography based on Bell's theorem," *Phys. Rev. Lett.*, vol. 67, no. 6, p. 661, 1991.
[14] B. Schumacher, "Quantum coding," *Phys. Rev. A, Gen. Phys.*, vol. 51, no. 4, pp. 2738–2747, Apr. 1995.
[15] I. L. Chuang, N. Gershenfeld, and M. Kubinec, "Experimental implementation of fast quantum searching," *Phys. Rev. Lett.*, vol. 80, no. 15, pp. 3408–3411, Apr. 1998.
[16] C. H. Bennett and G. Brassard, "A quantum information science and technology roadmap," *Part*, vol. 2, p. 12, Jul. 2004.
[17] Y. Wang, "Quantum computation and quantum information," *Stat. Sci.*, vol. 27, no. 3, pp. 373–394, Aug. 2012, doi: 10.1214/11-STS378.
[18] U. Alvarez-Rodriguez, M. Sanz, L. Lamata, and E. Solano, "Quantum artificial life in an IBM quantum computer," *Sci. Rep.*, vol. 8, no. 1, p. 14793, Oct. 2018.
[19] R. Courtland, "Google aims for quantum computing supremacy [news]," *IEEE Spectr.*, vol. 54, no. 6, pp. 9–10, Jun. 2017.
[20] J.-S. Chen, E. Nielsen, M. Ebert, V. Inlek, K. Wright, V. Chaplin, A. Maksymov, E. Páez, A. Poudel, P. Maunz, and J. Gamble, "Benchmarking a trapped-ion quantum computer with 30 qubits," 2023, *arXiv:2308.05071*.
[21] J. M. Pino, J. M. Dreiling, C. Figgatt, J. P. Gaebler, S. A. Moses, M. S. Allman, C. H. Baldwin, M. Foss-Feig, D. Hayes, K. Mayer, C. Ryan-Anderson, and B. Neyenhuis, "Demonstration of the trapped-ion quantum CCD computer architecture," *Nature*, vol. 592, no. 7853, pp. 209–213, Apr. 2021.
[22] E. Gibney, "Inside Microsoft's quest for a topological quantum computer," *Nature*, vol. 537, pp. 610–611, Oct. 2016.
[23] M. AbuGhanem, "Photonic quantum computers," 2024, *arXiv:2409.08229*.
[24] S. Curtis, "Quantum computing gets photonics spin: Susan curtis speaks to start-ups building quantum processors based on the photon," *Electro Opt.*, no. 320, pp. 14–17, 2022.
[25] H. Bluhm and L. R. Schreiber, "Semiconductor spin qubits—A scalable platform for quantum computing?" in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2019, pp. 1–5.
[26] M. De Michielis, E. Ferraro, E. Prati, L. Hutin, B. Bertrand, E. Charbon, D. J. Ibberson, and M. F. Gonzalez-Zalba, "Silicon spin qubits from laboratory to industry," *J. Phys. D, Appl. Phys.*, vol. 56, no. 36, Sep. 2023, Art. no. 363001.
[27] I. Kriekouki, "Industrial approach to quantum dots in fully-depleted silicon-on-insulator devices for quantum information applications," Ph.D. thesis, Dept. Micro Nanotechnologies, Microelectron., Université de Sherbrooke, Québec, QC, Canada, 2022.
[28] D. S. Weiss and M. Saffman, "Quantum computing with neutral atoms," *Phys. Today*, vol. 70, no. 7, pp. 44–50, 2017.
[29] K. Wintersperger, F. Dommert, T. Ehmer, A. Hoursanov, J. Klepsch, W. Mauerer, G. Reuber, T. Strohm, M. Yin, and S. Luber, "Neutral atom quantum computing hardware: Performance and end-user perspective," *EPJ Quantum Technol.*, vol. 10, no. 1, p. 32, Dec. 2023.
[30] P. Moor. (2023). *Atom Computing Announces Record-Breaking 1,225-Qubit Quantum Computer*. Forbes. [Online]. Available: https://www.forbes.com/sites/moorinsights/2023/10/24/atom-computing-announces-record-breaking-1225-qubit-quantum-computer/
[31] L. Childress and R. Hanson, "Diamond NV centers for quantum computing and quantum networks," *MRS Bull.*, vol. 38, no. 2, pp. 134–138, Feb. 2013.
[32] G. Liu and X. Pan, "Quantum information processing with nitrogen–vacancy centers in diamond," *Chin. Phys. B*, vol. 27, no. 2, Feb. 2018, Art. no. 020304.
[33] J. M. Rios, "Quantum manipulation of nitrogen-vacancy centers in diamond: From basic properties to applications," Ph.D. thesis, Harvard University, New York, NY, USA, 2010. [Online]. Available: http://www.fis.puc.cl/jmaze/MazeHUThesis.pdf
[34] G. Chroust, P. Doucek, and C. W. Loesch, *20 Years of IDIMT Conferences: Looking Back*. Singapore: SEA-Publications of the Institute for Systems Engineering and Automation, 2012.
[35] L. Jacak, P. Hawrylak, and A. Wojs, *Quantum Dots*. Berlin, Germany: Springer, 2013.
[36] J. Skiba-Szymanska, R. M. Stevenson, C. Varnava, M. Felle, J. Huwer, T. Müller, A. J. Bennett, J. P. Lee, I. Farrer, A. B. Krysa, P. Spencer, L. E. Goff, D. A. Ritchie, J. Heffernan, and A. J. Shields, "Universal growth scheme for quantum dots with low fine-structure splitting at various emission wavelengths," *Phys. Rev. Appl.*, vol. 8, no. 1, Jul. 2017, Art. no. 014013.
[37] S. Heng, D. Kim, T. Kim, and Y. Han, "How to solve combinatorial optimization problems using real quantum machines: A recent survey," *IEEE Access*, vol. 10, pp. 120106–120121, 2022.
[38] K. Ikeda, Y. Nakamura, and T. S. Humble, "Application of quantum annealing to nurse scheduling problem," *Sci. Rep.*, vol. 9, no. 1, p. 12837, Sep. 2019.
[39] D. de Falco and D. Tamascelli, "An introduction to quantum annealing," *RAIRO, Theor. Informat. Appl.*, vol. 45, no. 1, pp. 99–116, Jan. 2011.
[40] V. Kumar, G. Bass, C. Tomlin, and J. Dulny, "Quantum annealing for combinatorial clustering," *Quantum Inf. Process.*, vol. 17, no. 2, pp. 1–14, Feb. 2018.

[41] R. Ayanzadeh, M. Halem, and T. Finin, "Reinforcement quantum annealing: A hybrid quantum learning automata," *Sci. Rep.*, vol. 10, no. 1, p. 7952, May 2020.

[42] B. Apolloni, C. Carvalho, and D. de Falco, "Quantum stochastic optimization," *Stochastic Processes Appl.*, vol. 33, no. 2, pp. 233–244, Dec. 1989.

[43] P. Ray, B. K. Chakrabarti, and A. Chakrabarti, "Sherrington–Kirkpatrick model in a transverse field: Absence of replica symmetry breaking due to quantum fluctuations," *Phys. Rev. B, Condens. Matter*, vol. 39, no. 16, pp. 11828–11832, Jun. 1989.

[44] T. Kadowaki and H. Nishimori, "Quantum annealing in the transverse Ising model," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 58, no. 5, pp. 5355–5363, Nov. 1998.

[45] R. Honda, K. Endo, T. Kaji, Y. Suzuki, Y. Matsuda, S. Tanaka, and M. Muramatsu, "Development of optimization method for truss structure by quantum annealing," *Sci. Rep.*, vol. 14, no. 1, p. 13872, Jun. 2024.

[46] T. Lanting et al., "Entanglement in a quantum annealing processor," *Phys. Rev. X*, vol. 4, no. 2, May 2014, Art. no. 021041.

[47] C. Carugno, M. F. Dacrema, and P. Cremonesi, "Evaluating the job shop scheduling problem on a D-wave quantum annealer," *Sci. Rep.*, vol. 12, no. 1, p. 6539, Apr. 2022.

[48] M. Möller and C. Vuik, "On the impact of quantum computing technology on future developments in high-performance scientific computing," *Ethics Inf. Technol.*, vol. 19, no. 4, pp. 253–269, Dec. 2017.

[49] H. Riel, "Quantum computing technology," in *IEDM Tech. Dig.*, Dec. 2021, pp. 1.3.1–1.3.7.

[50] A. Acín, I. Bloch, H. Buhrman, T. Calarco, C. Eichler, J. Eisert, D. Esteve, N. Gisin, S. J. Glaser, F. Jelezko, S. Kuhr, M. Lewenstein, M. F. Riedel, P. O. Schmidt, R. Thew, A. Wallraff, I. Walmsley, and F. K. Wilhelm, "The quantum technologies roadmap: A European community view," *New J. Phys.*, vol. 20, no. 8, Aug. 2018, Art. no. 080201.

[51] V. K. Ralegankar, J. Bagul, B. Thakkar, R. Gupta, S. Tanwar, G. Sharma, and I. E. Davidson, "Quantum cryptography-as-a-service for secure UAV communication: Applications, challenges, and case study," *IEEE Access*, vol. 10, pp. 1475–1492, 2022.

[52] V. S. Barletta, D. Caivano, A. Lako, and A. Pal, "Quantum as a service architecture for security in a smart city," in *Proc. Int. Conf. Quality Inf. Commun. Technol.* Cham, Switzerland: Springer, Jan. 2023, pp. 76–89.

[53] H. Salloum, H. S. Aldaghstany, O. Orabi, A. Haidar, M. Bahrami, and M. Mazzara, "Integration of machine learning with quantum annealing," in *Proc. Int. Conf. Adv. Inf. Netw. Appl.* Cham, Switzerland: Springer, Jan. 2024, pp. 338–348.

[54] Y. Liu, S. Arunachalam, and K. Temme, "A rigorous and robust quantum speed-up in supervised machine learning," *Nature Phys.*, vol. 17, no. 9, pp. 1013–1017, Sep. 2021.

[55] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, Sep. 2017.

[56] J. Jäger and R. V. Krems, "Universal expressiveness of variational quantum classifiers and quantum kernels for support vector machines," *Nature Commun.*, vol. 14, no. 1, p. 576, Feb. 2023.

[57] A. D. King et al., "Quantum critical dynamics in a 5,000-qubit programmable spin glass," *Nature*, vol. 617, no. 7959, pp. 61–66, May 2023.

[58] S. Pakin and S. P. Reinhardt, "A survey of programming tools for D-wave quantum-annealing processors," in *Proc. 33rd Int. Conf. High Perform. Comput.*, Frankfurt, Germany. Cham, Switzerland: Springer, 2018, pp. 103–122.

[59] M. W. Johnson et al., "Quantum annealing with manufactured spins," *Nature*, vol. 473, no. 7346, pp. 194–198, May 2011.

[60] D. S. Holmes, A. M. Kadin, and M. W. Johnson, "Superconducting computing in large-scale hybrid systems," *Computer*, vol. 48, no. 12, pp. 34–42, Dec. 2015.

[61] J. Brooke, D. Bitko, T. F. Rosenbaum, and G. Aeppli, "Quantum annealing of a disordered magnet," *Science*, vol. 284, no. 5415, pp. 779–781, Apr. 1999.

[62] W. Wu, B. Ellman, T. Rosenbaum, G. Aeppli, and D. Reich, "From classical to quantum glass," *Phys. Rev. Lett.*, vol. 67, no. 15, pp. 2076–2079, Oct. 1991.

[63] D. M. Silevitch, C. Ancona-Torres, G. Aeppli, and T. F. Rosenbaum, "Quantum and classical glass transitions in $LiHo_xY_{1-x}F_4$," *Phys. Rev. Lett.*, vol. 101, no. 5, Mar. 2008, Art. no. 057201.

[64] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, "Quantum computation by adiabatic evolution," 2000, *arXiv:quant-ph/0001106*.

[65] W. M. Kaminsky, S. Lloyd, and T. P. Orlando, "Scalable superconducting architecture for adiabatic quantum computation," 2004, *arXiv:quant-ph/0403090*.

[66] E. Cohen and B. Tamir, "D-wave and predecessors: From simulated to quantum annealing," *Int. J. Quantum Inf.*, vol. 12, no. 3, Apr. 2014, Art. no. 1430002.

[67] S. Belenzon and L. C. Cioaca, "Guaranteed markets and corporate scientific research," Nat. Bur. Econ. Res., Cambridge, MA, USA, Tech. Rep., 2021.

[68] A. Ghosh and S. Mukherjee, "Quantum annealing and computation: A brief documentary note," 2013, *arXiv:1310.1339*.

[69] R. F. Lucas, G. Wagenbreth, J. J. Tran, D. R. Pratt, and D. M. Davis, "Practical adiabatic quantum computing: Implications for the simulation community," in *Proc. Interservice/Ind. Simulation, Training Educ. Conf.*, Orlando, FL, USA, 2013, pp. 1–9 and 12, Paper 13020.

[70] D. Lidar, "Theoretical and experimental error correction of programmable quantum annealing," in *New Directions in the Quantum Control Landscape*, 2013, p. 54.

[71] Y. Koshka, D. Perera, S. Hall, and M. A. Novotny, "Determination of the lowest-energy states for the model distribution of trained restricted Boltzmann machines using a 1000 qubit D-wave 2X quantum computer," *Neural Comput.*, vol. 29, no. 7, pp. 1815–1837, Jul. 2017.

[72] Y. Cao, S. Jiang, D. Perouli, and S. Kais, "Solving set cover with pairs problem using quantum annealing," *Sci. Rep.*, vol. 6, no. 1, p. 33957, Sep. 2016.

[73] D-Wave Syst. *Introduction To D-Wave Quantum Hardware*. Accessed: Sep. 29, 2024. [Online]. Available: https://docs.dwavesys.com/docs/latest/c_gs_4.html

[74] M. Jüenger, E. Lobe, P. Mutzel, G. Reinelt, F. Rendl, G. Rinaldi, and T. Stollenwerk, "Performance of a quantum annealer for ising ground state computations on chimera graphs," 2019, *arXiv:1904.11965*.

[75] D. Willsch, M. Willsch, C. D. G. Calaza, F. Jin, H. De Raedt, M. Svensson, and K. Michielsen, "Benchmarking advantage and D-wave 2000Q quantum annealers with exact cover problems," *Quantum Inf. Process.*, vol. 21, no. 4, p. 141, Apr. 2022.

[76] R. Ayanzadeh, S. Mousavi, M. Halem, and T. Finin, "Quantum annealing based binary compressive sensing with matrix uncertainty," 2019, *arXiv:1901.00088*.

[77] Y. Liu, "QUBO formulation for arithmetic progression graph labeling problem," Ph.D. thesis, ResearchSpace@ Auckland, Auckland, New Zealand, 2024.

[78] D-Wave Syst. (2024). *Hybrid Solvers Overview*. [Online]. Available: https://docs.dwavesys.com/docs/latest/c_gs_4.html

[79] S. Zbinden, A. Bärtschi, H. Djidjev, and S. Eidenbenz, "Embedding algorithms for quantum annealers with chimera and pegasus connection topologies," in *Proc. Int. Conf. High Perform. Comput.* Cham, Switzerland: Springer, Jan. 2020, pp. 187–206.

[80] O. Melchert, H. G. Katzgraber, and M. A. Novotny, "Site- and bond-percolation thresholds in $K_{n,n}$-based lattices: Vulnerability of quantum annealers to random qubit and coupler failures on chimera topologies," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 93, no. 4, Apr. 2016, Art. no. 042128.

[81] A. D. King et al., "Coherent quantum annealing in a programmable 2,000 qubit Ising chain," *Nature Phys.*, vol. 18, no. 11, pp. 1324–1328, Nov. 2022.

[82] N. Aksamit, A. Tchagang, Y. Li, and B. Ombuki-Berman, "Hybrid fragment-SMILES tokenization for ADMET prediction in drug discovery," *BMC Bioinf.*, vol. 25, no. 1, p. 255, Aug. 2024.

[83] D. Campos and H. Ji, "IMG2SMI: Translating molecular structure images to simplified molecular-input line-entry system," 2021, *arXiv:2109.04202*.

[84] E. J. Bjerrum, "SMILES enumeration as data augmentation for neural network modeling of molecules," 2017, *arXiv:1703.07076*.

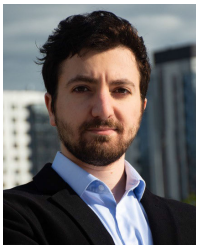
[85] J. Sieg, C. W. Feldmann, J. Hemmerich, C. Stork, F. Sandfort, P. Eiden, and M. Mathea, "MolPipeline: A Python package for processing molecules with RDKit in scikit-learn," *J. Chem. Inf. Model.*, vol. 64, no. 24, pp. 9027–9033, Dec. 2024.

[86] DataMol. (2024). *DataMol: A Comprehensive Platform for Molecular Data Analysis*. Accessed: Sep. 30, 2024. [Online]. Available: https://datamol.io/

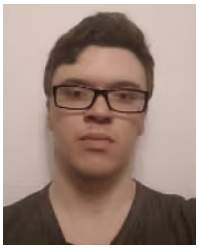
[87] RDKit. (2024). *RDKit: Open-Source Cheminformatics Software*. Accessed: Sep. 30, 2024. [Online]. Available: https://www.rdkit.org/

[88] P. Tossou, C. Wognum, M. Craig, H. Mary, and E. Noutahi, "Real-world molecular out-of-distribution: Specification and investigation," *J. Chem. Inf. Model.*, vol. 64, no. 3, pp. 697–711, Feb. 2024.
[89] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande, and J. Leskovec, "Strategies for pre-training graph neural networks," 2019, *arXiv:1905.12265*.
[90] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Jan. 2017, pp. 1–6.
[91] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," 2018, *arXiv:1809.10341*.
[92] C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T. Liu, "Do transformers really perform badly for graph representation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, Dec. 2021, pp. 28877–28888.
[93] B. Shin, S. S. Park, K. Kang, and J. C. Ho, "Self-attention based molecule representation for predicting drug-target interaction," in *Proc. Mach. Learn. Healthcare Conf.*, Jan. 2019, pp. 230–248.
[94] A. Vakhrushev, A. Ryzhkov, M. Savchenko, D. Simakov, R. Damdinov, and A. Tuzhilin, "LightAutoML: AutoML solution for a large financial services ecosystem," 2021, *arXiv:2109.01528*.
[95] M. V. C. Aragão et al., "A practical evaluation of AutoML tools for binary, multiclass, and multilabel classification," *Res. Square*, 2024, doi: 10.21203/rs.3.rs-4172933/v1.
[96] A. Rodrigues, T. Almeida, L. B. Silva, and C. Costa, "Diving into AutoML in medical imaging: Solution for non-ML practitioners," *IEEE Access*, vol. 12, pp. 151275–151302, 2024.
[97] LightAutoML Team. (2024). *LightAutoML Documentation*. Accessed: Sep. 30, 2024. [Online]. Available: https://lightautoml.readthedocs.io/en/v.0.4.0/
[98] P. Gijsbers, E. LeDell, J. Thomas, S. Poirier, B. Bischl, and J. Vanschoren, "An open source AutoML benchmark," 2019, *arXiv:1907.00909*.
[99] M. Awad and R. Khanna, "Support vector machines for classification," in *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*. Berkeley, CA, USA: Apress, 2015, pp. 39–66.
[100] P.-W. Wang and C.-J. Lin, "Support vector machines," in *Data Classification: Algorithms and Applications*, 1st ed. Chapman & Hall, Jul. 2014.
[101] N. Deng, Y. Tian, and C. Zhang, *Support Vector Machines: Optimization Based Theory, Algorithms, and Extensions*. Boca Raton, FL, USA: CRC Press, 2012.
[102] D. Willsch, M. Willsch, H. De Raedt, and K. Michielsen, "Support vector machines on the D-wave quantum annealer," *Comput. Phys. Commun.*, vol. 248, Mar. 2020, Art. no. 107006.
[103] H. Bhatia and F. Phillipson, "Performance analysis of support vector machine implementations on the D-wave quantum annealer," in *Proc. Int. Conf. Comput. Sci.* Cham, Switzerland: Springer, Jan. 2021, pp. 84–97.
[104] F. Orazi, S. Gasperini, S. Lodi, and C. Sartori, "Hybrid quantum technologies for quantum support vector machines," *Information*, vol. 15, no. 2, p. 72, Jan. 2024.
[105] D. Anguita, S. Ridella, F. Rivieccio, and R. Zunino, "Quantum optimization for training support vector machines," *Neural Netw.*, vol. 16, nos. 5–6, pp. 763–770, Jun. 2003.
[106] H. Neven, V. S. Denchev, G. Rose, and W. G. Macready, "QBoost: Large scale classifier training withadiabatic quantum optimization," in *Proc. Asian Conf. Mach. Learn.*, 2012, pp. 333–348.
[107] S. K. Jha and N. B. Adhikari, "Analysis of ensembled learning using classical and quantum boosting approaches for diabetes mellitus prediction," *Res. Square*, Apr. 2024, doi: 10.21203/rs.3.rs-4199185/v1.
[108] J. Dulny III and M. Kim, "Developing quantum annealer driven data discovery," 2016, *arXiv:1603.07980*.
[109] K. Karimi, N. G. Dickson, F. Hamze, M. H. S. Amin, M. Drew-Brook, F. A. Chudak, P. I. Bunyk, W. G. Macready, and G. Rose, "Investigating the performance of an adiabatic quantum optimization processor," *Quantum Inf. Process.*, vol. 11, no. 1, pp. 77–88, Feb. 2012.
[110] K. Jun, "QUBO formulations for a system of linear equations," *Results Control Optim.*, vol. 14, Mar. 2024, Art. no. 100380.
[111] Therapeutic Data Commons. (2023). *Admet Benchmark Group—Overview*. Accessed: Oct. 1, 2024. [Online]. Available: https://tdcommons.ai/benchmark/admet_group/overview/
[112] L. Guan, H. Yang, Y. Cai, L. Sun, P. Di, W. Li, G. Liu, and Y. Tang, "ADMET-score—A comprehensive scoring function for evaluation of chemical drug-likeness," *MedChemComm*, vol. 10, no. 1, pp. 148–157, 2019.
[113] D-Wave Syst. (Sep. 2020). *Hybrid Solvers for Quadratic Optimization*. [Online]. Available: https://www.dwavesys.com/media/soxph512/hybrid-solvers-for-quadratic-optimization.pdf

HADI SALLOUM is currently pursuing the bachelor's degree in artificial intelligence and data science. He focuses on applying advanced AI techniques to complex problems across various domains. He is a Researcher with the AI Institute, Innopolis University, Republic of Tatarstan, Russia. He has demonstrated exceptional commitment to physics, competing in the International Physics Olympiad 2021 in Lithuania and the European Physics Olympiad in Estonia. His research interests include innovative AI and data science applications, with expertise in quantum annealing and quantum computing to enhance computational capabilities.

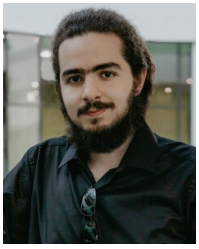


KAMIL SABBAGH received the bachelor's degree in computer science from Innopolis University, Republic of Tatarstan, Russia, in 2023. He is currently pursuing the master's degree in artificial intelligence and data science, with a focus on fairness in machine learning and image synthesis. As a Researcher with the AI Institute, he concentrates on addressing ethical issues in AI and developing advanced techniques in image synthesis.



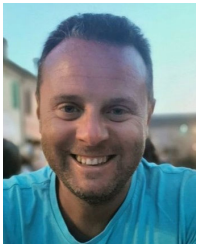
VLADISLAV SAVCHUK is currently an AI Researcher with the Laboratory of Artificial Intelligence in New Materials, Research Center for AI, Innopolis University. His work focuses on applying AI techniques to material discovery and optimization, advancing innovations in the field.

RUSLAN LUKIN is currently an AI Researcher and a Laboratory Head of Innopolis University, specializing in artificial intelligence for new materials and drug discovery. He is also the Founder of the AI in drug discovery startup, Inventum.ai. He has extensive experience in data science, machine learning, and computational chemistry, with a particular focus on deep learning and geometric learning. He has, developed multiple approaches for seeking innovative catalysts using deep learning and led various commercial AI projects in collaboration with leading companies. He is a frequent speaker at international conferences and has published multiple articles in prestigious journals, including Nature Portfolio.



OSAMA ORABI is currently pursuing the bachelor's degree in artificial intelligence and data science with Innopolis University, Republic of Tatarstan, Russia.

MARAT ISANGULOV is currently a leading Programmer-Mathematician with the Laboratory of Artificial Intelligence in New Materials, Research Center in the Sphere of Artificial Intelligence, Innopolis University.



MANUEL MAZZARA received the Ph.D. degree in computing science from the University of Bologna, Italy. He is currently a Professor of computer science with Innopolis University, Russia, with a research background in software engineering, service-oriented architecture, concurrency theory, formal methods, and software verification. He is also the Dean of the Faculty of Computer Science and Engineering and the Head of the International Cooperation Office, Innopolis University. He has published many relevant and highly cited articles, in particular in the field of service engineering and software architectures. He has collaborated with European and U.S. industries and governmental and inter-governmental organizations, such as the United Nations, always at the edge between science and software production. The work conducted by him and his team in recent years focuses on the development of theories, methods, tools, and programs covering the two major aspects of software engineering, such as the process side, related to how we develop software; and the product side.