Published Online First: July, 2024
https://doi.org/10.21123/bsj.2024.9278
P-ISSN: 2078-8665 - E-ISSN: 2411-7986

Baghdad Science Journal

Integrated System of Swarm Intelligence and Neural Network for Molecular Similarity Detection

Fadia Sami[1], **Hakan KOYUNCU**[2]*

[1] Department of Computer Engineering, Information Technologies, Altınbaş University, Istanbul, Turkey.
[2]Department of Computer Engineering, Faculty of Engineering and Architecture, Altınbaş University, Istanbul, Turkey.
*Corresponding Author.

Received 15/08/2023, Revised 25/12/2023, Accepted 27/12/2023, Published Online First 20/07/2024

Abstract

Molecular similarity, governed by the principle that "similar molecules exhibit similar properties," is a pervasive concept in chemistry with profound implications, notably in pharmaceutical research where it informs structure-activity relationships. This study focuses on the pivotal role of molecular similarity techniques in identifying sample molecules akin to a target molecule while differing in key features. Within the realm of artificial intelligence, this paper introduces a novel hybrid system merging Swarm Intelligence (SI) behaviors (Aquila and Termites) with Neural Networks. Unlike previous applications where Aquila or Termites were used individually, this amalgamation represents a pioneering approach. The objective is to determine the most similar sample molecule in a dataset to a specific target molecule. Accuracy assessments reveal a manual evaluation accuracy of 70.58%, surging to 90% with the incorporation of Neural Networks. Additionally, a three-dimensional grid elucidates the Quantitative Structure-Activity Relationship (QSAR). The Euclidean and Manhattan Distance metrics quantify differences between molecules. This study contributes to molecular similarity assessment by presenting a hybrid approach that enhances accuracy in identifying similar molecules within complex datasets.

Keywords: Molecular Similarity, Swarm intelligence (SI), Aquila, Termite, Neural Network.

Introduction

Introduction (Background and Problem Statement):

In the intricate landscape of artificial intelligence and optimization, addressing complex problems characterized by overwhelming complexity, uncertainty, and inherent stochasticity is a formidable challenge. Swarm intelligence (SI) algorithms, drawing inspiration from biological behavior and social interactions, have emerged as promising solutions. However, despite their recognized potential, a clear understanding of the practical applications of SI algorithms in real-world scenarios, particularly those involving intricate decision-making processes, remains elusive[1].

As scientists have delved into the study of biological behavioral intelligence since the Cape bee study in 1883[1], the need for effective approaches to decentralized decision-making, akin to how bees choose nesting locations, has become increasingly apparent. Swarm intelligence algorithms, encompassing artificial neural networks, fuzzy systems, and evolutionary computing, offer a unique avenue for addressing these challenges[1].

Research Objectives:

Our research aims to harness the potential of swarm intelligence algorithms in conjunction with meta-heuristic optimization techniques inspired by natural processes. These techniques, such as evolutionary algorithms, Simulated Annealing (SA), and Swarm Intelligence (SI) algorithms, demonstrate efficacy in solving challenging optimization problems[2,3].

Natural sciences rely on models for interpreting findings and making predictions. Theoretical chemistry focuses on elucidating chemical processes, refining methodologies for reactive process models. The dimensionality problem in molecular quantum dynamical calculations introduces limitations to precision, prompting the use of approximations for quantum dynamics. Molecular docking techniques predict chemical affinities and modalities for scientific and medicinal purposes. Flexible ligand docking strikes a balance between accuracy and efficiency, simulating the process in docking programs. The effectiveness of these simulations depends on the chosen search method and scoring algorithm, with the scoring function guiding the search method and assessing the quality of the docking conformation[4,5].

Contribution of the Study:

This study contributes to the growing interest in meta-heuristic optimization techniques by showcasing their applicability and success in simulating biological or physical processes. As these algorithms excel in avoiding local optima, they offer valuable solutions to real-world problems where traditional methods may fall short[1-3].

Outline of the Paper:

This paper is structured to delve into the application of swarm intelligence algorithms in addressing challenges encountered by analytical toxicologists, particularly in the forensic drug analysis of new illegal compounds. The subsequent sections explore molecular similarity analysis, cheminformatics, theoretical chemistry, molecular docking techniques, and the evaluation of pharmaceutical similarity[6,7].

Similarity in Various Fields:

The idea of similarity and its relation to human understanding are rooted in unconsciously formed associations based on prior experience. Human minds generate new ideas by comparing new information with stored knowledge, reflecting the ancient Greek philosophy that underlies modern science's understanding of similarity. Logical inference relies on analogous reasoning, which involves careful comparisons of structures. Similarity proves essential in various fields, including chemistry and mathematics, where analogous figures and systems play a crucial role. Quantifying similarity using clear parameters is advantageous for scientific applications[8].

Materials and Methods

Materials:

Swarm Intelligence (SI)

The term "SI" was first applied to the "intelligent" actions of cellular robotics systems. Subsequently, the idea developed into a well-established research field with methodologies, strategies, and AI problem-solving algorithms. The actions of fish, insects, and birds, as well as their capacity to function as a group of agents, are the main sources of inspiration for SI techniques. These social actors do really have a relatively low level of individual intelligence. Yet, when they socially interact with one another or their environment, they show signs of being able to complete difficult tasks without the aid of a centralized authority, like as the colony's queen. Anomaly detection has effectively used SI algorithms due to their development, simplicity, resilience, and adaptability [9]. There are various SI algorithms; here, concentrating on two (Aquila and Termite).

Aquila Optimizer (AO)

The Aquila, a raptor that is well-known throughout the Northern Hemisphere, was suggested by Abualigah in 2021 [3]. The Aquila species is the one that is most widely distributed. Aquila belongs to the family Accipitridae, like all birds. The rear of the neck of an Aquila typically has lighter Golden-brown plumage. This species of Aquila youngsters frequently have a white tail and very faint white markings on their wings. Aquila hunts a variety of prey, primarily ground mammals like squirrels, marmots, hares, and rabbits, using her speed, agility, strong feet, and enormous, sharpened talons. Aquila can be seen in nature, along with their distinctive characteristics. Aquila may maintain 200 km2 or larger holdings. They construct substantial nests above mountains and other elevated terrain. They reproduce in the spring and are likely to live together for the remainder of their lives because they are monogamous. The eggs, which the female can lay up to 4, are subsequently incubated for 6 to 120 weeks. In around 12 weeks, one or two newborns usually reach adulthood. These young aquila frequently reach their peak of confidence in the fall, at which point they disperse far in order to claim territory. Aquila is one of the most researched birds in the world because of its courageous hunting behavior. Male Aquila caught considerably more prey when hunting by alone. Aquila hunt squirrels, rabbits, and many other creatures with their speed and razor-sharp claws.

They have also been reported to target mature deer. The ground squirrel ranks as the second most 125 noteworthy animal in Aquila's diet. The Aquila are known to hunt primarily using four distinct methods, each of which has its own special advantages. Most Aquila are adept at switching between many hunting methods quickly and expertly depending on the situation [2]. It has a variety of hunting methods, but we'll focus on the expanded exploration strategy, which involves a vertical stoop and a high soar. This strategy involves the Aquila flying well above the ground to thoroughly scan the area before making a vertical fall after it has spotted the target [2,3,10].

Termite Colony

Termites are secretive, gregarious insects that lack vision. Depending on the species, they can dwell in colonies with up to a million termites. The social insects exhibit complex behavior, such as group decision-making and nest construction. Many optimization issues are currently being addressed using swarm intelligence, a method of rational collective decision-making. Termites, along with ants, bees, and other social insects, have one of the most complex forms of swarm cognition. In addition to interacting with one another, they also interact with their surroundings. The termite colonies display a division of labor and are made up of groups with distinct roles within the community. The laborers, the military, and the reproductive are the three main categories. Due to their blindness, termites use chemical signals (pheromones) and vibrations to communicate. The social structure of termites is aided by pheromones. They use scent to identify each other in the nest. Each colony creates its own distinct smell. Additionally, termites release pheromones to direct the colony to food or warn it of danger. The pheromones that the foraging termites emit while digging underground are released from glands on their abdomen. When a food source is found, the intensity of this sort of pheromone increases. As a result, the food supplier can hire more staff. Similar to this, whenever the colony is attacked or invaded, the soldiers release a different sort of pheromone, which aroma alerts the other termites to oncoming danger.

The behavior of the termite colony is also influenced by environmental conditions. Temperature and moisture are the two main components. Depending on the climates that termites like, foraging activity depends on soil temperature and is very low during extremely cold or hot weather. The ideal temperature ranges from 21^{o} C to 36^{o} C depending on the species. Termites require moisture, either from their surroundings or their food supply. Termites use their technical talents to build tubes and tunnels to preserve the proper amount of moisture within and maintain their soft cuticle. Termites in a colony make stochastic decisions that determine their movement patterns. Based on the strength of the pheromones in the area, termites choose their movement pattern. As a result, each termite can carry out complicated activities by following a few basic behavioral principles. The termite swarm reacts to the new circumstances in an adaptable manner. A wide variety of dynamic optimization issues can be solved using this technique. This termite intelligence has proved crucial in the development of a unique optimization technique for WAN traffic control [11].

Termites, globally distributed social insects with a sophisticated structure, exhibit a life cycle comprising the reproductive, soldier, and worker castes, each assigned specific colony maintenance tasks. The king and queen solely reproduce, contributing to millions of eggs annually, while termite workers tend to incubate and care for the eggs. Worker castes, constituting 70-80% of the colony, handle labor responsibilities, including food-related tasks. Soldiers, comprising 20-30%, guard the colony and engage in defense. Reproductive castes, represented by a single viable pair, the queen and king, ensure colony reproduction through annual nuptial flights. Termites communicate chemically and mechanically to coordinate activities, impacting colony growth and survival. Despite being mostly blind, termites rely on pheromones for navigation and sharing food source information on walkways[12].

Neural Network

Neural networks, inspired by the complex network of neurons in the human brain, have become powerful tools for enhancing learning systems. These networks consist of artificial information-processing units, with nodes, weights, and layers playing vital roles [13]. The topology and internal parameters of neural networks, such as the number of inputs, outputs, hidden layers, neurons, weights, biases, and activation functions, greatly impact their functionality and performance. Finding an optimal topology and internal parameters is crucial to ensure the quality of the final neural network model. There are various types of neural networks based on their architecture, connectivity, and specialized use cases. Some commonly used types include feedforward neural networks (FNN), convolutional neural networks (CNN), recurrent neural networks (RNN), long short-term memory (LSTM), generative adversarial networks (GAN), autoencoders, and reinforcement learning (RL) networks [14].

Feedforward neural network is a fundamental design where information flows from input nodes to output nodes in a forward direction without cycles. This type of network is often trained using strategies like adjusting weights based on the reverse of the output result [15]. FNNs consist of interconnected neurons organized in layers, and they can approximate any continuous function, making them versatile for solving various problems such as pattern recognition, clustering, classification, and more [13]. Activation functions play a crucial role in neural networks by determining the activation state of a neuron based on the input it receives [16]. Common activation functions include step functions, sigmoid functions (e.g., logistic function and hyperbolic tangent function), and the softmax function. These functions enable non-linear transformations and ensure smoothness, differentiability, and computational efficiency in the network [17].

To evaluate the performance of neural networks, loss functions are employed to measure the error between the network's predictions and the expected output. Mean square error (MSE) and categorical cross-entropy are commonly used loss functions. The backpropagation algorithm, a first-order gradient-descent technique, is widely used for optimizing neural networks. It propagates the output layer's error backward through the hidden layers, adjusting the weights based on the contribution to the prediction outcome [18, 15]. During training, the neural network iteratively updates its weights and minimizes the error by comparing its generated values with the ground truth.

Training techniques can involve stochastic/online training, where redundancy in the training pattern is addressed and dynamic learning is possible, or batch mode training, which ensures a local minimum and can be faster for larger datasets. Optimization techniques for neural networks involve architectural optimization, where factors like the number of nodes, layers, and activation functions are optimized. In the past, only the weights of connections were optimized, but modern approaches include metaheuristic-based optimization techniques that consider multiple objectives, such as generalization, model simplification, and minimizing an approximation error [13]. Hyperparameter tuning is a crucial step in deep neural networks, particularly for classification tasks. It involves finding the optimal hyperparameters that govern the neural network architecture and the training process. Hyperparameter optimization (HPO) is a key component of Automated Machine Learning (AutoML) and helps achieve better performance and accuracy in neural network models [19].

Molecules

One of the most important concepts in biology, chemistry, medicine, and pharmacy is the concept of a molecule, which is an electrically neutral group of two or more atoms bonded together through chemical events. A molecule typically has a certain structure, and this structure—particularly for organic molecules—can be highly complex. Molecule characteristics are typically greatly influenced by the structure of the molecule. The architectures of molecules have a significant impact on their attributes and capabilities [20].

Small-molecules

Small molecules have continuously helped to advance medicine and address unmet medical needs, saving countless lives in the process. Small molecules have also proved essential in biomedical research as chemical probes, helping to understand the biology of disease. Over the past century, traditional small-molecule medicines have dominated drug research. The drug discovery toolbox has, however, expanded to include more recent techniques, such as RNA-targeting small molecules (RSMs) and proteolysis-targeting chimeras (PROTACs), together with biological techniques, like antibody-based therapy and cell- and gene-therapy. Most big pharmaceutical firms now provide more modality-neutral funding for medication research [21].

Drugs

Drugs are substances, including proteins and chemical compounds, that control a biological process. Low molecular weight chemically produced substances are primarily referred to as small-molecule medicines. While many discovered compounds have a higher molecular weight than a typical small-molecule medicine, which may have a molecular weight below 500 Da. By creating complexes with their targets, small chemicals can influence how different proteins function, including protein-protein interactions. Small molecule drug discovery is a challenging process that calls for a wide range of skills and numerous methodologies. Through phenotypic screening with cell-based assays that enable the discovery of new targets, small compounds can be acquired. These compounds can also be acquired by target-based drug discovery, which typically entails candidate selection, assay development, Identification of the target, confirmation of the target, hit detection, hit to lead conversion, lead optimization, and further development. Small-molecule drug development relies heavily on medicinal chemistry since medicinal chemists choose the methods to be used for compound alterations. It is well knowledge that challenges with drug discovery include choosing a target, identifying an early hit, optimizing leads, and efficacies [22].

Molecule Similarity

To create molecules that are chemically superior to a target molecule while remaining identical to it is the goal of molecular optimization [23]. The methods of Molecular Similarity are so important in the fields of chemicals and pharmaceutical area. Such as protein-ligand docking, biological activity prediction, and database searching. Because of the similar property principle, finding similarity between compounds is very practical and it states that similar structures in molecules can result in comparable characteristics. Accordingly, many things in biological research can be achieved by finding the structure similarity between molecules. Nevertheless, finding the structural similarity between molecules is complicated since it is not a measurable property for them. Sometimes, external criteria cause different interpretations and that makes it hard to develop a precise computational method to find the molecular similarity.

Three basic parts are important to make a measure of molecular similarity: First, the representation of the molecule that encodes its features and at the same time includes their associated weights. Second, a comparing method for these representations to calculate the difference which means a small difference makes it more similar. Third, a function to evaluate the similarity between them [24].

In chemistry-related professions, determining how similar molecules are structurally to one another is a foundational work that can be advantageous for many tasks that come later. In general, properties of molecules are likely to be similar for those with similar structures. However, a small alteration in a molecule's structure can frequently result in a significant shift in the way that the molecule behaves and performs. Therefore, a fascinating and essential task in domains related to chemistry is determining how structurally similar different molecules are.

The examination of local and global structural similarity between a novel medication and old pharmaceuticals in the drug discovery process are the only two downstream activities that can profit from the capture of structural similarity between molecules. The search for molecules that are structurally similar to a given query molecule in chemical databases is one of the additional tasks [20]. Cheminformatics is built upon the idea of molecular similarity. It implies that there is a propensity for molecules with "similar" structures to have comparable properties.

The most popular methods for determining molecular similarity involve recording the molecule as a vector of numbers, which enables us to compare the vectors that represent two molecules according to their Euclidean or other distance. The Jaccard or Tanimoto similarity (TS) metric is frequently used when dealing with binary strings (between zero and one). Creating such a vector for the molecule involves, among other things, calculating (or measuring) the various "descriptors" of the molecule, such as clogP or total polar surface area, from the molecule's structure. The utilization of structural properties directly and their encoding as so-called molecular fingerprints is a more popular method for producing the encoding vector of numbers. The MACCS, atom pairs, torsion, extended connectivity, functional class, circular, and other well-known ones are only a few instances. Once encoded, the similarities can be compared to their Jaccard or Tanimoto counterparts. Sometimes a "difference" or "distance" is indicated and expressed as 1-TS (a true metric). The greatest and most well-known framework for carrying out all of this is RDKit (Pathon v3.6.8) (www.rdkit.org), which offers nine methods for developing molecular fingerprints at the moment. The issue is that the molecules that are "most similar" to a target molecule frequently differ greatly, both qualitatively and quantitatively in terms of the TS values of the various fingerprints [25].

QSAR

Hansch and Fujita, in the 1960s, rediscovered Quantitative Structure-Activity Relationship (QSAR), originally identified by Hammett in the 1930s. Over 70 years, QSAR evolved with approaches like 4DQSAR, HQSAR, 2DQSAR, and 3DQSAR, comparable to CoMSIA and CoMFA. Applied in diverse chemistry branches, including pharmaceutical, agricultural, environmental, and toxicological fields, QSAR is an established method. In pharmaceutical chemistry, it is pivotal for drug innovation, integrated into industrial drug design tools for over 50 years. QSAR relies on molecular data, providing structural and physical insights, aiding in the computer-aided identification of potential compounds. This approach accelerates compound synthesis, conserves resources, and facilitates the creation of materials, medications, and more. Despite challenges, the surge in QSAR research papers underscores its rapid advancement, emphasizing the importance of selecting appropriate descriptors, be they theoretical, empirical, semi-empirical, or derived from experimental traits, for an efficient connection[26].

Euclidean distance

Here, explain Euclidean distance because it was considered in our research work. The following formula can be used to determine how closely data match the Euclidean Distance formula in Eq 1:

$$d_{ij} = \sqrt{\sum_{k=1}^{n} (x_{ik} - y_{jk})^2} \quad \text{…….} \ 1$$

Where i acts as the cluster data center, j is the attribute data, k is the indicator for each data, and n is the total amount of data, and d is the distance between i and j. Data at the cluster center are denoted by x ik and data on individual nodes are denoted by yjk [27, 8].

Manhattan Distance

It is used to pinpoint the precise coordinate difference between two objects. The formula is as follows in Eq 2:

$$d_{ij} = \sum_{k=1}^{n} \left| x_{ij} - y_{ik} \right| \quad \text{…….} \ 2$$

Where i acts as the data center for the cluster, j is the data on the attribute, k is the symbol of each data, n is the total amount of data, xik is the data at the cluster center to k, and yjk is the data on each data to k[27]. The difference between Manhattan distance and Euclidean distance. Instead of the straight line distance between two locations, this distance is the sum of their east-west and north-south distance[28].

Related Work

In [24] the authors categorized their similarity measures in the literature into two classes: graph-based representation and vector-based representation. The first one represents the molecule as a graph depending on the molecular atom-bond structure and the comparison between molecules is achieved by graph similarity techniques. The approach which has been developed as a result of graph similarity techniques for comparison is limited to specific graph structures. The second one is called fingerprint which is a familiar concept in chemical informatics fields. It is a binary vector indicating the features of the molecule. Bit 1 represents the presence of molecule-associated feature and bit 0 represents its absence. It has computationally efficient pairwise comparisons which are easy to use. In this case, some comparison measures are used like Dice, Cosine, Tanimoto, and Euclidean distance. This class also has some drawbacks. In a molecular graph, fingerprints do not have the ability to assess for certain a particular pattern whether it is present or absent and that is because in the fingerprint for a pattern when a bit is set to 1 means it is present with some probability only. in addition, this approach does not consider molecular topology. Graph representation of a chemical compound is called a molecular or chemical graph which is categorized based on the structural dimensionality of the molecule into two-dimensional and three-dimensional. Each vertex is representing one atom. The edge between two vertices is representing their chemical bond. In the three-dimensional molecular graph, the edge between each pair of vertices is representing that the two corresponding atoms have a geometrical distance between them. The authors in their paper represented the molecule by the labeled reduced graph in three steps which they did explain by building a molecule's basic structural representation, identifying ring structures, and adding them as a vertex to the reduced graph.

In [20] The authors analyze two molecules using graph - neural - networks, output their representations, and then include the output of the two representations into a regression model to forecast the real ground truth. They point out that the calculation of the graph edit distance (GED), a widely used metric for determining how similar molecules' structures are to one another, is an NP-hard problem. According to them, the experimental findings demonstrate that their model performs noticeably better in GED prediction than previous molecular representation learning techniques. Additionally, it is demonstrated that their model is far faster than the technique used to determine the precise GED. The suggest methodology can serve as a guide for comparable drug discovery and molecular retrieval tasks. The authors investigate the issue of determining how similar molecules' structures are. The structural - similarity between molecules is typically expressed in terms of graph similarity because a molecule's structure may be represented as a graph. In general, there are numerous techniques for determining how similar two graphs are, such as iterative algorithms, feature extraction techniques, and graph edit distance. Here, they opt for graph edit distance (GED), a simple and purpose-built metric that works with all different kinds of graphs. However, one well-known problem with calculating GED is that it is an NP-hard task, which means there isn't an algorithm that can do it in a polynomial amount of time. They suggest employing graph neural networks to approximately calculate GED in order to handle the challenging problem of doing so. A class of neural networks called graph - neural - networks (GNNs) is used to process data that is represented by graph data structures. Common GNNs use a neighborhood – aggregation - technique, which incrementally updates a node's representation by combining that of its neighbors and itself. The full graph representation is then obtained by averaging all node representations. They process a pair of molecular graphs using a GNN to produce their corresponding representations, which are then input into an MLP regressor to approximate the ground-truth GED. It has been theoretically demonstrated that graph representations learned by GNNs can preserve the graph structure. As a result, the entire model may be trained from beginning to end. They use actual molecular datasets to empirically test the performance of their suggested model. The experimental findings demonstrate that their model is capable of accurately predicting the GED between molecules, with a prediction error that is significantly lower than the range of ground-truth GEDs. Additionally, their model performs noticeably better than non-GNN molecular representation learning techniques. In comparison to the best baseline approach, it is able to decrease the rooted mean squared error (RMSE) by 18.6%–31.1%. Additionally, their approach cuts the running time by 18.5 seconds to 702.5 seconds and performs substantially faster than the precise GED calculating technique. They also show that their model can really capture the structural details of molecules by seeing the chemical representations it has learned.

In [29] Particle Swarm Optimization, a more lightweight heuristic optimization method, is recommended by the authors (PSO). A huge library of molecular building blocks and chemical interactions serve as a representation for the discrete chemical space Hartenfeller suggested to use PSO to in 2008. In this case, they employ PSO to provide a continuous chemical representation. Since the swarm's particles go through this simulation, which corresponds to actual molecules in the chemical space, they refer to their method as "molecular swarm optimization" (MSO). They demonstrate using three separate experiments how the suggested method may be used to optimize molecules with respect to a single objective, while subject to restrictions involving chemical substructures, and with respect to a multi-objective value function. This can be used to represent the chemical space latently via a compressed embedding. The latent space that results from the model's training on a large dataset of over 75 million chemical structures from diverse sources represents a wide range of chemical space that can be investigated. In their prior work, they also showed how to effectively build molecular descriptors for QSAR models using the learned molecular representation (quantitative structure-activity relationships). Furthermore, when changes in the latent space are reversed, smooth changes in the discrete chemical space's structural and molecular properties follow. For technical details concerning our system, interested readers are advised to read the original publication. They asserted that their method might fast enhance a certain starting chemical's anticipated drug-likeness or biological activity. This demonstrates how their approach to optimization can easily navigate the chemical space created by their embedding that has been pre-trained while still completing single-parameter optimization in a timely manner. However real-world drug design circumstances are substantially dissimilar from these examples. No structural constraints are used now to manage the structure's growth. This implies that the novel, improved compounds might have a structural variation from the indicated starting points or include moieties that are hazardous or unstable. New drug discovery initiatives usually focus on chemical families and their analogs. As a result, in the section that follows, they advise putting limitations on the chemical structure when optimizing. They contend that their techniques can successfully maximize molecules for a number of goals, such as increasing target binding affinity, partition coefficient logP, or anticipated drug-likeliness as measured by a quantitative structure-activity relationship (QSAR) model. Their suggested approach outperforms the baseline approach in terms of locating the best solutions while achieving a significant decrease in processing time. In the more standardized benchmark package GuacaMol, here it exceed baseline approaches in 9 out of 12 tasks that weren't already fully addressed by the best baseline method. They conclude by demonstrating the effectiveness of the suggested technique in additional trials. It must be stressed that although the optimization cycles in this study produce positive results for improving molecular properties, they are still based on expected values for the qualities. This can be highly problematic when QSAR models are applied in circumstances outside of their intended field of use. Thus, they advise restricting the application of our suggested technique to areas of the chemical space that can be adequately represented by the used functions. Using active learning to integrate in-silico optimization with practical experiments would be an even better approach. So, while extending into uncharted chemical space, the QSAR model might be altered, perhaps preserving some level of predicted accuracy.

In [30] the Peroxisome proliferator-activated receptor Y (PPARy) is a target for the onset of diabetes mellitus, and its agonists have been developed as a medication that serves as an important research tool, according to the authors. A number of N-benzyl benzamide derivatives have been subjected to 3D-QSAR and molecular docking investigations using (CoMFA), (CoMSIA), and surflex-dock approaches to develop models that look at the structural requirements of PPARy agonists. The main structural characteristics of PPARy agonists that are responsible for biological activity may be identified based on the data, according to the scientists, who asserted that these models have excellent statistical reliability and good predictive potential. They stated that this study might be valuable in elucidating the PPAR agonist potential of N-benzyl benzamide derivatives. The training set consisted of 27 compounds, and the test set included 6 compounds. In the training and test sets, the CoMFA and CoMSIA models showed a respectable capacity for prediction. Compound 24 was chosen as the template and related reference molecule because it had a higher potency than compound 19, which was discovered to be the case. Surflex-dock was used to dock the substances to PPAR's binding site and get potential binding conformations. The newly created compound N1, N9, and N12 have docked to the relevant binding sites in addition to the template (compound 24c). Despite having the highest activity in the training set, compound 24c, compounds N1, N9, and N12 have higher docking scores. The results of the current study showed that the CoMFA and CoMSIA models might be used to design new, powerful compounds and to calculate the QSAR of PPARy agonists. This also helped forecast the chemical composition of the newly created drug and the significant activity characteristic effects on PPARy. The Distill alignment approach has been used to accomplish molecular alignment, a crucial step in the creation of CoMFA and CoMSIA models.

The Study limitation

The proposed study, utilizing a hybrid system of Aquila and Termite behavior combined with Neural Networks, aims to determine the most similar molecule in a dataset compared to a target molecule using a 3D grid and Quantitative Structure-Activity analysis. While this approach presents innovative aspects, it is essential to acknowledge its limitations:

The effectiveness of the integrated hybrid system heavily relies on the parameters and settings of the Aquila Algorithm, Termite Behavior, and Neural Networks. Variability in these parameters may lead to different results. Additionally, the generalization of the model to diverse datasets and molecular structures might be a challenge. The success of the Aquila Algorithm, which controls the rotation and translation of molecules, is contingent on the initial parameters and random number generation. Sensitivity to these initial conditions may affect the reproducibility and robustness of the results. The implementation of swarm intelligence, particularly the Aquila Algorithm and its emulation of swarm behaviors, introduces complexity. The intricate dynamics of swarm intelligence algorithms may be challenging to fully comprehend and control, potentially impacting the reliability and interpretability of the results. The study focuses on the quantitative structure-activity relationship (QSAR) as a measure of similarity. While QSAR is a valuable metric, it may not capture all nuances of molecular similarity, especially in cases where specific molecular interactions play a crucial role. The approach assumes uniformity in the representation of molecular structures in the 3D grid. Variations in molecular complexity and size may not be fully accounted for, potentially leading to oversimplifications in the similarity assessment. The process of organizing molecules in 3D grids and applying the integrated hybrid system may demand significant computational resources. This limitation can impact the scalability and accessibility of the method, especially for large datasets.

Methods:

By organizing molecules in space and assigning their molecular fields to a 3D grid, R.D. Cramer with M. Milne established the first 3D comparison of molecules in 1979 [31]. The current research trying to find the similarity between some three-dimensional molecules in the data set compared with one molecule called the target molecule. In fact, it aims to find the most similar molecule in the data set (sample molecule) compared to the target molecule. Quantitative Structure-Activity has been used to measure the similarity between them. The first thing is to fetch the molecules from its file and represent them in three-dimensional way. Each molecule is a number of atoms represented as three-dimensional coordinates. To measure the molecule's quantitative structure-activity it is putting in a three-dimensional grid with particular lengths (x, y, and z).

All operations above are controlled by the Integrated Hybrid System of (Aquila and Termite) Behavior and Neural Networks Algorithm. First, use the Aquila Algorithm which is a type of swarm intelligence that controls the rotation and the translation of the molecule. It uses a method to produce random numbers that are used to rotate and translate the molecule and during the time the range of these numbers gets less same as the Aquila when it attacks the prey it closes the angle of the attack. Here, it is mimic how the Aquila expands and decreases the space of its circle. Then, use the generation of Aquila's behaviors as a swarm of termite workers. considering each method as a termite worker and dedicate each worker to rotate and translate one specific sample molecule from the dataset. And use a function that examines each step of the transformation and it considers each progress the same way when the foraging termites emit the pheromones while digging underground.

Neural Network is used by finding the similarity or the difference between the sample molecule and the target molecule. After passing a molecule into the three-dimensional grid, calculate each point's value in the grid. In this way it gets two vectors of numbers, one sample vector and one target vector. After performing the Swarm Intelligence (Aquila and Termite) steps and the methods of translation and rotation to change the positions and the gesture of sample molecules a specific number of times, it made the molecule number (7) so similar to the target molecule. The first step in our Neural Network is to pass the sample molecule number (7) and the target molecule to the Neural Network class. Then, calculate the difference between each corresponding sector of the vectors. In fact, each vector is divided into five sectors (five neurons in Neural Networks).

The calculated value of the difference between each sector considers the weight value. So, in this case, here it is getting five weight values. The weights values in this Neural Network are fixed. Each time performing the algorithm of Aquila and Termite, the class of Neural Network is recalled to pass one sample molecule after it has been translated and rotated (transformed). After passing the sample molecule into the three-dimensional grid and getting its vector of numbers, it gets passed again into the Neural Network's class. The class divides the vector into five sectors (this consider the first layer of the Neural Network). In the second layer, it compares each sector of the sample molecule (sample vector) to the target molecule's sector (target vector's sector). Then it compares each result with the fixed corresponding weight. Now, there are five results after performing the second layer and if three of these results' values are bigger than their corresponding weights, the Neural Network's class will find their summation and considers it as the similarity between the two vectors. The condition of passing three bigger numbers here is considering the threshold of the network. The Fig 1. below shows a three-dimensional molecule in a three-dimensional grid:

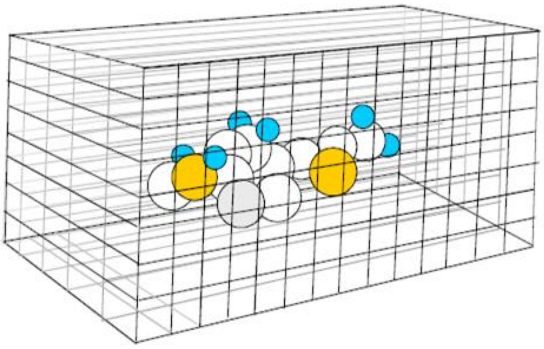


Figure 1. One Molecule in a Grid.

First, to find the quantitative structure-activity for the target molecule, find the distance between each atom in the molecule and the first point of the grid by using Euclidean distance. Then do the same mechanism with all points of the grid and save the results in a vector named TMVector (Target Molecule Vector). The Fig 2. below shows how to measure the QSA for Target Molecule using Euclidean Distance:

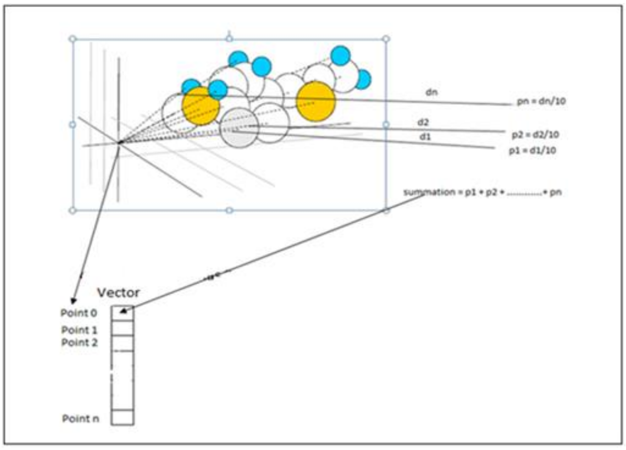


Figure 2. Measuring the QSA for Target Molecule by Euclidean Distance.

After downloading the molecule from the particular file. Use a local loop and each cycle of the loop represents dealing with one molecule. In the beginning, translate (shift) the molecule in a particular distance to the positive side of the grid to avoid using negative numbers. Then translate and rotate the molecule (transform) by particular dimensions to make it more similar to the target molecule. Here, it measures the quantitative structure-activity for the first sample molecule and save it in the SMVector (Sample Molecule Vector).

Now, compare the sample molecule with the target molecule using Manhattan distance to measure the difference between them and save the difference in the first room of an array called the local array initiated before using the local loop and it has the same loop's length. In each loop cycle, save the difference in the next room of the local array. The Fig 3. below shows how to measure the difference between two molecules using the Manhattan distance:

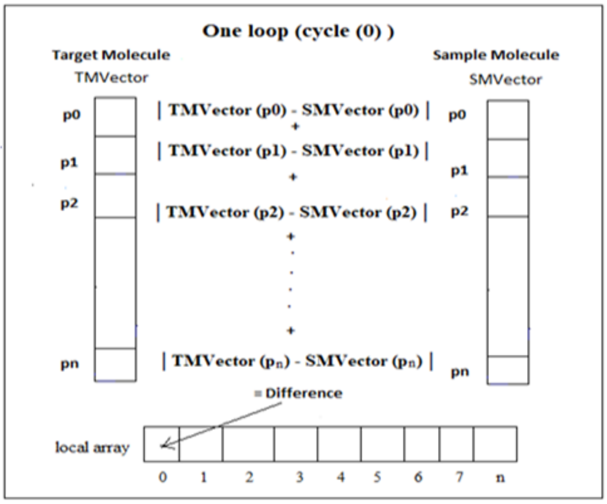

One loop (cycle (0))
Target Molecule
TMVector
Sample Molecule
SMVector
p0
p1
p2
pn
| TMVector (p0) - SMVector (p0) |
+
| TMVector (p1) - SMVector (p1) |
+
| TMVector (p2) - SMVector (p2) |
+
+
| TMVector (pn) - SMVector (pn) |
= Difference
local array
0
1
2
3
4
5
6
7
n

Figure 3. Measuring the difference between Target Molecule and Sample Molecule using Manhattan distance.

After finishing the local loop's cycles here gets an array full of numbers and each number represents the difference between one molecule of the data set compared to the target molecule. To use the Aquila algorithm, use another loop with particular cycles called the Generation loop. The Generation loop holds the local loop which represents one termite in colony. In each cycle of the global loop, the array rearranges in an ascending manner to arrange the sample molecules in the same order and save it in an array called DiffValArranged. At the same time, it saves the index of the smaller difference in the local array which is the index of the molecule in the data set in the third array called the final array. After finishing the global loop, the final array is check which is full of indexes and find which index that frequent the most (this considers foraging termites emit the pheromones while digging underground). The most frequent index is the index of the most similar molecule in the data set comparing to the target molecule. The Fig 4. below shows how to rearrange the values in the local array to DiffValArranged array:

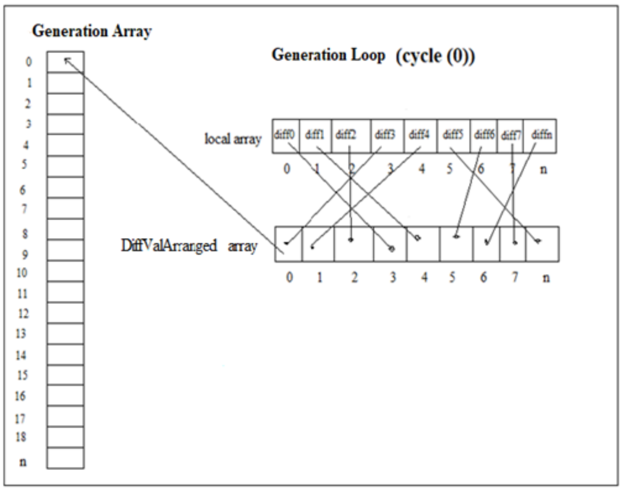


Figure 4. Rearrange the values in local array to DiffValArranged array.

Experiment

A. Data Loading Phase:

A file named steroid has been used to save data about the molecules information. It is a collection of coordinates containing decimal positive and negative numbers and it is arranged by placing three numbers in each line to represent an atom. Three java classes have been used to load the data from the steroid file.

The first class is named Molec Calculator. Its task is to calculate how many molecules there are in the file and it does so by using a method named (read) creating three objects of three classes that have been imported from the java library and they are (File, File Input Stream, and Buffered Reader). It uses a while loop to read lines from the file and it increases the counter (molecule Count) by one when it reaches the space line. Ultimately the method returns the number of molecules in the file by using the variable (moleculeCount). At the end of the method, the molecule Count has been increased by one because the object of the Buffered Reader class does not read the last space in the file.

Here is a segment of a simple sample molecule within the dataset, comprising a group of atoms.

Atoms with values (x, y, z)	Atoms with values (x, y, z)	Atoms with values (x, y, z)	Atoms with values (x, y, z)
-4.9613 -3.282 1.0341	1.6684 1.6262 -1.708	2.9306 1.0507 -3.7533	1.2143 -2.2798 -0.7817
-2.6356 -2.5292 1.3053	1.861 0.1881 -1.2092	-3.3381 0.3008 -1.345	1.8226 -1.7812 0.8272
-1.4762 -1.834 0.645	3.2881 0.1936 -0.641	-2.7969 -1.3371 -1.8326	0.7268 0.3834 0.652
-0.2448 -2.3478 0.7521	4.0475 1.1486 -1.5906	-5.0735 -1.4247 -0.9018	-0.5578 -0.8753 -1.8356
0.9717 -1.7195 0.1333	2.9826 1.8199 -2.4972	-4.5163 -0.6493 0.6171	-0.9444 2.0074 -0.8647
0.7383 -0.2456 -0.2491	1.5984 2.639 -0.5398	-3.5147 -3.3397 -0.5033	-1.7211 1.2253 -2.2731
-0.6025 -0.1468 -1.0122	-2.031 0.526 0.9938	-5.299 -2.7675 1.7582	0.1926 2.6641 -2.9443
-1.7795 -0.548 -0.0879	3.306 3.259 -2.8074	-2.3304 -3.5268 1.6532	0.4493 0.9412 -3.378
-0.8172 1.2497 -1.649	4.3409 3.7841 -2.430	-2.9663 -1.9355 2.17	1.8487 -0.4836 -2.0809
0.3726 1.6562 -2.5452	2.304 4.0411 -3.6069	-0.1116 -3.277 1.303	3.7242 -0.8162 -0.6337

The second class is named Atom Calculato. Its task is to calculate how many atoms are there in the file for each molecule and it does so by using a method named (read) creating three objects of three classes that have been imported from the java library and they are (File, File Input Stream, and Buffered Reader). At the beginning, an array of integer numbers named (atom Numbers) is get created to hold atoms' numbers. The length of the array is the same as the molecule number in the file which has been obtained by the Molec Reader class. It uses a while loop to read lines from the file and it increases the counter (molecule Count) by one when it reaches the space line and assigns a counter named (atom Coun) to zero. if...else statement is used to check whether it is a space or a new data line. When it is a data line the program increases the counter (atom Coun) by one each time. Ultimately the method returns an array of integers that represent atoms' numbers in each molecule in the file by using the variable (atom Numbers).

The third class is named data Reader. Its task is to read molecules' coordinates and save them into an array of molecule types and each molecule is an array of coordinates (atoms). It works by using a method named (read) creating three objects of three classes that have been imported from the java library and they are (File, File Input Stream, and Buffered Reader). It initializes two counters (molec Count and atom Count) to be used inside the while loop which read lines from the file. It initializes an array to save molecules named Molecs Array with a length equal to the number of molecules in the file. It initializes another array to save atoms in each molecule in the previous array and its length varies from one molecule to another by initializing it periodically and regularly inside the loop. The method uses an if...else statement to check whether it is a space or a line of data. It uses the split method to separate values and the pars Double method to change the values from String type into Double type and finally save it into a coordinate type. Ultimately the class returns an array named molec Array which is an array of molecules and each molecule is full of atoms (coordinate types).

B. Representing Phase (Basic Classes):

Molec3D: It is a class to represent three dimensional molecule. It has a field side to define its properties such as atom Number, atom Index, and an array of Molec3DCoord. To be more specific, the Molec3D is an array of Mol3DCoord class.

Mol3DCoord: It is a class to represent a three-dimensional atom in the molecule. Its data field contains x, y, and z variables to represent the three-dimensional coordinate. It has a constructor to make the object of coordinate and it has three parameters. The most two important methods for this class are distance To and power To. The first one (distance To) is used to measure the distance between two coordinates. In our software, it has been used to measure the distance between the point of the grid and the atoms' coordinates of the molecule. In fact, it has been used to measure the distances for all points of the grid according to the molecule's atoms. The second method (power To) is used to measure the power of the molecule atoms that affect the points of the grid. So, it calculates the power for each point of the grid according to the molecule's atoms inside the grid and it does so by dividing the distance by 10.

Mol3DGrid: it is a class to represent the grid used to hold a molecule to measure the distances and powers of its points according to the molecule's atoms. It has a data field with three variables xAxis, yAxis, and zAxis. Also, it has a constructor to construct an object to use through the software.

C. Operation Phase:

Operations: Class to hold methods that are used to do operations on the molecule and they are listed below: change MtoArray: a method that receives one molecule and changes it into an array of Mol3DCoord (coordinates) to deal with its variable (x, y, and z). It returns an array of three-dimensional coordinates. translation: a method that receives one molecule and three numbers (x, y, and z) to shift the molecule in three dimensions. Ultimately, it returns the molecule with new positions. Get Grid Power: a method that receives an array of three-dimensional coordinates which represents one molecule and an object of Mol3DGrid. It calculates the distances between one point and the molecule's atoms. It obtains the power of the point by dividing the summation of the distances by 10. When it mention the power of the point, it means the Xray power that gets produced by the molecule's atoms. The method uses three loops to visit each point in the grid and during that it calculates the power of each point. It returns an array of doubles types and each room in the array represents one point's power. Fig.5. below explains how to calculate points power inside cubic grid:

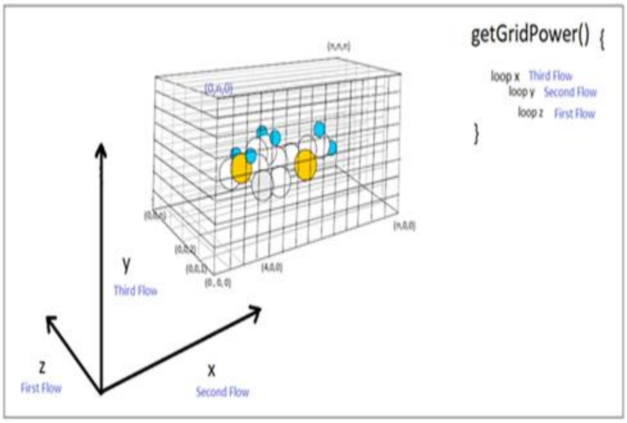


Figure 5. Explains how the method calculate points power inside cubic grid.

Get Molec Difference: a method that receives two arrays and each one holds powers of the grid's points. The first array represents the target molecule and the second one represents the sample molecule. It uses Euclidean distance to find the difference between two arrays. The method returns a double value type which represents the difference between two molecules. Get Centroid: a method that receives an array of three-dimensional coordinates which represents one molecule. Its task is to obtain the centroid of the molecule. The method returns a three-dimensional coordinate that represents the centroid of the molecule.

The flowchart of software is explained below:

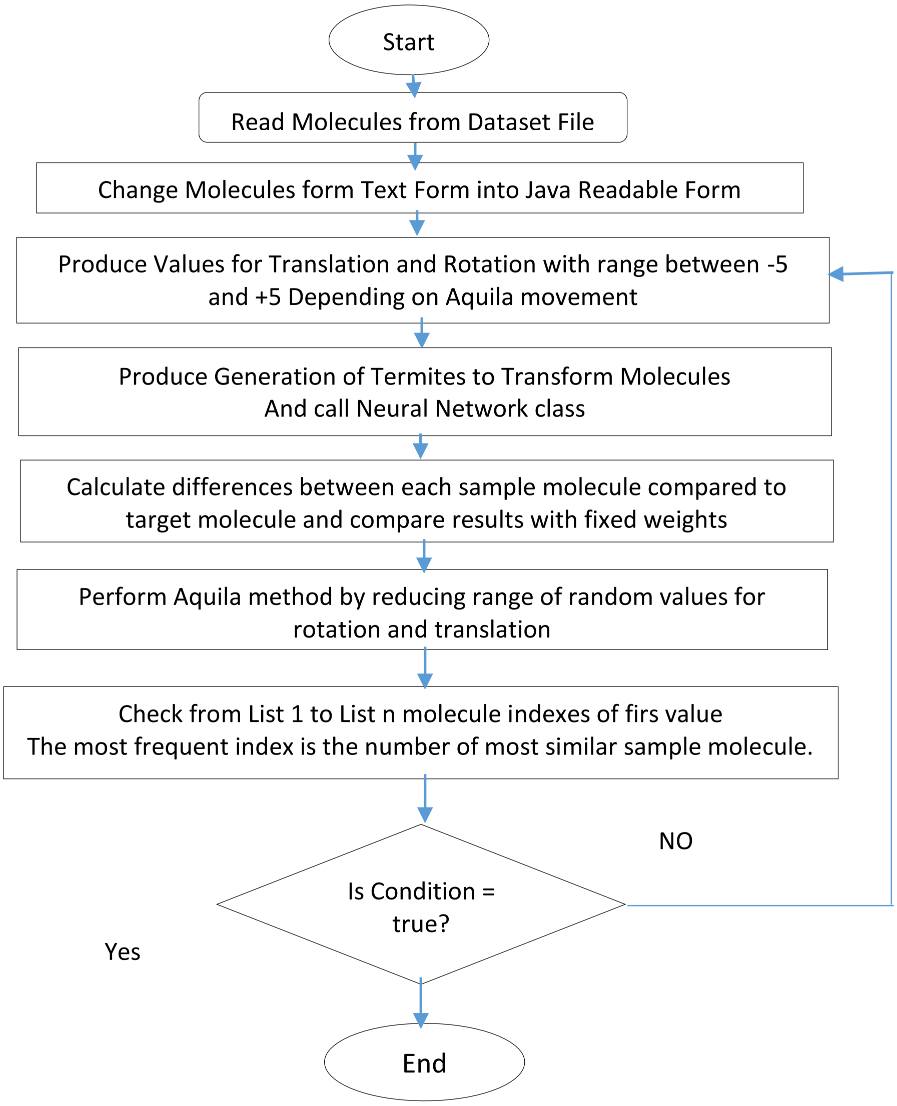

Start
Read Molecules from Dataset File
Change Molecules form Text Form into Java Readable Form
Produce Values for Translation and Rotation with range between -5 and +5 Depending on Aquila movement
Produce Generation of Termites to Transform Molecules And call Neural Network class
Calculate differences between each sample molecule compared to target molecule and compare results with fixed weights
Perform Aquila method by reducing range of random values for rotation and translation
Check from List 1 to List n molecule indexes of firs value The most frequent index is the number of most similar sample molecule.
NO
Is Condition = true?
Yes
End

Figure 6. The Steps of Software Algorithm

Results and Discussion

Here, one experiment was conducted using the hybrid system and the results can be seen in Table 1 below:

Table 1. Shows the results of performing one loop of the Hybrid System

Termites Generation Number: 0										
Aquila movement (Molecules Transformation)					x = -1.0	y = 0.0	z = -3.0	Rx = -2.0	Ry = -5.0	Rz = 3.0
Difference between target molecule and sample molecules from number 0 sample molecule to number 10sample molecule calculated in G-P unit										
3.17	3.11	3.7	3. 24	3.37	3.14	3.05	2.72	2.99	3.33	3.86
Differences in Ascending Arrangement										
2.72	2.99	3.05	3.11	3.14	3.17	3.24	3.33	3.37	3.7	3.86
(Molecule Number)										
7	4	3	0	5	1	10	6	9	8	2

Here you can see from the results in Table 1 that sample molecule number (7) is the most similar molecule to the target molecule because it has lowest difference most of the time compared to the target molecule. The first row in Table 1 represents the generation of the termites. The second row represents the values for translating and rotating the sample molecules. The third row shows the values of the difference between the target molecule and sample molecules from number 0 to number 19 and GP (Grid Power) represents the unit to measure the difference. The fourth row shows the differences in ascending arrangement and the last row shows the indexes of the sample molecules depending on ascending arrangement in fourth row. After performing the program for hundred times, the results that show the most similar molecule to the target one is molecule number (7). When doing the steps to obtain the standard deviation for the optimal molecule (number 7) depending on its position compared to the other sample molecules' positions in the arrangement array after performing the program 30 times and got the standard deviation with a value of (5.03).

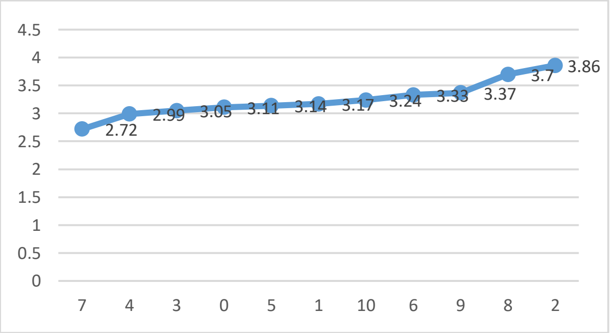
Table 2. Shows the results of performing one loop of the Hybrid System.

Termites Generation Number: 29										
Aquila movement (Molecules Transformation)					x = 5.0	y = 0.0	z = 4.0	Rx = -4.0	Ry = -4.0	Rz = 0.0
Difference between target molecule and sample molecules from number 0 sample molecule to number 10sample molecule calculated in G-P unit										
29.76	24.5	19.67	18.16	18.14	22.54	28.72	19.5	30.73	29.76	30.1
Differences in Ascending Arrangement										
18.14	18.16	19.5	19.67	22.54	24.5	28.72	29.76	29.76	30.1	30.73
(Molecule Number)										
0	4	7	3	5	1	10	6	9	8	2

In Table 2 sample molecule number (7) is the third similar molecule compared to the target molecule and that depends on the values of translation and rotation that the Hybrid System has been used.

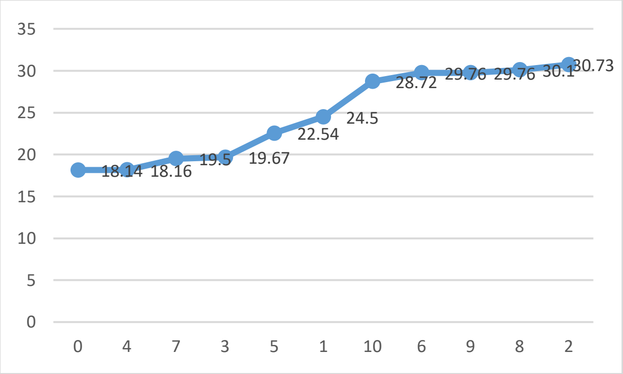
In one of our experiments, here used six parameters to translate and rotate the sample molecules in the data set. Three parameters for translation: (x = -1.0, y = 0.0, z = -3.0) and three parameters for rotation: (Rx = -2.0, Ry = -5.0, Rz = 3.0). and got the most similar sample molecule compared to the target one is molecule number (7) as shown in chart 1. below:

Chart 1. Ascending order of the differences between sample molecules and target molecule.



When different parameters were used, different results were appeared but still near. Six parameters were used to translate and rotate the sample molecules in the data set. Three parameters for translation: (x = 5.0, y = 0.0, z = 4.0) and three parameters for rotation: (Rx = -4.0, Ry = -4.0, Rz = 0.0). find that the most similar sample molecule compared to the target one is molecule number (0) and sample molecule number (7) is the third most similar sample molecule as shown in chart 2 below:

Chart 2. Ascending order of the differences between sample molecules and target molecule.



To check the accuracy of our work first decided to find the centroid of the target molecule and the centroid of molecule number (7) which is the most similar sample molecule to the target one. Then translate both of them to the center of the three-dimensional grid which is the point (6, 6, 6) manually and respectively without using our Hybrid System. After that, found the difference between them. After applying in above, got the difference which is 1.92 GP and the difference when use the Hybrid System was 2.72 GP. To find the difference, apply the next steps: Percentage decrease = |2.72 – 1.92| / 2.72 = 0.8 / 2.72

= 0.29411764705882 = 29.411764705882% So the accuracy is 70.588235294118%.

Our neural network has an input layer, a hidden layer, and an output layer. Five neurons make up the input and hidden layers. Each neuron in the input layer finds the summation of the four values of the sample vector. Each neuron in the hidden layer calculates the value of F, which is the result of subtracting x from y, and then compares it to the fixed weight. The counter will go up by one if the outcome is less than the matching weight. Y1 represents the summation of the first four values of the target vector. Y2 represents the summation of the second four values of the target vector and so on with y3, y4, and y5 respectively. The neuron in the output layer finds the summation of f values which are the result of the hidden layer and passes it after determining that the counter register's value is greater than three, which is the threshold of the output layer.

After finding that the molecule number (7) is so similar to the target molecule during changing its position and gesture. then used it to fix the values of the weights in our Neural Network by subtracting the values of the sample vector's sectors (vector number 7) from the values of the target vector's sectors. Now, perform the algorithm of Swarm Intelligence (Aquila and Termites) and during that, call the class of Neural Network with each step to get the similarity of the difference between each molecule of sample molecules and the target molecule. Below a sample of the results obtained during the performance of the Integrated System. (generation 498 and generation 499):

Termites Generation Number: 498
Aquila movement (Molecules Transformation) by: x = 0.2 y = 0.12 z = 0.02Rx = -0.09 Ry = -0.09 Rz = -0.08
Calling Neural Network on Each Molecule

Termites Generation Number: 499
Aquila movement (Molecules Transformation) by: x = -0.08 y = -0.13 z = -0.18Rx = 0.12 Ry = -0.23 Rz = -0.04
Calling Neural Network on Each Molecule

The best similarity was achieved at Molecule num: 4
MolecNO: 4: TransNO: 473 Diff: 0.23 TransVal: [x=0.0 y=0.0 z=0.0 Rx=0.0 Ry=0.0 Rz=0.0]

From the result above it can be seen that using Neural Network integrated with Swarm Intelligence gives us more precise similarities. By using the integrated system for 100 times, got more than 90 percent of the results denoted that molecule number 4 is the most similar to the target molecule and that after performing precise transformation on it (translation and rotation) with very small poses in the three axes. Because of our experiments above can say that the accuracy for the integrated system is 90 percent.

Conclusion

The Meta-Heuristic mechanism was used in this study. It covered how to locate the sample molecule in a data collection that is most comparable to the target molecule. The methodology was dependent on quantitative structure-activity relationships. To advance the pharmaceutical sector, the molecular similarity is sought after. Molecules have been measured using the Euclidean distance steps on a three-dimensional grid. Grid points have been employed to represent each molecule in space, and they were an effective instrument for managing the fitness function's development and Swarm Intelligence's steps. After measuring molecules in a three-dimensional grid, the Manhattan distance was used to determine how different the molecules were from one another. The software code was written in the Java programming language, and made use of its library of programming classes to provide ways for representing three-dimensional molecules, three-dimensional grids, translation, and rotation methods, as well as numerous other methods to compare molecules. To get the results, every method was entirely coded from scratch. The study read a few articles that were beneficial since they gave us experience dealing with molecular similarity and swarm intelligence. Many subjects have been discussed during the research such as Molecules, Small Molecules, Drugs, Molecule Similarity, Similarity Approaches, Molecular Descriptor, Quantitative Structure-Activity Relationship, Quantum Similarity Theory, Receptor and Ligand-based approaches, Molecular Docking, Euclidean Distance, Manhattan Distance, Swarm Intelligence, Ant colony optimization, Particle swarm optimization, Artificial bee colony, Firefly algorithm, Cuckoo Search Optimization, Aquila Optimizer, and Termites Colony. The behaviors of the hybrid system of Aquila and termites have been used to guide the operations to discover the most similar molecule above. The findings demonstrated that Swarm Intelligence (the Hybrid System of Aquila and Termites) in addition to Neural Network, a successful way to locate the most similar molecule in the data set compared to the target molecule, is a viable response to the study issue.

Recommendations for Future Work

Future research in vector similarity computation should explore ensemble approaches to combine diverse deep learning models, optimizing for robustness. Investigate dynamic learning rate strategies for adaptive optimization, extending transfer learning to leverage pre-trained models from various domains, and handling temporal aspects in datasets. Enhance explain ability techniques for interpreting neural networks' decisions in vector similarity, and assess robustness against adversarial attacks. Hybrid models, combining deep learning with traditional techniques, can offer a balanced approach. Tailor investigations to application-specific needs such as content recommendation and anomaly detection. Integrate user interactions for adaptive models. These strategies aim to refine deep learning methods, advancing accuracy and robustness in vector similarity computation, impacting diverse sectors like machine learning, recommendation systems, and data analysis.

Acknowledgment

I want to express my appreciation to my supervisor Prof. Dr. Hakan Koyuncu for all support during my study. It is a pleasure for me to express my gratitude to my family who shared with me all the difficulties and beautiful moments during my studies for the Master's degree.

Authors’ Declaration

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are ours. Furthermore, any Figures and images, that are not ours, have been included with the necessary permission for re-publication, which is attached to the manuscript.

- No animal studies are present in the manuscript.
- No human studies are present in the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee at Altınbaş University, Istanbul, Turkey.

Authors' Contribution Statement

F.S. and H.K. contributed to the design and implementation of the research, to the analysis of the results and to the writing of the manuscript.

References

1. Tang J, Liu G, Pan Q. A Review on Representative Swarm Intelligence Algorithms for Solving Optimization Problems: Applications and Trends. IEEE/CAA J Autom Sin. 2021 Oct.;8(10): 1627-1643. https://doi.org/10.1109/JAS.2021.1004129.
2. Abualigah L, Yousri D, Elaziz MA, Ewees AA, Al-qaness MA, Gandomi AH. Aquila Optimizer: A novel meta-heuristic optimization Algorithm. Comput Ind Eng. 2021; 157: 1-63. https://doi.org/10.1016/j.cie.2021.107250
3. Wang S, Jia H, Abualigah L, Liu Q, Zheng R. An Improved Hybrid Aquila Optimizer and Harris Hawks Algorithm for Solving Industrial Engineering Optimization Problems. Processes. 2021; 9(9): 1551.1-28. https://doi.org/10.3390/pr9091551.
4. Zauleck JPP. Improving grid-based quantum dynamics: From the inclusion of solvents. [Dissertation]. Faculty of Chemistry and Pharmacy, Ludwig Maximilians University at Munich; 2017; 1-123.
5. Li C, Sun J, Li L-W, Wu X, Palade V. An Effective Swarm Intelligence Optimization Algorithm for Flexible Ligand Docking. IEEE/ACM Trans Comput Biol Bioinform. 2022 May; 19(5): 2672-2684. https://doi.org/10.1109/TCBB.2021.3103777.
6. Norfadzlia MY, Azah KM, Satrya FP. Swarm Intelligence-Based Feature Selection for Amphetamine-Type Stimulants (ATS) Drug 3D Molecular Structure Classification. Appl Artif Intell. 2021; 35(12): 914-932. https://doi:10.1080/08839514.2021.1966882.
7. Gandini E, Marcou G, Bonachera F, Varnek A, Pieraccini S, Sironi M. Molecular Similarity Perception Based on Machine-Learning Models. Int J Mol Sci. 2022; 23(11): 1-2. https://doi.org/10.3390/ijms23116114.
8. Gallegos Saliner A. Molecular Quantum Similarity in QSAR: Applications in Computer-Aided Molecular Design. Doctoral Thesis for the obtaining of the degree of Doctor in Theoretical and Computational Chemistry. Universitat de Girona, Institut de Química Computacional; 2004.
9. Mishra S, Sagban R, Yakoob A, Gandhi N. Swarm intelligence in anomaly detection systems: an overview. Int. J.. Comput. Appl. 2018; 43(2): 109-118. https://doi.org/10.1080/1206212X.2018.1521895.
10. Zhang Y, Xu X, Zhang N, Zhang K, Dong W, Li X. Adaptive Aquila Optimizer Combining Niche Thought with Dispersed Chaotic Swarm. Sensors. 2023; 23(2): 755: 1-24. https://doi.org/10.3390/s23020755.
11. Ammal RA, PC S, SS V. Termite inspired algorithm for traffic engineering in hybrid software defined networks. Peer J Comput Sci. 2020; 6: e283. 1-21. https://doi.org/10.7717/peerj-cs.283
12. Hoang-Le Minh, Thanh Sang-To, Guy Theraulaz, Magd Abdel Wahab, Thanh Cuong-Le. Termite life cycle optimizer. Expert Syst Appl. 2023; 213(Part C): 119211: 1-54. https://doi.org/10.1016/j.eswa.2022.119211.
13. Varun Kumar Ojha, Ajith Abraham, Václav Snášel. Metaheuristic design of feedforward neural networks: A review of two decades of research. Eng Appl Artif Intell. 2017; 60: 98-116. https://doi.org/10.1016/j.engappai.2017.01.013.
14. Hugh Cartwright, Artificial Neural Networks (Methods in Molecular Biology, 2190), Humana press 3rd ed. 2021; 73-297. https://doi.org/10.1007/978-1-0716-0826-5.
15. Moldovan A, Caţaron A, Andonie R. Learning in Feedforward Neural Networks Accelerated by Transfer Entropy. Entropy. 2020; 22: 1-19. https://doi.org/10.3390/e22010102.
16. Szandała T. Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks. arXiv. 2020; 2010: 09458: 1-24.
17. Sharkawy AN. Principle of Neural Network and Its Main Types: Review. J Adv Appl Comput Math. 2020;

7: 8-19. https://doi.org/10.15377/2409-5761.2020.07.2.

18. Shaik NB, Pedapati SR, Taqvi SAA, Othman AR, Dzubir FAA. A Feed-Forward Back Propagation Neural Network Approach to Predict the Life Condition of Crude Oil Pipeline. Processes. 2020; 8: 1-13. https://doi.org/10.3390/pr8060661.
19. Tong Yu, Hong Zhu. Hyper-Parameter Optimization: A Review of Algorithms and Applications. arXiv: 2003.05689. 2020; 1-56. https://doi.org/10.48550/arXiv.2003.05689.
20. Deng S, Yu Y. Predicting Structural Similarity between Molecules Using Graph Neural Networks. In: 2022 10th International Conference on Bioinformatics and Computational Biology (ICBCB); Hangzhou, China; 2022: 78-84. https://doi.org/10.1109/ICBCB55259.2022.9802484.
21. Beck H, Härter M, Haß B, Schmeck C, Baerfacker L. Small molecules and their impact in drug discovery: A perspective on the occasion of the 125th anniversary of the Bayer Chemical Research Laboratory. Drug Discovery Today. 2022; 2(6): 1560-1574. https://doi.org/10.1016/j.drudis.2022.02.015.
22. Li Q, Kang C. Mechanisms of Action for Small Molecules Revealed by Structural Biology in Drug Discovery. Int J Mol Sci. 2020; 21(15): 1-18. https://doi.org/10.3390/ijms21155262.
23. Fu T, Xiao C, Glass LM, Sun J. MOLER: Incorporate Molecule-Level Reward to Enhance Deep Generative Model for Molecule Optimization. IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE). 2021; 34(11): 5459-5471. https://doi.org/10.1109/TKDE.2021.3052150.
24. Hernandez M, Zaribafiyan A, Aramon M, Naghibi M. A Novel Graph-based Approach for Determining Molecular Similarity. 1QB Information Technologies. arXiv. 2016; 1-16. https://doi.org/10.48550/arXiv.1601.06693.
25. Samanta S, O'Hagan S, Swainston N, Roberts TJ, Kell DB. VAE-Sim: A Novel Molecular Similarity Measure Based on a Variational Autoencoder. Molecules. 2020; 25(15): 1-16. https://doi.org/10.3390/molecules25153446.
26. Hiteshi Tandon, Tanmoy Chakraborty, Vandana Suhag. A Concise Review on the Significance of QSAR in Drug Design. Chem Biomol Eng. 2019; 4(4): 45-51. https://doi.org/10.11648/j.cbe.20190404.11.
27. M Faisal, E M Zamzami, Sutarman. Comparative Analysis of Inter-Centroid K-Means Performance using Euclidean Distance, Canberra Distance and Manhattan Distance. J Phys: Conf Ser. 2019.
28. Yongjian Sun, Shaohui Li, Yaling Wang, Xiaohong Wang. Fault diagnosis of rolling bearing based on empirical mode decomposition and improved Manhattan distance in symmetrized dot pattern image. Mech Syst Signal Process. 2021; 159: 107817: 1-23 https://doi.org/10.1016/j.ymssp.2021.107817 .
29. Robin Winter, Floriane Montanari, Andreas Steffen, Hans Briem, Frank Noe, Djork-Arn´e Clevert. Efficient multi-objective molecular optimization in a continuous latent space. Chem Sci. .2019; 10(34): 8016-8024. http://dx.doi.org/10.1039/C9SC01928F
30. Jian Y, He Y, Yang J, Han W, Zhai X, Zhao Y, Li Y. Molecular Modeling Study for the Design of Novel Peroxisome Proliferator-Activated Receptor Gamma Agonists Using 3D-QSAR and Molecular Docking. Int J Mol Sci. 2018; 19(2): 630: 1-15. https://doi.org/10.3390/ijms19020630.
31. Potemkin VA, Grishina AM, Potemkin AV. Grid-based Continual Analysis of Molecular Interior for Drug Discovery, QSAR and QSPR. Curr Drug Discov Technol. 2017; 14(3): 181-205. https://dx.doi.org/10.2174/15701638146661702071 44018.

Published Online First: July, 2024
https://doi.org/10.21123/bsj.2024.9278
P-ISSN: 2078-8665 - E-ISSN: 2411-7986

Baghdad Science Journal

نظام متكامل لذكاء السرب والشبكة العصبية لاكتشاف التشابه الجزيئي

فادية سامي جاسم[1] ، هاكان كوينجو[2]

[1]قسم هندسة الحاسوب، تكنولوجيا المعلومات، جامعة ألتنباش، إسطنبول، تركيا.
[2]قسم هندسة الحاسوب، كلية الهندسة والعمارة، جامعة ألتنباش، إسطنبول، تركيا.

الخلاصة

التشابه الجزيئي مفهوم واسع وله تشعبات عديدة عبر العديد من الجوانب الكيميائية حيث ان الفكرة السائدة كيمائياً بان الجزيئات متشابه المظهر لها خصاص متشابهة في التأثير. ولهذا السبب، فإن تقنيات حساب التشابه الجزيئي لها فوائد متنوعة في صناعة المستحضرات الصيدلانية، كما هو الحال في سياق اتصالات الهيكل والنشاط للجزيئات. ان الهدف من التشابه الجزيئي هو تحديد جزيئات العينة التي تشبه الجزيء المستهدف بعد تحريكها وتدويرها بطرق الذكاء الاصطناعي. يعد ذكاء السرب والشبكات العصبية جزء من الذكاء الاصطناعي اللتان تم استخدامهما على نطاق واسع في مجموعة متنوعة من التطبيقات الكيميائية. في هذا البحث تم استخدام نظام هجين من ذكاء السرب حيث تم الدمج بين سلوكيات النمل الابيض وطائر العقاب للعثور على جزيء العينة الأكثر تشابهًا في مجموعة البيانات وفقًا لجزيء مستهدف واحد. ان الطرق العديدة للتحقق من دقة التشابه تعتبر طرق نسبية وبذلك قد تم استخدام طريقة تقليدية للتحقق من دقة النتائج وقد حصلنا على نسبة 70.58% ولكن بعد استخدام الشبكات العصبة قد ازدادت الدقة الى 90%. تم استخدام لوح شبكي ثلاثي الابعاد لا يجاد العلاقة الكمية بين جزيئات العينة مقارنة بالجزيء المستهدف حيث تم تطبيق معادلة المسافة الاقليدية والمانهاتن للحصول التشابه والاختلاف فيما بينهم.

الكلمات المفتاحية: التشابه الجزيئي, ذكاء السرب, طائر العقاب, النمل الابيض, الشبكات العصبية.