Comparative Analysis of Drug Compound Representation and perceived Reactivity: Chemception Embedding vs Chemical Fingerprinting

Swapnil Jha, Milena Voskanyan
Project Module SoSe2024, Universität Potsdam
Karl-Liebknecht-Straße 24-25, 14476 Potsdam, Germany
{swapnil.jha, voskanyan}@uni-potsdam.de

Abstract

This study presents a comparative analysis of drug compound representations and their impact on the predictive modeling of reactivity with cancer cells. Two primary approaches are explored and compared: Chemception Embedding[1], which uses convolutional neural networks (CNNs[2]) on images generated from SMILES codes[3], and traditional chemical fingerprinting techniques[4]. The study evaluates the performance of these two approaches to drug compound representation by training models to predict chemical reactivity. The goal is to assess which representation method best captures the essential features of the compounds and offers the best predictive performance. Validation metrics such as Mean Absolute Error (MAE), Root Mean Square Error ($RMSE$), and Coefficient of Determination (R^2) are utilized for the performance assessment. This analysis provides insights into the advantages and limitations of Chemception embedding compared to traditional fingerprint-based methods.

1 Introduction

Predicting chemical reactivity and sensitivity to drug compounds is a critical aspect of drug discovery and development. It requires representing chemical compounds in a form that enables machine learning models to effectively capture their structural and functional properties. Traditionally, chemical fingerprints—numerical representations of molecular structures—have been widely used for this purpose[5]. For instance, MACCS fingerprints encode the presence or absence of substructures in a molecule[6], offering a straightforward and interpretable method for chem-informatics tasks. However, using a single chemical fingerprint has its limitations, particularly in its ability to generalize across diverse chemical spaces and capture subtle, high-level structural features.

In recent years, deep learning techniques have gained significant traction in the field of cheminformatics, particularly with the advent of methods like Chemception. Chemception[1] is a CNN-based model that utilizes image representations of chemical compounds generated from SMILES (Simplified Molecular Input Line Entry System) codes[3]. SMILES is a textual format representing chemical structures that can be converted into 2D molecular images. By leveraging the power of CNNs, Chemception aims to automatically extract meaningful features—i.e., relevant chemical properties—from these images, offering a potentially richer and more flexible representation compared to traditional fingerprints[3].

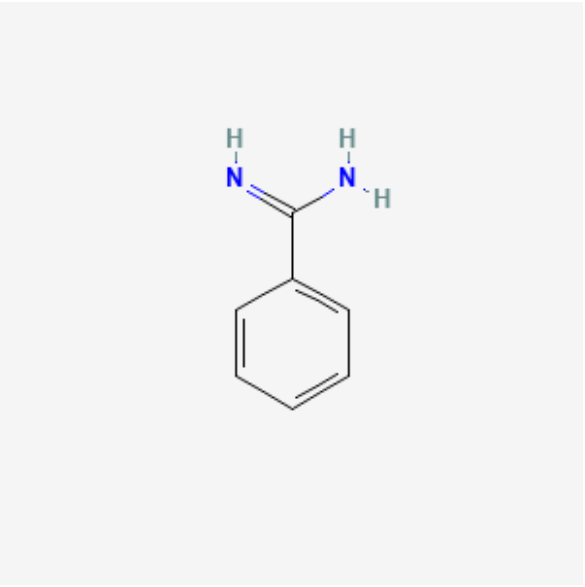

Figure 1: Visual Representation of drug Benzamidine SMILES Code-$C1 = CC = C(C = C1)C(= N)N$[7]

This report explores the effectiveness of these two distinct approaches. Our study is based on the dataset provided by Genomics of Drug Sensitivity in Cancer[8]. The remainder of the report details the methodology used to generate the SMILES codes,

create corresponding chemical objects and molecular images, and train both the Chemception-based CNN model and the fingerprint-based neural network architectures. We further evaluate and compare the performance of these models, discussing their strengths and limitations. Finally, we provide recommendations for future research directions and potential improvements in modeling chemical reactivity.

2 Previous work

In recent years, the intersection of research in deep learning and chem-informatics has grown significantly, leading to advancements in the representation and prediction of chemicals and their properties. Several studies have explored different techniques for encoding molecular structures, ranging from traditional chemical fingerprinting to more sophisticated image-based representations using neural networks, making them readable by machine learning algorithms. This study builds upon these existing works, focusing on the comparative performance of Chemception embedding and Morgan[9] and MACCS fingerprints in predicting chemical reactivity.

2.1 Traditional Chemical Fingerprinting Methods

Chemical fingerprints, particularly **Extended-Connectivity Fingerprints (ECFPs)**, have long been used in quantitative structure-activity relationship (QSAR) models[10] to predict molecular properties, including chemical reactivity. Rogers and Hahn (2010)[9] introduced ECFPs, which encode molecular substructures as bit vectors, effectively capturing atom-level connectivity crucial for structure-based predictions.

In the context of predicting chemical reactivity in organic materials, Lee and Kang (2020)[11] compared Morgan fingerprints and **MACCS keys**. Their findings demonstrated the strengths of both methods: Morgan fingerprints provided detailed structural insights, while MACCS keys, consisting of predefined structural features, offered a more straightforward chemical depiction.

2.2 Deep Learning and Chemception

The Chemception model, developed by Goh et al. (2017)[12] utilizes convolutional neural networks based on Google's Inception Convolutional Neural Network[13] to analyze molecular images generated from SMILES codes. This approach enables the model to automatically learn molecular descriptors, thereby enhancing predictive accuracy.

Following this Goh et al. (2018)[14] further emphasized the potential of Chemception for drug property prediction, establishing that CNNs could effectively extract complex features from chemical structures. The framework is influenced by earlier works on CNNs for image recognition, particularly the architecture proposed by Simonyan and Zisserman (2015)[15], which has inspired applications across various domains, including chemistry.

2.3 Comparative Analysis

While the image-based approach of Chemception has advantages, such as enabling the learning of intricate molecular features, traditional fingerprinting methods like Morgan and MACCS keys maintain their valuable positions due to their established interpretability and performance. This study builds on these foundational works by comparing these two approaches on a common dataset, aiming to determine the most effective representation for capturing essential chemical features to predict chemical reactivity in cancer cells.

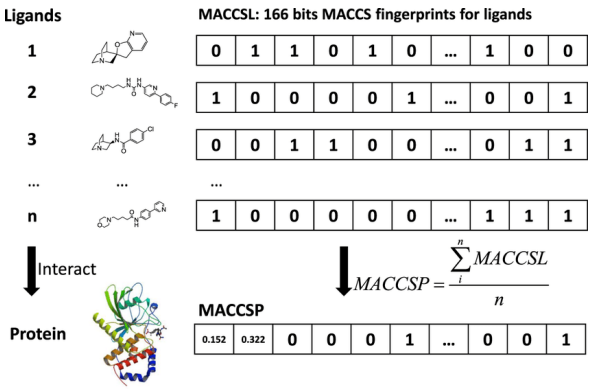

Figure 2: The scheme of generating MACCS keys. Note that the numbers are artificial, not real numbers.

3 Data

3.1 Data Acquisition and Understanding

The dataset for our study was acquired from the Genomics of Drug Sensitivity in Cancer[8], specifically the **GDSC2 Fitted Dose Response** dataset from *GDSC Release 8.5*[16]. The Genomics of Drug Sensitivity in Cancer (GDSC) database is the largest publicly accessible resource providing information on cancer cell drug sensitivity and molecular markers related to drug response. The data is freely available without restrictions.

The dataset comprises 19 columns, but the accompanying information does not fully describe all of them. The available dataset documentation, sourced from the Genomics of Drug Sensitivity in Cancer[18], was last updated on September 21, 2017. Due to its outdated nature, it lacks details on several columns within the dataset.

The NLME_CURVE _ID column was excluded after an investigation suggested it might be associated with an R package, as referenced in the R package documentation[19]. Since its relevance to the current dataset remains unclear, this feature has been omitted from the training data.

Despite extensive efforts, we were unable to locate the relevant COSMIC IDs on the official COSMIC dataset website. Based on this, we concluded that incorporating this feature is not feasible within the scope of this project.

We analyzed the following three columns: Cosmic_ID, SANGER_MODEL_ID and CELL _LINE_NAME, and identified a relationship between them. Since both Cosmic_ID and SANGER_MODEL_ID are identification numbers, we opted to retain CELL_LINE_NAME out of the three.

Upon further examination of the columns, we found several where the majority of the values were 'None' or 'Unclassified'. We then assessed the information loss from removing these columns, which was determined to be only 0.84%. As a result, these columns were also removed.

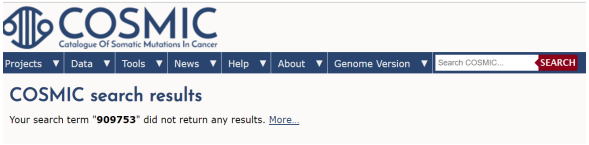

Figure 3: An example outcome when searching for COSMIC_ID in the COSMIC database

3.2 Dataset Preprocessing

After an extensive study and thorough analysis of the given data, four input parameters were finalized: CELL_LINE_NAME, DRUG_NAME, MIN_CONC and MAX_CONC to predict the reactivity column LN_IC50.

- The Minimum Concentration and Maximum Concentration columns were log-transformed, as some values were extremely small and close to zero, while others were quite large. The NumPy library was used for this task.
- One-hot encoding was applied to the Cell Line Names since it was a categorical variable.
- Out of the 286 unique drug names, 255 SMILES codes were generated using the PubChemPy library, 27 were manually annotated, and 31 were classified as drugs in trials with no available SMILES code information online. These 31 drugs were subsequently removed from the dataset.
- SMILES codes were processed differently depending on the model approach.

3.3 Dataset Preprocessing - Chempception

For the Chemception model, SMILES codes were converted into RDKit chemical objects, then further transformed into molecule image vectors using Chemception embeddings, and finally into a set of 2D images. The RDKit library functions were used for these transformations.

3.3.1 Flaws in Chemception Embedding

For two specific drugs, *Cisplatin* and *123829*, the RDKit chemical object failed to convert into Chemception embedding-based image vectors. This failure is likely attributed to the presence of polar covalent bonds in these compounds, where the electrons forming the bond are unequally distributed. As a result, these two drugs were also removed from the dataset.

3.3.2 Challenges in the Reproducibility of Chemception Performance

In addition to the failure to convert all the drugs into Chemception-based vector image encodings, the reproducibility test of an existing study using a different dataset of IC50[17] also performed worse than claimed. The original research reported an R^2 score of 0.71; however, only an R^2 score of 0.48 was achieved with the original architecture, code, and dataset when run on a Google Colab TPUv2-8 system.

3.4 Dataset Preprocessing - Morgan and MACCS

For the fingerprint model, SMILES codes were used to generate fingerprints with the help of the RDKit library. The two types of fingerprints chosen for this predictive analysis were Morgan fingerprints and MACCS fingerprints.

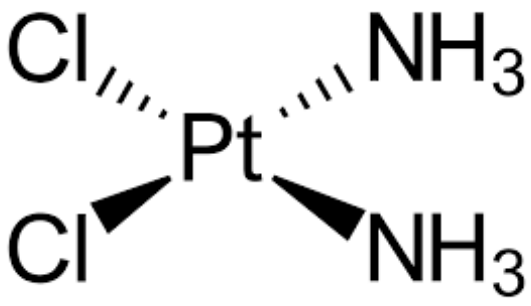

Figure 4: Visual Representation of drug Cisplastin SMILES Code-$N.N.Cl[Pt]Cl$

3.4.1 Morgan Fingerprints

Morgan fingerprints, also known as circular fingerprints or Extended Connectivity Fingerprints (ECFP), are generated from local atom environments, capturing atom connectivity and bonding patterns within a defined radius.

- **Radius** -The radius determines the depth of the included environment, making it highly customizable based on the requirements of the analysis. A *radius* of 2 was chosen for the current study.
- **Bits** - Morgan fingerprints can have a bit length specified as 512, 1024, 2048, etc., depending on the level of detail required. A bit length of 2048 was chosen for the current study.

These fingerprints are particularly useful in data-driven tasks, such as virtual screening or training machine learning algorithms, due to their ability to represent complex molecular environments in a scalable manner. The scalability and adaptability of these fingerprints make them a preferred choice for handling large datasets, especially in drug discovery and molecular similarity searches.

3.4.2 Molecular ACCess System Keys

MACCS fingerprints, or MACCS keys, are predefined substructure-based molecular descriptors commonly used in chem-informatics. These fingerprints are generated from 166 predefined substructures, with each bit representing the presence of a specific substructure. The RDKit library returns a 167-bit MACCS key, where the number of bits corresponds to their positions, and bit 0 is always set to 0.

MACCS fingerprints are simple and easy to interpret, making them ideal for tasks that require pattern matching in substructure searches. Their standardized and intuitive design enables efficient analysis in applications where the rapid identification of chemical properties is critical.

4 Methodology

The methodology for this project involved two distinct model architectures: a convolutional neural network (CNN) for Chemception-based molecular representations and a multilayer perceptron (MLP) architecture for molecular fingerprint representations. These approaches are described in detail in the following subsections.

The dataset in both cases was divided into a 90%-10% split to create a training set and a test set. The training set was further divided into an 80%-20% split to generate training and validation sets, resulting in a final distribution of 72%-18%-10% for training, validation, and test sets. This was achieved using the train-test split function from the scikit-learn library.

4.1 Chemception

The original Chemception model only accepts images as input, which was insufficient for our purposes. Therefore, additional input and concatenate layers were added to incorporate other features, such as minimum concentration, maximum concentration, and cell line name.

The RDKit object was converted into a Chemception-compatible format at 80×80 pixels for effective feature extraction.

4.1.1 Model Architecture

The entire codebase for this study is open-source and is available on our GitHub[20].

The proposed model is a multi-input architecture designed to process molecular image data and numerical data simultaneously. It integrates both image features extracted via Inception layers and numerical features through fully connected layers. The architecture is detailed as follows.

1. **Input Layers**:
    - **Image Input**: The image input tensor has a shape of $X_{img} \in R^{H \times W \times C}$, where H is the height, W is the width, and C is the number of channels of the molecular image. In this case, the input shape corresponds to the dimensions of the molecular image dataset.

- **Feature Input**: The second input tensor $X_{feat} \in R^F$ represents the numerical features where F is the total number of features, including two chemical concentration values (minimum concentration and maximum concentration) and one-hot encoded cell line name representations.

2. **Inception Blocks**:

   - The architecture uses Inception layers to extract features from the image input. The Inception0 block and two subsequent inception blocks follow the same overall structure.
     $\mathbf{T_1} = \text{Conv2D}(16, (1,1)) \rightarrow \text{ReLU} \rightarrow \text{Conv2D}(16, (3,3)) \rightarrow \text{ReLU}$
     $\mathbf{T_2} = \text{Conv2D}(16, (1,1)) \rightarrow \text{ReLU} \rightarrow \text{Conv2D}(16, (5,5)) \rightarrow \text{ReLU}$
     $\mathbf{T_3} = \text{Conv2D}(16, (1,1)) \rightarrow \text{ReLU}$ (Inception0 block)
   - In Inception blocks following Inception0, an additional layer of MaxPooling2D is added.
     $\mathbf{T_3} = \text{MaxPooling2D}((3,3), \text{stride} = (1,1)) \rightarrow \text{Conv2D}(16, (1,1)) \rightarrow \text{ReLU}$
   - The outputs of the towers T_1, T_2, and T_3 are concatenated along the depth, yielding an output tensor $X_{concat} \in R^{H \times W \times 48}$

3. **Pooling and Flattening Layers**:

   - After the inception blocks, a MaxPooling2D layer is applied globally with a pool size equal to the spatial dimensions of the feature maps, reducing the image features to a tensor of $X_{pool} \in R^{1 \times 1 \times 48}$
   - The pool tensor is then flattened into a vector $X_{flat} \in R^{48}$.

4. **Feature Concatenation:**

   The flattened image vector $X_{flat} \in R^{48}$ is concatenated with the numerical feature $X_{feat} \in R^F$, resulting in a feature vector $X_{concat-final} \in R^{48+F}$ where F represents the total number of numerical features, i.e. 971 (=969 unique cell line names + min. conc. + max. conc.)

5. **Fully Connected Layer:**

   The concatenated vector is passed through a fully connected (dense) layer with 100 hidden units:

   $$X_{dense} = \text{Dense}(100) \rightarrow \text{ReLU}$$

   This transformation generates the latent representation of features $X_{dense} \in R^{100}$

6. **Output Layer:**

   The final layer is a single unit dense layer with a linear activation function producing the scalar output necessary for a regression task:

   $$Y_{pred} = \text{Dense}(1) \rightarrow \text{Linear}$$

   This transformation generates the predicted value $Y_{pred} \in R$, corresponding to the *LN_IC50*.

4.1.2 Image Data Augmentation

Image data augmentation was employed to enhance the diversity of the image dataset by applying random transformations to the molecular images. The augmentations include, but are not limited to, random rotations of up to 180 degrees, horizontal and vertical shifting of up to 10% of the image, and horizontal and vertical image flipping. Newly generated pixels were filled with a constant value of 0 (i.e., black). This process was implemented through a custom *DataGenerator* class, which allowed for the augmentation of image data while maintaining the integrity of other features. The aim of this process is to improve the generalization of the model by training it with varied versions of the input data, simulating different molecular orientations and noise while preserving the original structural properties of the molecules.

4.1.3 Architecture Summary

$[X_{\text{img}}, X_{\text{feat}}] \rightarrow \text{Inception0} \rightarrow \text{Inception} \rightarrow \text{Inception} \rightarrow \text{MaxPooling2D} \rightarrow \text{Flatten} \rightarrow \text{Concatenate} \rightarrow \text{Dense}(100) \rightarrow \text{Dense}(1)$

4.2 Morgan and MACCS

Only Morgan and MACCS fingerprints were used for our modeling because they represent different aspects of the molecule. Using additional types of fingerprints, such as torsional fingerprints, could introduce overlapping information with the Morgan fingerprints, leading to collinearity in the training features.

4.2.1 Model Architecture

The proposed model is a multi-input architecture designed to process molecular fingerprint vectors (Morgan and MACCS) along with chemical concentrations and cell line name features. The architecture integrates these distinct inputs using dense layers and concatenation layers. The following are the details of the architecture.

1. **Input Layers**:
   - **Morgan Fingerprints Input**: The first input tensor has a shape of $X_{Morgan} \in R^{d_1}$, where d_1 is the the bits of the Morgan fingerprints, i.e. 2048.
   - **MACCS Fingerprints Input**: The second input tensor is shaped $X_{MACCS} \in R^{d_2}$, where d_2 represents the fixed width of 167 (166 bits and a 0 bit by RDKit Library).
   - **Feature Input**: The third input tensor is shaped $X_{feat} \in R^F$, where F is the number of features (i.e. chemical concentrations, and one-hot encoded cell line names data).
2. **Dense Layers for Fingerprints**:
   - **Morgan Fingerprints Layers**: The Morgan fingerprints are passed through two fully connected (Dense) layers:
     $X_1 = \text{Dense}(64) \rightarrow \text{ReLU} \rightarrow \text{Dense}(32) \rightarrow \text{ReLU}$
     This results in a latent representation of Morgan fingerprints, $X_1 \in R^{32}$.
   - **MACCS Fingerprints Input**: Similarly, the MACCS fingerprints are also passed through two fully connected (Dense) layers:
     $X_2 = \text{Dense}(64) \rightarrow \text{ReLU} \rightarrow \text{Dense}(32) \rightarrow \text{ReLU}$
     This results in a latent representation of MACCS fingerprints, $X_2 \in R^{32}$.
3. **Feature Concatenation:**
   The outputs from both the fingerprint pathways, along with feature input, $X_{feat} \in R^F$ are concatenated.
   $X_{concat} = \text{Concatenate}([X_1, X_2, X_{feat}])$
   This results in a latent representation of $X_{concat} \in R^{1035}$. (32 from Morgan + 32 from MACCS + rest from features)
4. **Fully Connected Layer:**
   The concatenated vector is passed through a fully connected layer (dense layer) with 100 hidden units:
   $X_{dense} = \text{Dense}(100) \rightarrow \text{ReLU}$
   This transformation generates the latent representation of features $X_{dense} \in R^{100}$
5. **Output Layer:**
   The final layer is a single unit dense layer with a linear activation function producing the scalar output necessary for a regression task:
   $Y_{pred} = \text{Dense}(1) \rightarrow \text{Linear}$
   This transformation generates the predicted value $Y_{pred} \in R$, corresponding to the *LN_IC50*.

4.2.2 Architecture Summary

$[X_{Morgan}, X_{MACCS}, X_{feat}] \rightarrow \text{Dense}(64) \rightarrow \text{ReLU} \rightarrow \text{Dense}(32) \rightarrow \text{ReLU} \rightarrow \text{Concatenate} \rightarrow \text{Dense}(100) \rightarrow \text{ReLU} \rightarrow \text{Dense}(1)$

4.3 Training Procedure

- **Loss Function**: Mean Squared Error (MSE) was chosen as the loss function.
- **Optimizer**: Adam optimizer with an initial learning rate of 25×10^{-6} was used.
- **Epochs and Batch Size**: The model was trained for 300 epochs with varying batch sizes.
- **Callbacks**: Early stopping was implemented with a patience of 20 epochs, along with a reduction in the learning rate upon plateauing of the validation loss, which also had a patience of 5 epochs. Additionally, the best weights were saved to prevent unnecessary training.
- **ReduceLROnPlateau**: The minimum learning rate was set to 1^{-15}.

5 Results

Several key metrics were utilized to evaluate the performance of the two models: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and the Coefficient of Determination (R-Squared). The results reveal clear differences in predictive

accuracy and overall effectiveness between the two approaches.

Notably, across both models, using a batch size of 128 yields slightly better performance compared to a batch size of 64, particularly in terms of R-Squared and RMSE. This observation suggests that larger batch sizes enhance the learning process, potentially by providing more stable gradient estimates during training.

Focusing first on the Chemception model, while its performance is adequate, it does fall short in terms of precision. The RMSE values indicate that the average squared difference between actual and predicted outcomes is relatively high, suggesting significant deviations from true values. Similarly, the MAE corroborates this finding, as it shows that the average error in predictions is approximately 1.43 units.

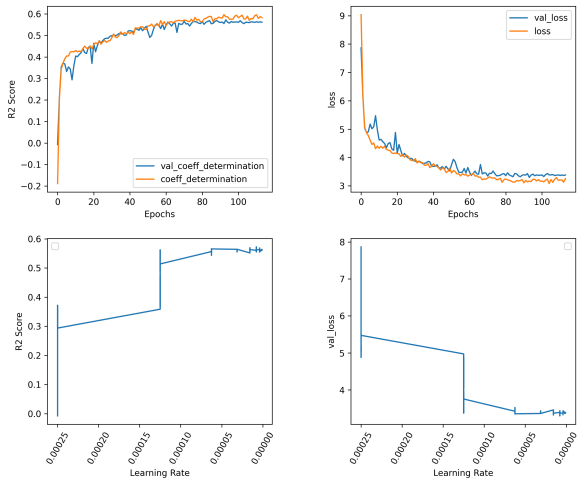

Figure 5: Model Training of Chemception based model depicting increasing R^2 and decreasing validation loss.

Moreover, the MAPE values indicate that the predictions of the Chemception model, on average, deviate by around 2%, which is moderate but not ideal for high-precision tasks. The R-Squared value suggests that the model explains approximately 58% of the variability in the data, leaving a considerable portion of the variance unexplained. This implies that the Chemception model may not fully capture the intricate relationships between molecular structure and cell sensitivity.

In contrast, the Morgan and MACCS fingerprints-based model demonstrates significantly superior performance across all metrics.

The RMSE of 0.99 illustrates that the squared prediction errors are nearly half those of the Chemception model, indicating a marked improvement

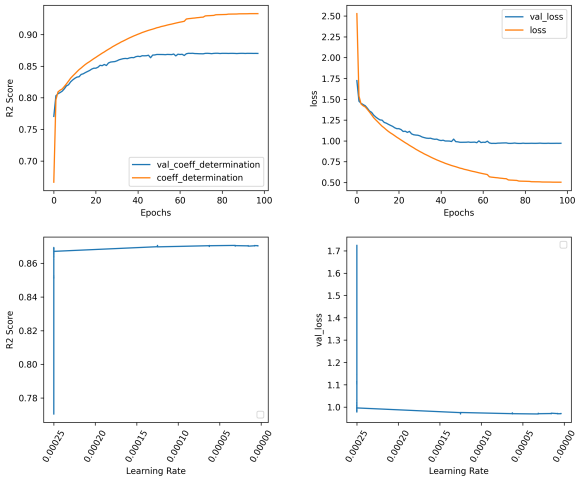

Figure 6: Model Training of Fingerprint based model depicting increasing R^2 and decreasing validation loss.

in accuracy. Additionally, the MAE of 0.74 further reinforces this point, showing that the average absolute prediction error is much lower, meaning the predictions of the model are considerably closer to the true values.

Model	R2	RMSE	MAE	MAPE
Chemception(128)	0.58	1.81	1.43	2.27%
Chemception(64)	0.56	1.96	1.42	2.45%
Fingerprint(128)	**0.88**	**0.99**	**0.74**	**0.80%**
Fingerprint(64)	0.87	1.01	0.78	0.81%

Table 1: Validation metrics with hold-out validation method. Note that the number in bracket represents the batch size.

Furthermore, the MAPE value of 0.80% reflects highly accurate predictions, with errors under 1%—a significant improvement over the Chemception model. The standout R-Squared value of 0.88 demonstrates that the fingerprint model explains around 88% of the variance in the data. This highlights its ability to capture the underlying relationships between molecular descriptors and cell sensitivity, suggesting that the Morgan and MACCS fingerprints provide a rich and detailed representation of molecular structures that is highly informative for the prediction task.

6 Significance Testing

When evaluating the best models' predicted values with true values, there are noticeable differences in the significance testing results.

The Chemception model produced a T-statistic of -18.66 and an extremely small p-value of 4.47e-77. Since this p-value is well below the typical

threshold of 0.05, it indicates that the difference between the predicted and true values is highly unlikely to have occurred by chance or noise in data. The negative T-statistic suggests that the Chemception model consistently underestimates the true values. Although the predictions are statistically significant, they tend to be systematically lower than expected. The large T-statistic further emphasizes the extent of this underestimation.

In contrast, the fingerprint-based model yielded a T-statistic of 1.60 and a p-value of 0.1107. Because the p-value exceeds 0.05, the difference between its predictions and the actual values is not considered statistically significant. This suggests that any observed differences between true and predicted values could be attributed to random variation. The much smaller T-statistic confirms that the gap between its predictions and the true values is minor and not statistically significant.

Model	T-statistic	P-value
Chemception(128)	-18.66	0.00
Fingerprint(128)	**1.60**	**0.11**

Table 2: Comparison of significance testing of true and predicted values of LN_IC50.

7 Conclusion

This study presents a comprehensive comparative analysis of drug compound representations and their impact on predictive modeling of reactivity with cancer cells. By exploring two primary approaches—Chemception Embedding, which leverages convolutional neural networks (CNNs) applied to images generated from SMILES codes, and traditional chemical fingerprinting techniques—this research assesses the effectiveness of each representation method in capturing the essential features of drug compounds.

The study used a standardized dataset and exact same input features, and the model was trained on a T4-v2 TPU system with exact same configurations, contributing to the study design with the least number of external variables, signifying the importance and validity of the claims made by this study. The results clearly demonstrate that the Morgan and MACCS fingerprints-based model outperform the Chemception model across all the validation metrics. The fingerprint model effectively accounts for a significant amount of the variance in drug reactivity data, whereas the Chemception model falls short in this regard. The better predictive accuracy of the fingerprint models, alongside their lower error rates, suggests that traditional chemical fingerprinting techniques are still quite more effective for modeling drug reactivity with cancer cells than the more modern and complex Chemception approach.

This analysis not only highlights the advantages of traditional fingerprint methods in terms of predictive performance but also underscores the limitations of Chemception embedding, particularly its challenges in capturing intricate molecular relationships especially for molecules containing polar covalent bonds.

8 Acknowledgment

Within the scope of this project, the contributions of each team member are as follows:

Swapnil Jha: Data understanding, Reproducibility of Chemception network on IGC50 dataset, adapting Chemception to incorporate more training features for improved model performance, model training, debugging initial failure of chemception with infinite loss (inability to generate vector for 2 drugs), hyperparameter tuning.

Milena Voskanyan: Data understanding, data preprocessing, manual annotation of drugs, study on types of fingerprints, model architecture and training for fingerprints, and overall significance testing.

References

[1] Duvenaud, D. K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., & Adams, R. P. (2015). Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems, 28*.

[2] Purwono, P., Ma'arif, A., Rahmaniar, W., Fathurrahman, H., Frisky, A., & Haq, Q. (2023). Understanding of Convolutional Neural Network (CNN): A Review. *International Journal of Robotics and Control Systems, 2(4), 739-748.*

[3] Weininger, D. (1988). SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of chemical information and computer sciences, 28(1)*, 31-36.

[4] Douglas, G. S., Emsbo-Mattingly, S. D., Stout, S. A., Uhler, A. D., & McCarthy, K. J. (2007). Chemical fingerprinting methods. *Introduction to environmental forensics, 2.*

[5] Yang, J., Cai, Y., Zhao, K., Xie, H., & Chen, X. (2022). Concepts and applications of chemical fin-

gerprint for hit and lead screening. *Drug Discovery Today*, 27(11), 103356.

[6] Gao, K., Nguyen, D. D., Sresht, V., Mathiowetz, A. M., Tu, M., & Wei, G. W. (2020). Are 2D fingerprints still valuable for drug discovery?. *Physical chemistry chemical physics, 22(16)*, 8373-8390.

[7] PubChem, https://pubchem.ncbi.nlm.nih.gov/compound/2332. Last accessed 4 Oct 2024.

[8] Genomics of Drug Sensitivity in Cancer, https://www.cancerrxgene.org/downloads/bulk_download. Last accessed 4 Oct 2024.

[9] Rogers, D., & Hahn, M. (2010). Extended-connectivity fingerprints. *Journal of chemical information and modeling, 50*(5), 742-754.

[10] Ding, Y., Chen, M., Guo, C., Zhang, P., & Wang, J. (2021). Molecular fingerprint-based machine learning assisted QSAR model development for prediction of ionic liquid properties. *Journal of Molecular Liquids, 326*, 115212..

[11] Lee, B., Yoo, J., & Kang, K. (2020). Predicting the chemical reactivity of organic materials using a machine-learning approach. *Chemical science, 11*(30), 7813-7822.

[12] Goh, G. B., Siegel, C., Vishnu, A., Hodas, N. O., & Baker, N. (2017). Chemception: a deep neural network with minimal chemistry knowledge matches the performance of expert-developed QSAR/QSPR models. *arXiv preprint arXiv:1706.06689*.

[13] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan D., Vanhoucke V., & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

[14] Goh, G. B., Siegel, C., Vishnu, A., Hodas, N., & Baker, N. (2018, March). How much chemistry does a deep neural network need to know to make accurate predictions?. In 2018 IEEE Winter Conference on Applications of Computer Vision (WACV) (pp. 1340-1349). IEEE.

[15] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[16] GDSC Release 8.5, https://cog.sanger.ac.uk/cancerrxgene/GDSC_release8.5/GDSC2_fitted_dose_response_27Oct23.xlsx. Last accessed 4 Oct 2024.

[17] Chemception on IGC50, https://github.com/Abdulk084/Chemception/blob/master/chemcemption.ipynb. Last accessed 4 Oct 2024.

[18] GDSC Fitted Data Description, https://cog.sanger.ac.uk/cancerrxgene/GDSC_release8.5/GDSC_Fitted_Data_Description.pdf. Last accessed 4 Oct 2024.

[19] NLME R Package Documentation, https://cran.r-project.org/web/packages/nlme/nlme.pdf. Last accessed 4 Oct 2024.

[20] PrecisionMedicine GitHub Repository, https://github.com/swapniljha001/PrecisionMedicine/. Last accessed 10 Oct 2024.