Part 2

Project: Explainable AI-Driven Adverse Drug Reactions Prediction Toward Pediatric Drug Discovery & Development

All source:https://drive.google.com/drive/u/0/folders/1tgMaF3fOFp_wqFd7rTWBiovmlc-8i08_

Objective

To develop a deep learning model that classifies compounds based on their anticancer activity using integrated chemical and biological data. The project involves building **binary classifiers** using real-world datasets of drug compounds, gene expression, and cell line responses, with SMILES strings and molecular fingerprints representing chemical structures, and expression profiles capturing cellular context. The aim is to explore multi-modal deep learning approaches for predicting drug sensitivity, contributing to precision oncology and pediatric drug discovery.

Part - A2: Model Development & Training

First, I Develop and evaluate on baseline models: Fingerprint MLP, **SMILES CNN**, and **Multi-Modal(using multiple types of data)** then do the following tasks:

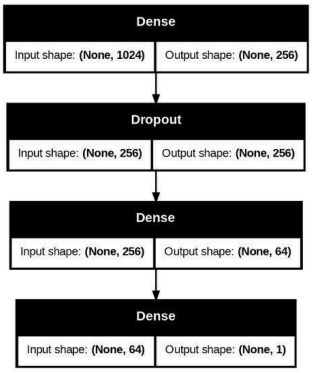
1. **Select the best-performing baseline** based on validation AUC, accuracy, and loss.
2. **Tune hyperparameters** (e.g., learning rate, dropout, architecture) for optimal performance.
3. **Compare** the final model against existing methods, including transfer learning approaches.

• Model Architecture:

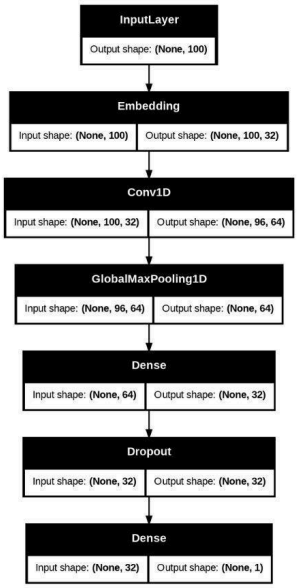
Due to suboptimal performance in preliminary testing, **molecular graph representations were excluded from the final model designs**. Then I compare three architectures for modeling molecular properties and predicting drug sensitivity. The SMILES CNN captures sequential patterns in tokenized SMILES strings using convolutional layers, while the Fingerprint MLP learns interactions within high-dimensional binary fingerprints through fully connected layers. Recognizing that these representations are complementary sequential syntax versus structural subgraph presence, I also propose a Multi-Modal Fusion model that integrates both inputs in a dual-branch architecture. **This approach aims to enhance generalization and robustness by leveraging the strengths of each modality, as evaluated under a consistent drug response classification task.** So three baseline architectures were developed to evaluate different molecular representation approaches:

Baseline Architecture

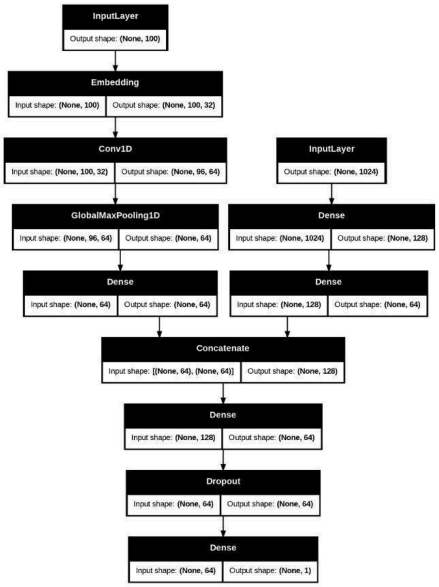
Fingerprints Architecture (MLP)



SMILES Architecture (1D CNN)(tokenizing)



Multi-Modal Architecture (Fingerprints + SMILES)



Model	Input type	Architecture	Key layers	Output
SMILES CNN model	Tokenized SMILES strings	CNN for sequential molecular representation	- Embedding (vocab=40, dim=32) - 1D Conv (64 filters, kernel=5, ReLU) - Global Max Pooling - Dense (32 units) - Dropout (0.3)	Sigmoid (binary classification)
Fingerprint MLP model	1024-bit fingerprints	Feedforward fully connected	- Dense (256 -> 64 -> 1) - ReLU activations - Dropout (0.3)	Sigmoid (binary classification)
Multi-modal Fusion model	SMILES + Fingerprints	Dual-branch architecture combining CNN and MLP features	- SMILES branch: same as SMILES CNN - FP branch: Dense (128 → 64) - Feature concatenation - Dense (64 units) - Dropout	Sigmoid (binary classification)

Training Implementation:

The setup ensures fairness across the three baseline model types: **Fingerprint - MLP**, **SMILES - CNN**, and **Multi-Modal (Fingerprints + SMILES)** by maintaining consistent training configurations. All models are trained using the **binary crossentropy loss**, **Adam optimizer** (learning rate = 0.001), and evaluated using **accuracy** and **AUC** metrics. Early stopping is applied based on validation AUC with a **patience of 10**, and model checkpoints save the best-performing weights. Training is capped at **50 epochs** with a **batch size of 64**, and a **35% validation split** is used via stratified splitting to preserve class balance. The data is processed in exactly the same way for all models by using equal sample sizes, running strict checks, and consistently tokenizing and padding SMILES strings. This uniform approach means that any performance differences are due to the model architectures themselves, not how the data was prepared.

Overfitting Mitigation:

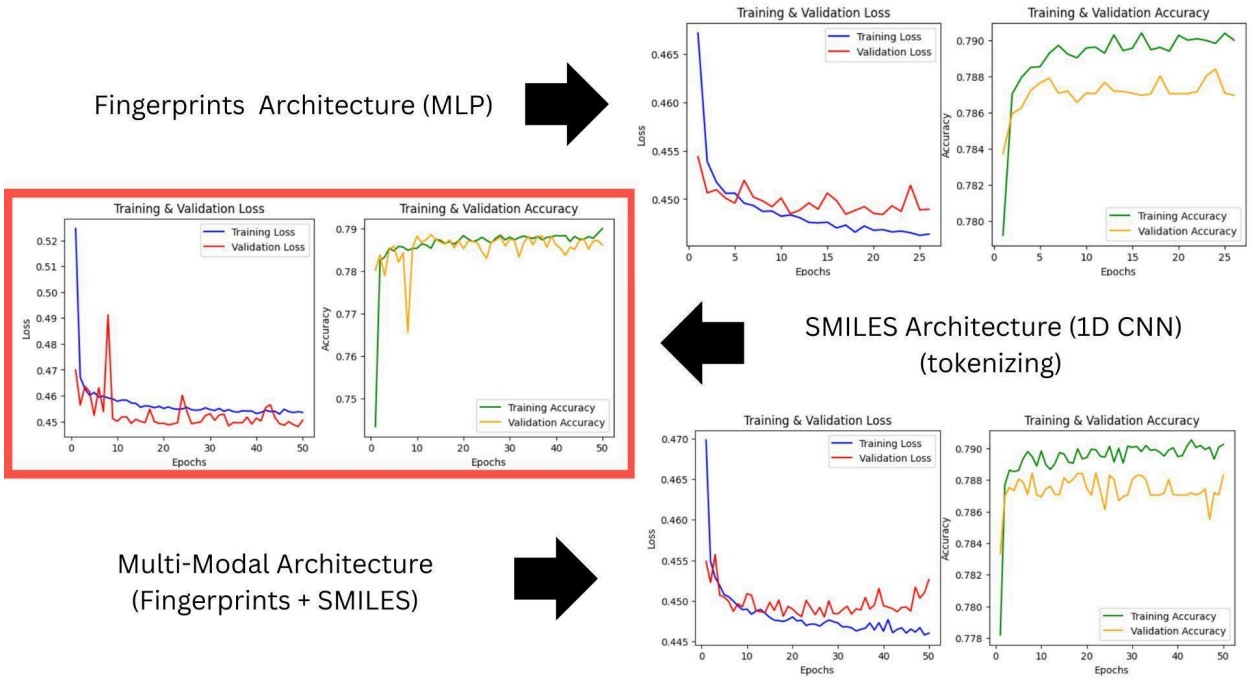
Overfitting is mitigated primarily through **early stopping**, which halts training once the validation AUC stops improving, and **model checkpointing** to retain only the best-performing weights. Additionally, architectural techniques such as **dropout layers** (applied between dense layers in the MLP and CNN) and **regularization (e.g., L2 weight decay)** can be employed to prevent over-reliance on specific features. Stratified splitting further supports generalization by preserving class distributions in both training and validation sets, ensuring representative learning.

Model Evaluation:

Key performance metrics were reported across three baseline architectures: **Fingerprint**, **SMILES**, and **Multi-Modal**. Using the validation/test set. Each model was evaluated using standard classification metrics, including **loss**, **accuracy**, and **area under the ROC (AURUC)**. The results are as follows:

Model	Accuracy (ACC)	AUROC	Loss
Fingerprint model	0.7870	0.8457	0.4498
SMILES model	0.7861	0.8451	0.4505
Multi-modal model	0.7883	0.8461	0.4527

All models perform comparably, with the **Multi-Modal model slightly** leading in predictive performance despite a marginally higher loss



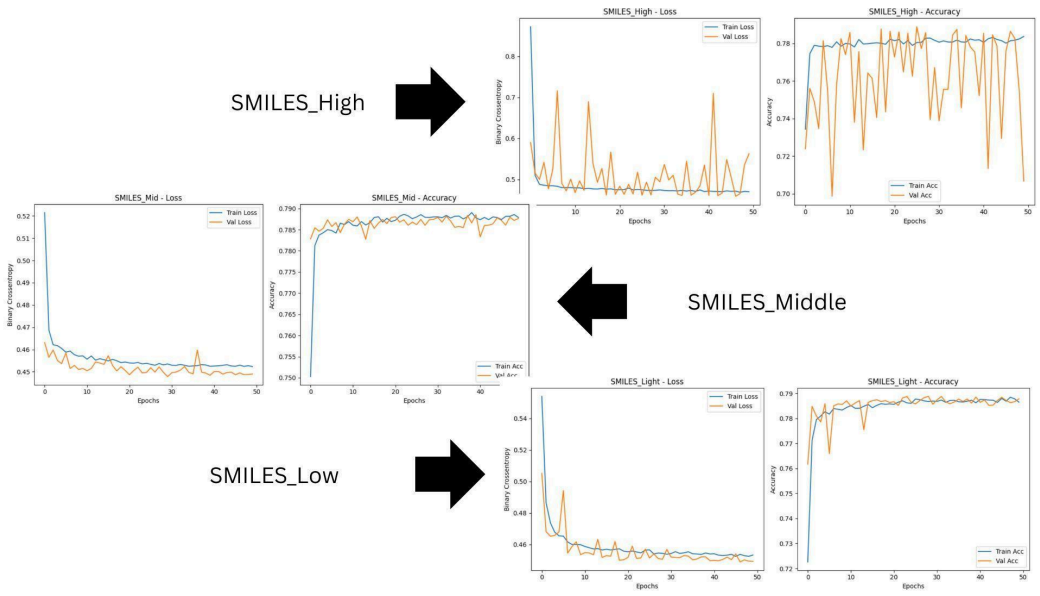
But based on the training curves, the **SMILES 1D CNN model** shows the best fit, with closely aligned training and validation loss and accuracy, indicating excellent generalization and stability. The **Multi-Modal Architecture (Fingerprints + SMILES)** also fits well, with minimal overfitting and consistent performance after early epochs. In contrast, the **Fingerprints MLP model** shows slight overfitting, as training accuracy continues to improve while validation accuracy plateaus. Overall, **SMILES 1D CNN mode** is the most balanced and best-fitted among the three and having more potential to be improved than **Multi-Modal Architecture (Fingerprints + SMILES)**

Fine-Tuning hyperparameters

Architectural tuning & Hyperparameter tuning

SMILES Model Complexity Exploration, Given that the SMILES-based architecture demonstrates strong performance in both AUROC and accuracy, I further investigate its capacity by separating it into three variants with varying architectural complexity. Starting from the well-fitted base model using a 1D CNN and max-pooling over tokenized SMILES input, I design three configurations: **High, Moderate, and Low complexity**. Each variant systematically adjusts **the number of convolution filters**, **dense layer sizes, dropout rate, and regularization strategy.** This fine-tuning task aims to evaluate the trade-offs between model complexity and performance, and to determine the most efficient architecture for molecular property prediction using SMILES strings:

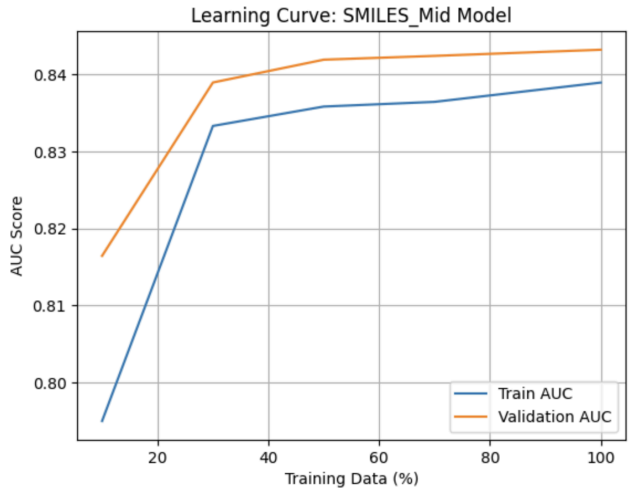
Feature	SMILES_High	SMILES_Mid	SMILES_Light
Conv1D Filters	128	64	32
Kernel Size	5	5	3
Dense Layers	2 (64, 32 units)	1 (32 units)	1 (16 units)
Regularization	L2 on dense layers	None	None
Dropout Rate	0.5 after each dense layer	0.3	0.2
Batch Normalization	After each dense layer	None	None
Model Complexity	High	Moderate	Low
Expected Behavior	Best for large, complex datasets	Balanced performance and efficiency	Fast, low-resource training, less expressive



Metric	SM-Light	SM-Mid	SMILES_HIGH
Best Val AUC	0.8451 (Epoch 49)	0.8458 (Epoch 40)	0.8437 (Epoch 36)
Best Val Accuracy	0.7888 (Epoch 29)	0.7885 (Epoch 40)	0.7889 (Epoch 26)
Best Val Loss	0.4490 (Epoch 47)	0.4478 (Epoch 30)	0.4589 (Epoch 47)
Stability (late training)	Slightly noisy	Plateaued, stable	Mild fluctuations, stable
Early performance	Slower ramp-up	AUC ~0.835 at Epoch 2	Starts at AUC 0.8255 (Epoch 1)
Overfitting risk	Low, minor fluctuations	Low, minimal gap	Moderate — good performance with some noisy epochs
Runtime per epoch	~4–9s	~4–6s	~ 5–11s
Warnings/Errors	Checkpoint + EarlyStopping warnings	None	None

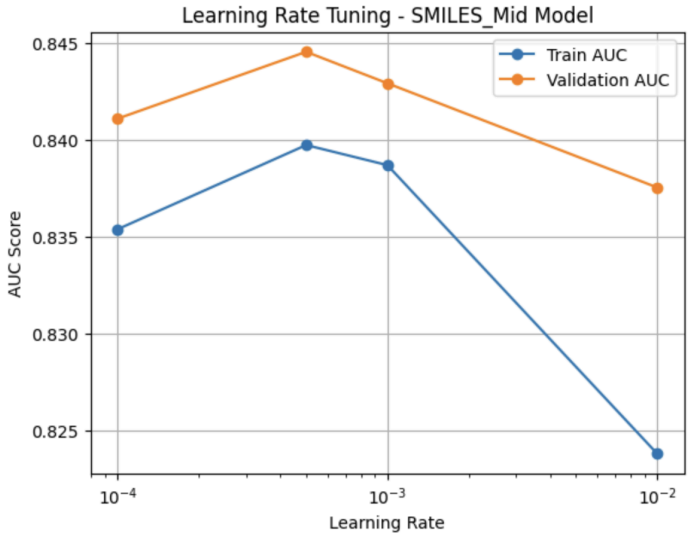
The SM-Mid architecture demonstrated consistently strong performance, with minimal variance and signs of early convergence.

Learning Curve (Train % vs AUC)



Model performance was evaluated using training subsets of 10%, 30%, 50%, 70%, and 100%. The AUC plateaued at 50% training data, indicating that larger subsets did not significantly enhance accuracy and may increase overfitting. Consequently, **I adopted a 50:50 train-validation split** to reduce training time and mitigate overfitting without sacrificing predictive performance.

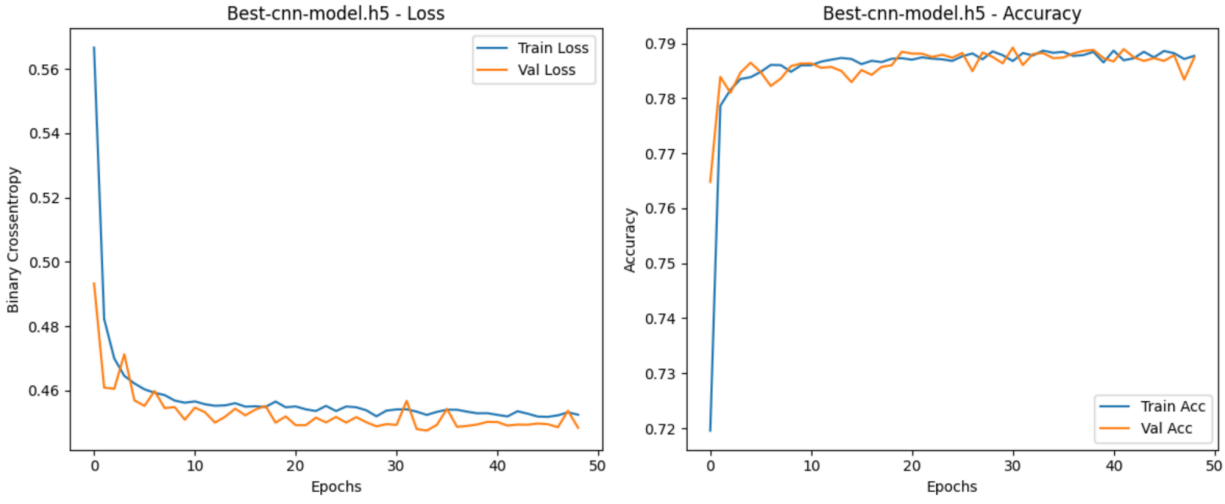
For Learning Rate Tuning:



I also conducted a grid search over the learning rates [0.01, 0.001, 0.0005, 0.0001]. The model trained with learning rate = 0.0005 achieved the highest validation AUC and best generalization, indicating a well-balanced convergence speed and stability.

Best CNN model:

- Selected Model: SM-Mid (CNN architecture)
- Learning Rate: 0.0005
- Training Split: 50% train / 50% validation
- Epochs: 200



This configuration offers a balance of accuracy, efficiency, and robustness, making it the optimal choice for downstream evaluation.

Model comparison (another model vs CNN by using external dataset)

External dataset: https://ctd2-data.nci.nih.gov/Public/Broad/CTRPv2.0_2015_ctd2_ExpandedDataset/

```
GDSC shape: (96103, 12)
CTRP shape: (383857, 21)

 Unique GDSC drugs: 227
Unique CTRP drugs: 545
Overlapping drugs: 1

 GDSC label counts:
 sensitivity_label
1    64413
0    31690
Name: count, dtype: int64

 CTRP label counts:
 sensitivity_label
1    200757
0    183100
Name: count, dtype: int64
```

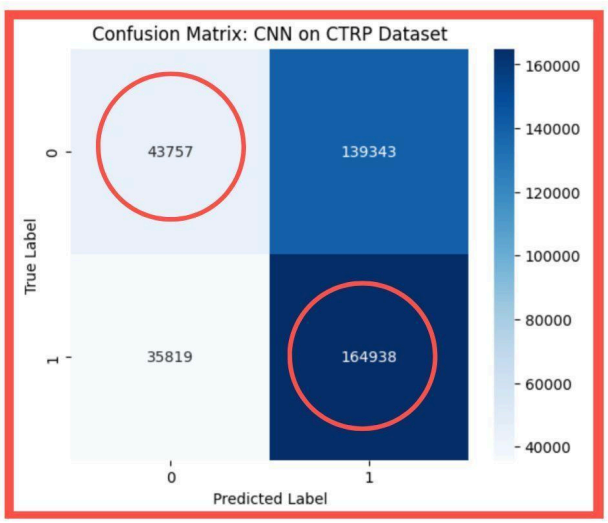
To evaluate model generalization, I preprocessed the CTRPv2.0 dataset [links] by filtering compounds with valid **EC50 values** and generating binary sensitivity labels using a log-transformed EC50 threshold **(log(EC50) ≤ 1.5)**. These were merged with compound metadata using the **master_cpd_id**, and entries **without SMILES were removed.** The final CTRP dataset included 383,857 entries across 545 unique compounds. A comparison with the GDSC dataset revealed only 1 overlapping SMILES compound among 227 GDSC and 545 CTRP unique drugs, confirming minimal overlap. This ensures that **CTRP can serve as a valid external test se**t for evaluating the generalizability of drug sensitivity prediction models trained on GDSC.

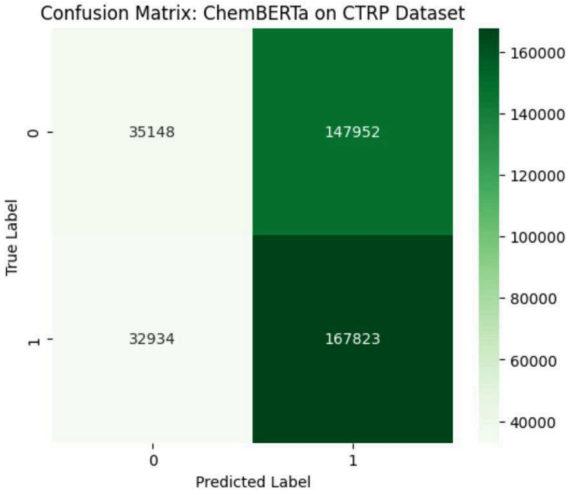
Pretrain model that used for this datasets :

- Model: https://huggingface.co/seyonec/ChemBERTa-zinc-base-v1
- Detail: https://metatext.io/models/seyonec-ChemBERTa-zinc-base-v1

Step	Training Loss
500	0.518300
1000	0.479100
1500	0.468000
2000	0.462200
2500	0.457400
3000	0.452600
3500	0.456200
4000	0.455300
4500	0.453300
5000	0.448600
5500	0.453800
6000	0.449400

I fine-tune a pretrained transformer model for drug sensitivity classification. I used the **ChemBERTa** model (seyonec/ChemBERTa-zinc-base-v1), which was pretrained on SMILES representations of molecules. The GDSC dataset was preprocessed by **filtering valid SMILES** and **binary sensitivity labels**, then **tokenized** using **RobertaTokenizer**. The dataset was split into training and validation sets, converted into PyTorch-compatible format, and fine-tuned over three epochs using the **Hugging Face Trainer API**. After training, the model's performance was evaluated on the validation set, and both the fine-tuned model and tokenizer were saved for **prediction and comparison.**

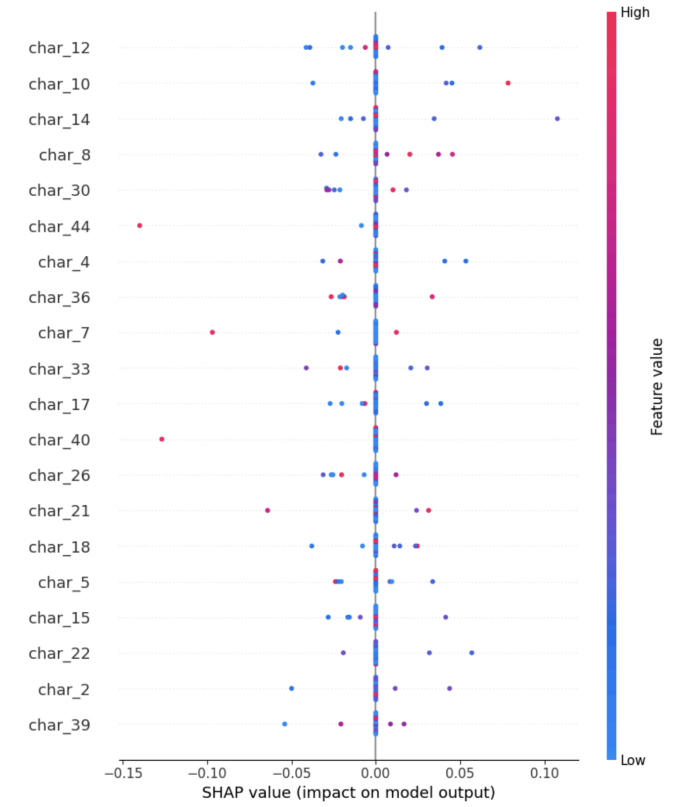




I evaluated two models: 1.**models ChemBERTa** (a transformer-based model pretrained on molecular SMILES) and 2. **CNN model(Best CNN model)** trained on **character-level SMILES** to predict drug sensitivity labels **on the external CTRP dataset. While both models achieved comparable performance on identifying sensitive compounds (F1-score of 0.65 for label 1** (1 = sensitive, 0 = resistant)**)**, **the CNN model outperformed ChemBERTa in overall accuracy (0.54 vs. 0.53)** and showed more balanced **precision** and **recall** across both classes.

The CNN model also demonstrated a higher macro-averaged F1-score (0.49 vs. 0.46), indicating better generalization across drug responses. **Given that early-stage drug screening prioritizes high recall for sensitive compounds** because missing an effective compound (a false negative) can be costly because a promising candidate might be **discarded before further testing.** High recall ensures nearly all potential hits are captured even if it means investigating a few extra false positives which is acceptable because subsequent validation stages can **efficiently filter out the less promising candidates.**

SHAP CNN Model



Position	Most Common Char	Count	Sample Size
char_12	c	4	20
char_10	c	10	20
char_7	c	12	20
char_15	c	9	20
char_8	c	6	20
char_4	c	10	20
char_20	c	9	20
char_5	c	9	20
char_11	c	6	20
char_3	c	11	20
char_2	c	8	20
char_22	c	6	20
char_13	c	8	20
char_19	c	7	20
char_26	c	10	20
char_9	c	11	20
char_21	c	9	20
char_65	[PAD]	14	20
char_17	c	7	20
char_28	c	6	20

high-impact positions is 'c' (aromatic carbon).

To implement SHAP, I loaded a CNN model **(best_SM_Mid.h5)** trained on character-level SMILES, reconstructed the original tokenizer, and preprocessed external SMILES from the CTRP dataset by tokenizing character-by-character, converting to integer sequences, and padding to fixed length (MAXLEN = 100). SHAP analysis was configured using KernelExplainer with 50 background samples and 20 test samples.

Each dot represents a compound's value at that position, color gradient:

- Red = high feature value (higher token index from tokenizer)
- Blue = low feature value (lower token index)

Position-Based Feature Importance

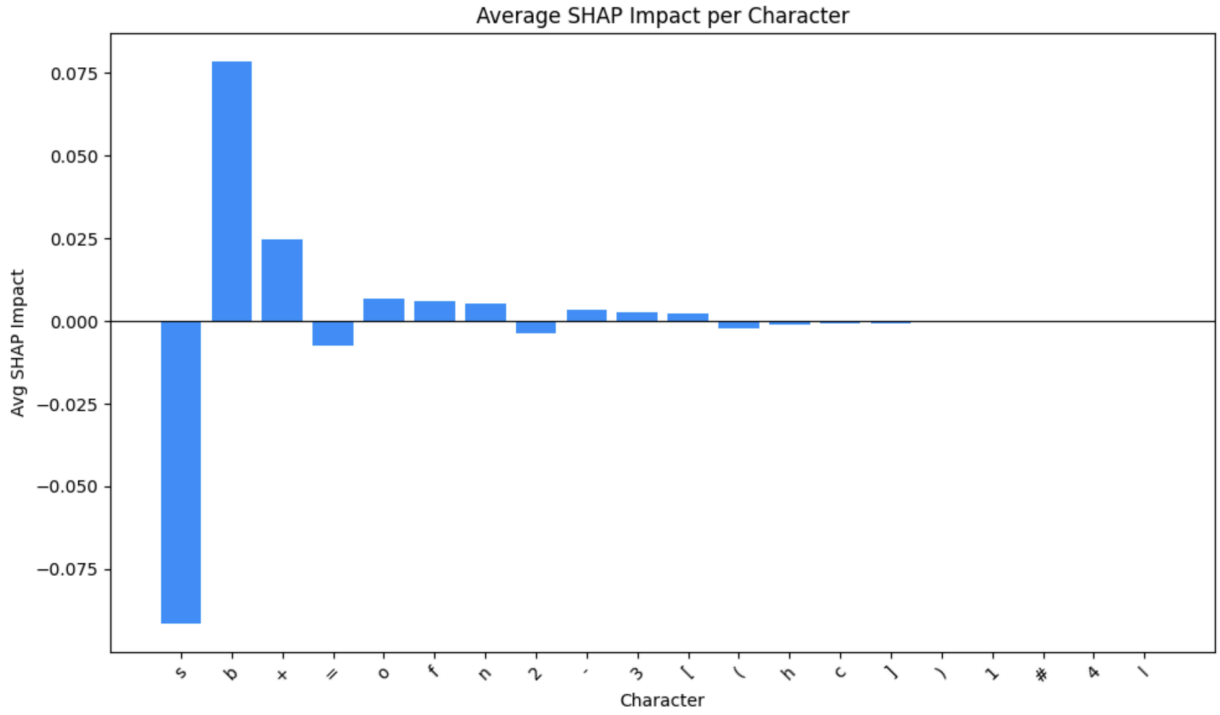
Most influential positions: char_12, char_10, char_14, char_8, char_30, char_44

These positions demonstrate the strongest positive and negative impacts on model predictionsFeature values cluster tightly, indicating specific tokens at these positions drive model decisions

Character Token Distribution

The most frequent character in

SHAP Value by Character Token



Summary table:

Character	Average SHAP Impact	Interpretation
s	-0.091	Strongly decreases predicted activity
b	+0.078	Strongly increases predicted activity
+	+0.025	Moderately increases activity
c	-0.001	Negligible impact despite high frequency

where, s = aromatic sulfur , b = aromatic boron [links], thus:

- s (aromatic sulfur) had a strong negative impact (-0.091) on predicted drug activity
- b (aromatic boron) had a strong positive impact (+0.078) on predicted drug activity