Journal of Computational Biophysics and Chemistry

Deep Sequence models for Ligand based Virtual Screening

Viswajit Vinod Nair, Sonaal Pathlai Pradeep, Vaishnavi Sudheer Nair, Pournami P N
Gopakumar G and Jayaraj P B *

Received Day Month Year
Revised Day Month Year
Accepted Day Month Year

The past few years have witnessed machine learning techniques take the limelight in multiple research domains. One such domain that has reaped the benefits of machine learning is Computer-Aided Drug Design (CADD), where the search space for candidate drug molecules was decreased using methods such as virtual screening. Current state-of-the-art sequential neural network models have shown promising results and we would like to replicate similar results with virtual screening using the encoded molecular information known as SMILES. Our work includes the use of attention-based sequential models - the LSTM with attention and an optimized version of the Transformer network specifically designed to deal with SMILES (ChemBERTa). We also propose the "Overall Screening Efficacy" (OSE), an averaging metric that aggregates and encapsulates a models performance over multiple datasets. We found an overall improvement of about 27% over the benchmark model, which relied on parallelized Random Forests.

Keywords: Virtual Screening and Deep Learning and Sequential Models and RNNs and LSTMs and Attention Gates and Transformers

1. Introduction

Traditionally, drug design was done with wet-lab experiments, also known as in vitro experiments, with a large number of candidate molecules. A traditional drug discovery pipeline includes target identification, target validation, lead identification, candidate optimisation, pre-clinical tests and clinical tests. Each phase in this pipeline lasts at least a year. Developments in the field of Computer-Aided Drug Discovery (CADD) over the past few years has made the process faster, cheaper and more efficient[1]. With the exponential increase in the data available over the past few years and the development of GPU hardware, scientists and engineers have been able to automate stages in the traditional drug discovery pipeline, thus being able to decrease the bottleneck and duration for the entire process. Experiments done with the aid of computational power are called in silico experiments.

Virtual screening is a technique used to reduce the search space of desirable molecules. Virtual screening can be mainly of two types: Structural-Based Virtual Screening (SBVS) and Ligand-Based Virtual Screening (LBVS). In LBVS, the

*Corresponding author - jayarajpb@nitc.ac.in

bioassay of ligands on a specified target molecule is known from the previously conducted wet-lab experiments, the data of which is available online. This data could be preprocessed and fitted to a model to recognise potential properties and patterns that could relate to the likelihood of a ligand-target interaction. The model could then be used to analyse the activity of a new ligand on the given target.

Machine learning has been lauded as "the new electricity", with various models being used in domains ranging from recommendation-based systems to image classification[2, 3]. One of the shortcomings of traditional machine learning algorithms is its necessity for an engineer to extract relevant features from the dataset [4]. This is not the case with deep learning, where the model can decipher, extract and extrapolate on features from the provided dataset. This feature of deep learning has proven itself in the past few years, where basic deep learning models have triumphed over highly sophisticated machine learning algorithms.

Attention-based models have been able to mitigate problems related to vanishing gradients and have proven their efficacy to solve the problem with sequential data. Transformers, which are the successors of attention-based models, have brought in a new era of deep learning in the field of natural language processing and its ideas have started penetrating other domains such as computer vision.

We propose two deep learning models that incorporate the attention mechanism, to perform virtual screening of active and inactive molecules for *Mycobacterium tuberculosis*. We also propose and explain our novel metric "Overall Screening Efficacy". This metric can capture performance over multiple datasets while taking into account the class imbalance of each dataset and how that affects performance.

The rest of the paper is organized as follows: Section 2 contains the literature survey, Section 3 contains the methodology, Section 4 contains the evaluation results, Section 5 covers the discussions and analysis and Section 6 concludes the paper.

2. Literature Survey

Bahi et al. acknowledged the declining efficiency of traditional drug discovery approaches in their paper due to its increasing expense, risk and time[5]. The traditional drug discovery pipeline can take more than 15 years and can cost millions[6]. Furthermore, 90% of candidate drugs fail during human trials. Virtual screening has since been able to decrease the search space of candidate molecules that could bind to a target. The authors further pointed out that deep learning models are faster and more efficient since extracting and transforming features to make them suitable for linear discriminators or tree-based models limit their modelling power, especially while working with highly complex datasets. They proposed a model, named the *DNN-VS*, which used just 5 hidden fully connected layers and with 4 dropout layers with the ReLU function. The hyperbolic tangent function was used in the output layers which can be directly associated with the propensity of a given molecule to interact with a target protein. Despite the simplicity of the deep learn-

ing model, experimental results showed that this model outperformed the previous state-of-the-art models.

Development in the field of AI has never stopped accelerating since the dawn of the 21st century, which has shown the rise in popularity of many different architectures of models such as the Convolutional Neural Networks (CNN)[7] and Recurrent Neural Networks (RNN). CNN's are primarily used in tasks related to the image classification where the property of spatial independence holds. Many papers were published in the field of CNN, some of the most influential being *Residual Neural Networks*[8] and *Inception Models*[9], which continue to be some of the most cited papers in the field of deep learning. RNN's are more suited in domains where there is a need to process sequential data. Information gathered from the previous timesteps are used to influence the learning as well as decision-making rules of the model. Some of the more popular models used in this field are Gated Recurrent Units (GRU)[10] and Long Short-Term Memory (LSTM) units[11].

One of the early work of the Jayaraj et al. named GRAPHSCREEN [12] uses Maximal Common Subgraph method to screen ligands using GPU. Another virtual screening tool was developed by Jayaraj et al. in 2019 which exploited the use of GPUs using Compute Unified Device Architecture (CUDA) programming to perform ligand-based virtual screening[13]. The authors determined that the Support Vector Machine (SVM) model, along with different kernels and a soft-margin, produced good results for classification, but the large dataset used meant that the training time was too long. This was because standard optimisation techniques which are used for solving the quadratic programming problem didn't scale well for larger datasets. In an attempt to overcome this hurdle, the authors used a training set decomposition technique and a novel GPU implementation which divided the entire dataset and executed these subsets in multiple CUDA cores. Comparisons between serial and GPU implementations of SVM using different kernels showed that the performance of both the implementations are the same, but the training time was sped up by atmost 50 times for larger datasets. Another model built by the authors, known as GPURFSCREEN, which used random forests on a GPU environment reaped similar results and also substantially reduced training time.

2D CNNs are generally characterised by their high AUC value and low loss, but not the highest accuracy. Conversely, 1D CNNs are more flexible and are implemented by low capacity models, hence suffer because of the use of one single fingerprint which might not be the proper representation of core bioactivity features. Mendolia et al. resorted to using ensemble classifiers which relatively improved results[14]. The authors evaluated performance over two training schemes, one strongly biased towards the training set without class balancing. The second training scheme is strongly biased towards inactive candidates because it is representative of the general population of candidate compounds. For both training schemes, the results of these CNN models outperformed standard machine learning models like Random Forests and Support Vector Machines.

Deep Transfer Learning aim to transfer knowledge from a 'Source' to a 'Target'.

Transfer learning is useful to learn for small data problems and increases model power for specific tasks, thereby decreasing cost and ramping up convergence for drug discovery. The most common DTL method used in processes like molecular activity predicition is transferred parameter initialization, also known as fine-tuning . For example, Goh et al.[15] proposed *ChemNet*, a CNN pre-trained on ChEMBL data and fine-tuned with HIV, Tox21 and FreeSolv data. After fine-tuning, the results showed a significant improvement. Another type of transfer learning is feature-based learning. Iovanac et al. used an autoencoder architecture which acts as a feature extractor to map the molecules to a shared latent space uses the same model for pK_a value prediction[16].

Transformers are the state-of-the-art in the field of natural language processing. The heart of the transformer model is the self-attention mechanism, which was proposed by Vaswani et al.[17]. In their seminal paper, the authors determined that the gates within a LSTM cell aren't necessary for learning and that the attention blocks suffice to learn from any sequential input. Transformers were used in the field of cheminformatics by Morris et al.[18], where an encoder-decoder architecture was used to measure binding affinity onto a target.

A very common issue that plagues machine learning and deep learning models is generalizability which are not only caused by the model but also from the way data is fed to the model. Using a testing dataset, which follows the same distribution as that of the training set, is prone to report accuracies that needn't represent how well the deployed model might work[19, 20, 21]. Scantlebury et al., in their paper[22] argue that this is especially critical in ligand-based virtual screening, where models are more likely to learn patterns and properties from the ligands, as opposed to learning the properties of the physical interaction between the ligands with the target protein, which is more target-dependent. The authors in their paper augmented their training set with active ligands which were incorrectly positioned and labelled as decoys and reported higher performances, which they associated with the model being able to pick up favourable interactions between the ligand and target atoms.

3. Methodology

3.1. *Dataset*

The "AID datasets", taken from PubChem, consist of various bioassays which describe various molecules, their composition, bonding and other chemical properties. The data was downloaded as "SDF files", with separate files for active and inactive molecules. "SDF" (Structure Data Format) is meant principally for structural information. Multiple compounds are delimited by lines consisting of four dollar signs ($$$$). One of the features of the SDF format is its ability to include associated data.

We used the cheminformatics tool RDKit[23] to transform the data into a dataframe where all the associated data is converted to columns and the molfile

data is represented as an image. This transformed data can now be processed efficiently by deep learning frameworks.

Each molecule in each AID dataset is also represented in its SMILES(Simplified Molecular Input Line Entry System) format[24]. The immense amount of structural and stereo-chemical information held in this format proves useful for learning models and is also preferred to other notations such as InChI. InChI provides a unique identifier for chemical structures whereas SMILES is simpler and is used widely for storage and interchange of chemical structures. The ability to enumerate SMILES also presents an opportunity for data augmentation which can increase model performance. Hence, we use SMILES as the input for our models and perform conditional data augmentation to further improve model performance and deal with class imbalance.

Being a string based format, SMILES can be processed by sequential deep learning models. Hence, in our research we focus on leveraging the power of SMILES and state-of-the-art sequential deep learning models for activity prediction.

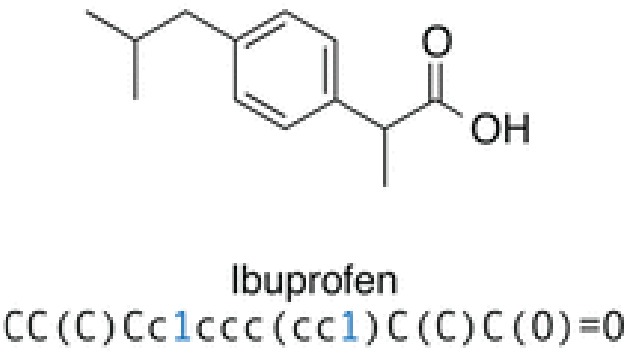

Fig. 1: Ibuprofen and its corresponding SMILES notation

3.2. *LSTM with Attention Model*

3.2.1. *SMILES Augmentation and Encoding*

As part of our preprocessing step, we will be encoding our input. We used an open source software[25] which vectorizes the SMILES molecular representation into a vector format, which has been shown to be more effective. Prior to vectorization, all SMILES strings are enumerated into their different forms, as shown in Figure 3. Since the dataset is highly imbalanced, the batches produced from it will also be imbalanced which can hamper the model's training. To counter this, we modified the iterator such that balanced batches are produced while training by duplicating

the class with the lower samples.This augmentation wasn't performed while testing.

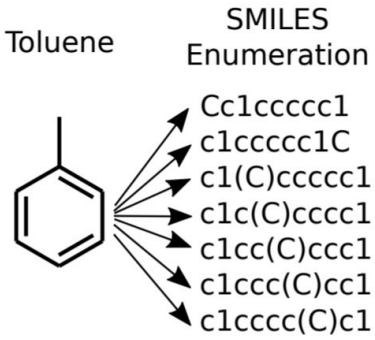

Fig. 2: Example of SMILES enumeration

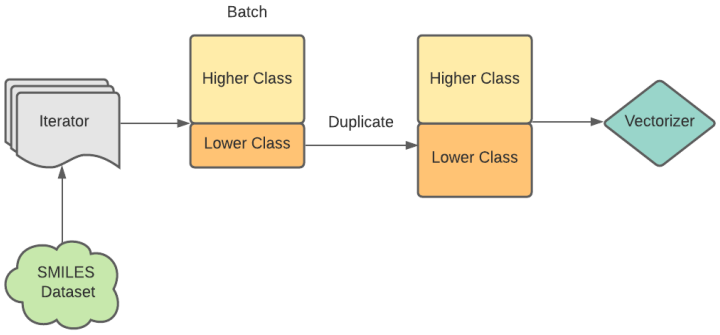

Fig. 3: Single-class augmentation for balanced batches

3.2.2. *Model Architecture*

The LSTM-Attention model, can be divided into 3 parts - the LSTM network, Attention layer and the dense layer. Conventionally, Bi-LSTMs have been used instead of unidirectional LSTMs for sequential tasks as they can learn dependencies in both directions. However, since Bi-LSTMs have twice the weights and take longer to learn, and SMILES is a simpler representation when compared to natural text, we use a linear LSTM network consisting of 2 layers each of 128 cells.

The attention layer is based on the Bahdanau Attention mechanism [26]. This layer helps the LSTM model to further make connections between different units

of the LSTM layer. This is inspired by how humans may read sequential data by giving different emphasis to different sections of a sentence.

$$c_t = \sum_{i=1}^{n} a_{t,i} h_i \tag{1}$$

$$a_{t,i} = align(y_t, x_i) = \frac{e^{score(s_{t-1}, h_i)}}{\sum_{i'=1}^{n} e^{score(s_{t-1}, h_{i'})}} \tag{2}$$

$$score(s_t, h_i) = v_a^\intercal T tanh(W_a([s_t; h_i])) \tag{3}$$

Equation 1 outputs a context vector and equation 2 calculates the softmax of a predefined alignment score. In their paper, Bahdanau et al. used a feed forward neural network to learn the alignment scores[26]. With tanh used as the activation function, the score function given in equation 3 where both v_a and W_a are weight matrices to be learned in the alignment model.

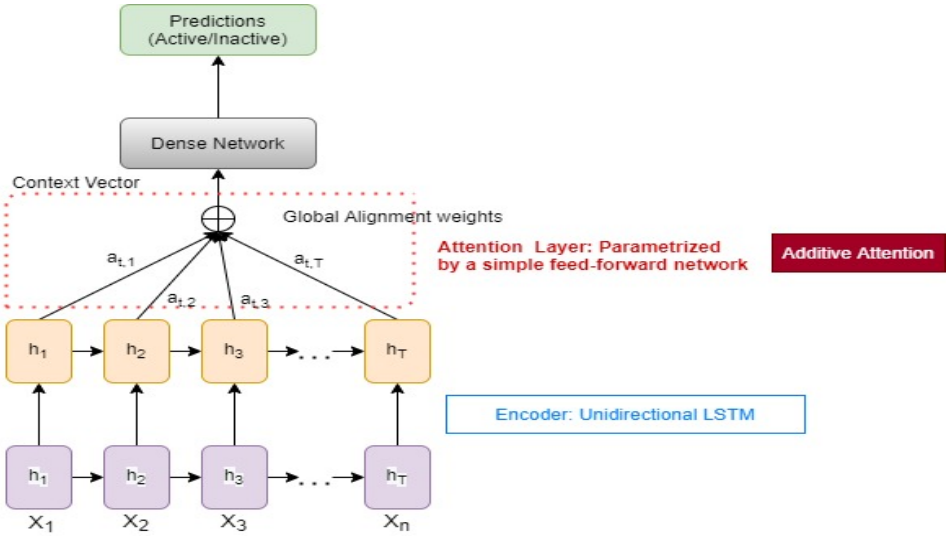

Fig. 4: LSTM with Attention (Bahdanau [26])

Figure 4 shows how attention is integrated into the model. The information from the attention layer is aggregated and passed through 4 dense layers including the output layer. To improve with generalization, dropout layers are added between the dense layers with a dropout value of 0.4. The ReLU activation is used for all dense layers except the output layer which uses sigmoid activation function.

The binary-crossentropy loss is used with the model being optimized by the Adam optimizer which starts with a learning rate of 0.001 for a maximum of 250 epochs. To maximize results and avoid overfitting, we have also used a callback

provided by TensorFlow - ReduceLROnPlateau, to reduce the learning rate when convergence stalls. Here we have used a patience of 10 epochs and the learning rate reduction factor of 0.2.

The model was implemented in Python using the Keras Functional API. The implementation of the Bahdanau attention layer was borrowed from an online source[27].

3.3. *ChemBERTa Classifier*

3.3.1. *ChemBERTa Model*

ChemBERTa[28] is a collection of RoBERTa[29] pretrainings applied on SMILES molecular data for chemical modelling, drug design and property prediction. RoBERTa is a robustly optimized retraining of the already powerful BERT model[30] which is based on transformers. The original transformer consisted of an encoder decoder network based on the attention mechanism. Since BERT performs just language modeling, it consists of only deep encoder networks. RoBERTa, developed by Facebook, further optimizes BERT by modifying a few parameters such as batch size and learning rates. RoBERTa was also trained on an order of magnitude more data than BERT for a longer amount of time.

Currently, ChemBERTa models have been pre-trained on the task of Masked-Language Modelling of SMILES strings. A random sample of the tokens in the input SMILES string (sentence in the case of RoBERTa) is selected and replaced with the special token [MASK]. The MLM objective is a cross-entropy loss on predicting the masked tokens, as shown in 4, where P refers to the empirical distribution of SMILES and Q refers to the predicted distribution. One of the pre-trainings are taken and is finetuned to our datasets to perform a binary classification task.

$$H(P, Q) = E_P[-logQ] \tag{4}$$

3.3.2. *SMILES Augmentation and Balancing*

As part of augmentation, we performed a preset number of random enumerations for each SMILES string. The code uses many RDKit functions such as *MolFromSmiles(), RenumberAtoms(), MolToSmiles()*[23]. The enumeration is carried out in a two-step process such that class imbalance can also be mitigated and any possible duplication is handled. First, each SMILES is enumerated by a replication factor, which we have chosen as 3. Subsequently, additional enumerations our carried out on the class with lower number of samples. After this balancing, duplicates are removed to obtain the final balanced dataset. Figure 5 shows the process in detail.

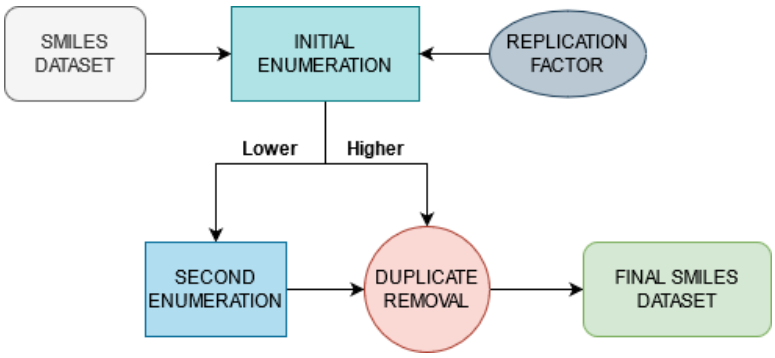

Fig. 5: Two-step enumeration for balancing the SMILES dataset

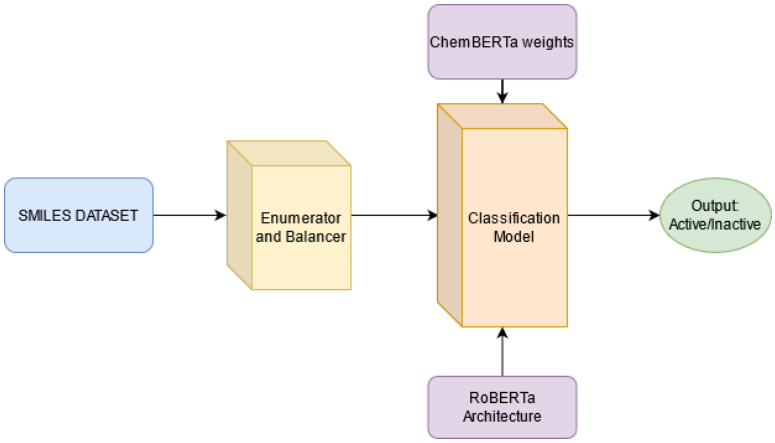

Fig. 6: ChemBERTa Classifier

3.3.3. *Classifier*

For fine-tuning ChemBERTa to a classification task, we make use of the ClassificationModel class from the simpletransformers[31] library. As shown in Figure 6, this module accepts the pretrained model weights (ChemBERTa) and the model type (RoBERTa) and predictions are made.

3.3.4. *Hyperparameters*

To facilite in training, the early stopping callback offered by the simple transformers library is used. This callback monitors the training loss for a preset number of epochs (known as the patience) and stops training when no substantial improvement is observed. While training, the patience was set to 4 epochs and could be trained for atmost 100 epochs. The Adam optimizer was used with a learning rate of $0.4 * 10^{-5}$.

4. Evaluation

Each dataset was divided into training and testing sets with a split of 0.2. The randomness of the SmilesEnumerator class used in encoding causes the testing features to have a slightly different representation at each pass. Hence, the transformation for the testing features and subsequent score calculation is repeated 100 times and a voting is carried out over the predictions for each testing data. The majority class is considered as the final prediction for the LSTM+Attention model. For the ChemBERTa Classifier, we make use of the pretrained model named "SMILES_tokenized_PubChem_shard00_160k" made available by the authors on HuggingFace[32], an open source provider of transformers and other NLP models. To compare our results, we use the benchmark i.e the parallel random forest algorithm (GPURFSCREEN)[33].

4.1. *Preliminary Evaluation*

Our initial analysis of model performance over each data set is done by the following standard metrics: Precision, Recall, F1-score and Accuracy. For virtual screening, the active class is arguably the more important one and as a result Precision and F1-score are the most important scores that need to be maximized. As we can see in Table II, III and IV, all our models improve on the random forest model in most datasets. Figures in Table I show a graphical comparsion of precision and F1 scores for each model across all datasets. The biggest score increment is seen in AID 778. There is also a significant increase in scores in high-imbalance datasets viz. AID 893 and AID 2330. ChemBERTa, being a version of RoBERTa, a superior implementation of the Transformer architecture, is the best performing model.

4.2. *Time Analysis*

A major drawback of deep learning models is their training time and development cost. Those are applicable here as well, with our sequential models taking significantly more time to train than the baseline GPURFSCREEN model. Moreover, the balanced enumeration of SMILES also increases the size of each dataset, further increasing the training time for our models. LSTM with Attention takes the most amount of time to train. ChemBERTa does not take as much as it is just fine tuning a pretrained model. The hardware configuration used for training and testing is shown in Table V

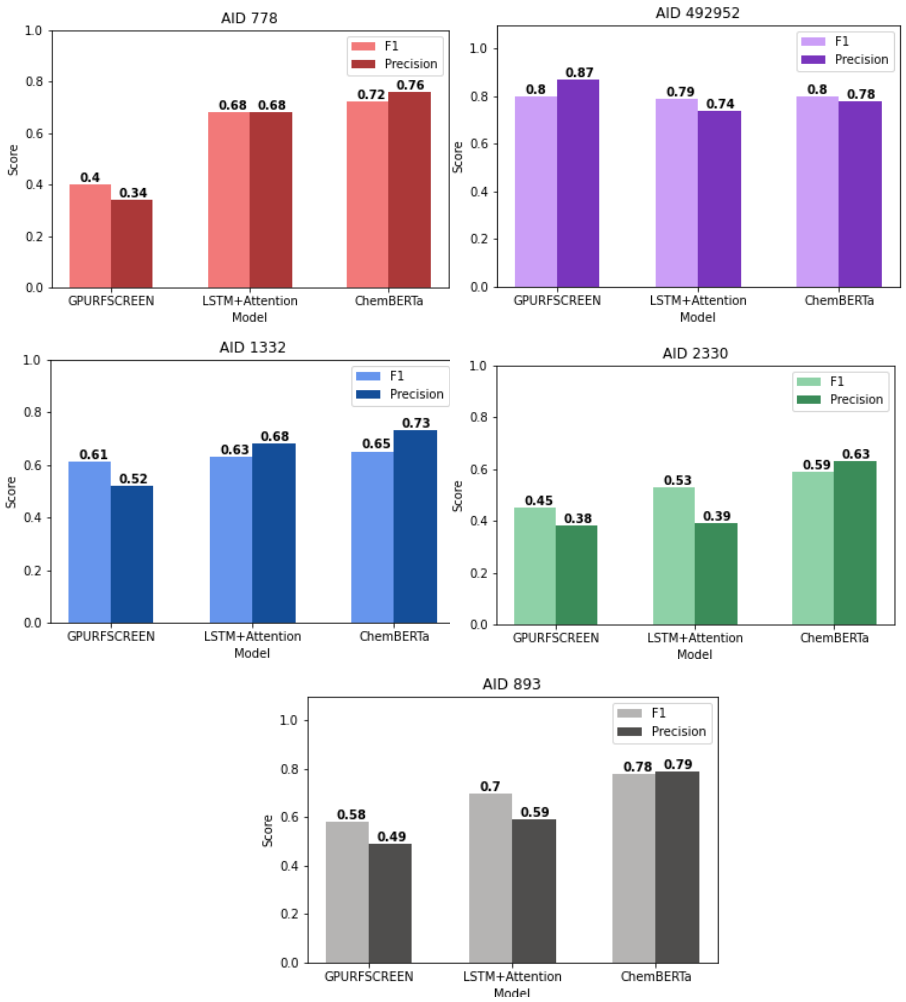

Table I: F1 and Precision score across various datasets

5. Discussion

Analysis of results has clearly shown the superiority of deep learning models in the effectiveness of virtual screening. The benchmark, GPURFSCREEN leveraged all the features of each dataset using a tool called PowerMV to convert SDF files to column features. On the other hand, we have made use of solely the SMILES representation of each molecule. Despite that, our deep learning implementations are able to outperform the benchmark by significant margins.

Table II: Performance of GPURFSCREEN

Dataset	Precision	Recall	F-Score	Accuracy
AID 1332	0.52	0.74	0.61	0.92
AID 492952	0.87	0.73	0.80	0.72
AID 2330	0.38	0.54	0.45	0.72
AID 893	0.49	0.72	0.58	0.94
AID 778	0.34	0.48	0.40	0.78

Table III: Voted Performance of LSTM with Attention

Dataset	Precision	Recall	F-Score	Accuracy
AID 1332	0.68	0.60	0.63	0.79
AID 492952	0.74	0.85	0.79	0.72
AID 2330	0.39	0.87	0.53	0.77
AID 893	0.59	0.86	0.70	0.83
AID 778	0.68	0.69	0.68	0.67

Table IV: Performance of ChemBERTa-Classifier

Dataset	Precision	Recall	F-Score	Accuracy
AID 1332	0.73	0.58	0.65	0.81
AID 492952	0.78	0.82	0.80	0.74
AID 2330	0.63	0.56	0.59	0.88
AID 893	0.79	0.77	0.78	0.90
AID 778	0.76	0.69	0.72	0.68

Table V: Hardware configuration

Particulars	GPU details
Model	NVIDIA Tesla V-100
CUDA cores	5120
Clock Speed	1380 MHz
Memory	32510 MB
Memory bandwidth	900 GB/s
Peak performance	120 TFlops
Compute Capability	7
CUDA Version	11.3

The weights used for ChemBERTa are from a pretraining that was done for Masked-Language Modelling. Those weights were then fine-tuned on our binary

classification task. The optimized transformer architecture of ChemBERTa lead to improvements over the benchmark and LSTM+Attention models. Although our models outpeform the benchmark in most datasets, the change is not entirely uniform and could depend on a number factors which include the class imbalance and the difference in the type and assimilation of data.

5.1. *Overall Screening Efficacy*

Class imbalance is one of the major issues faced while developing binary classification solutions. Figure 7 details the class distribution in the AID datasets. With respect to virtual screening of molecules, we propose a metric named as "Overall Screening Efficacy" (OSE), which encapsulates the overall screening performance of a model over all datasets. With this metric, we intend to prioritise the performance of each model on datasets with higher class imbalance. This can help us evaluate the effectiveness of a model on imbalanced data. OSE is defined as a weighted average of all the F1 scores of each dataset. With the F1 score being a second order metric, OSE becomes a 3rd order metric over multiple datasets. Such a metric is especially useful in generalizing performance of virtual screening models where dataset has a common target for different bioassay data.

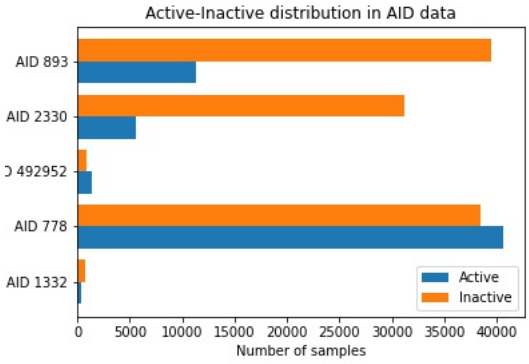

Fig. 7: Distribution of samples in AID datasets

$$OSE = \frac{\sum_{i=1}^{n} w_i F1_i}{\sum_{i=1}^{n} w_i} \tag{5}$$

Where w_i is the weight for the i^{th} dataset. It is calculated as:

$$w_i = \frac{Number\ of\ Inactive\ Classes}{Total\ number of samples}$$

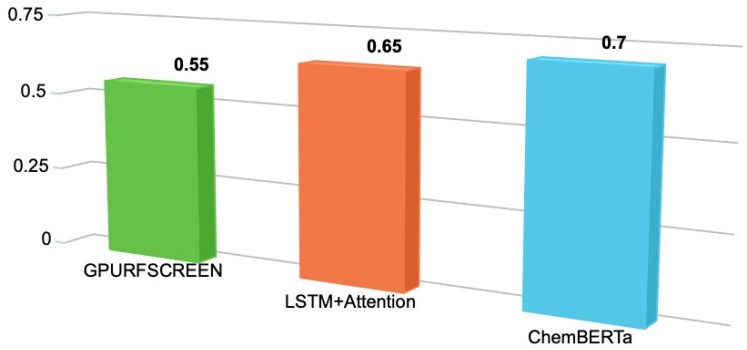

Fig. 8: Comparison of OSE scores

Figure 8 shows a comparison of OSE scores across all our models and the baseline model. Our models show improvement over the baseline. As expected, ChemBERTa performs the best, beating the LSTM+Attention model and GPURFSCREEN.

Our proposed metric, OSE, encapsulates model performance over all datasets for a particular target (in this case *Mycobacterium tuberculosis*). With respect to OSE, LSTM+Attention shows an 18% increase over the benchmark, whereas ChemBERTa shows a 27% increase.

6. Conclusion and Future Work

In our work, we've analysed the efficacy of using deep learning models on virtual screening. Attention based models have been the standard in natural language processing, with transformers being the state-of-the-art. We have implemented many models and have performed a comparative analysis. We have also proposed using a weighted metric which could help us analyse the overall performance of a model over multiple datasets.

The improvement of our deep learning models over the baseline shows the potential for such models in biochemistry and it's sub-fields. The COVID-19 pandemic has also amplified the need for faster and effective virtual screening, and proper application of deep learning models can meet this need.

7. Acknowledgements

We would like to thank Department of Computer Science & Engineering, NIT Calicut for their constant support to complete this work. We also thank Central Computer Center, NIT Calicut for providing the GPU infrastructure for training our Neural Networks.

References

1. Kapetanovic, I. Computer-aided drug discovery and development (CADDD): In silico-chemico-biological approach. *Chemico-Biological Interactions* **2008**, *171*, 165–176.
2. Lane, T.; Russo, D. P.; Zorn, K. M.; Clark, A. M.; Korotcov, A.; Tkachenko, V.; Reynolds, R. C.; Perryman, A. L.; Freundlich, J. S.; Ekins, S. Comparing and Validating Machine Learning Models for Mycobacterium tuberculosis Drug Discovery. *Molecular Pharmaceutics* **2018**, *15*, 4346–4360.
3. Gertrudes, J.; Maltarollo, V.; Silva, R.; Oliveira, P.; Honorio, K.; da Silva, A. Machine Learning Techniques and Drug Design. *Current Medicinal Chemistry* **2012**, *19*, 4289–4297.
4. Jayaraj, P.; Mithun, K.; Gopakumar, G.; Jaleel, U. A. A GPU based virtual screening tool using SOM. *International Journal of Computational Biology and Drug Design* **2021**, *14*, 64–80.
5. Bahi, M.; Batouche, M. Deep Learning for Ligand-Based Virtual Screening in Drug Discovery. **2018**, 1–5.
6. Schuhmacher, A.; Gassmann, O.; Hinder, M. Changing R&D models in research-based pharmaceutical companies. *Journal of translational medicine* **2016**, *14*, 105.
7. LeCun, Y.; Bengio, Y., et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks* **1995**, *3361*, 1995.
8. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition* **2016**, 770–778.
9. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition* **2015**, 1–9.
10. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* **2014**,
11. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural computation* **1997**, *9*, 1735–1780.
12. Jayaraj, P.; Rahamathulla, K.; Gopakumar, G. A GPU based maximum common subgraph algorithm for drug discovery applications. 2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). 2016; pp 580–588.
13. Jayaraj, P.; Jain, S. Ligand based virtual screening using SVM on GPU. *Computational Biology and Chemistry* **2019**, *83*, 107143.
14. Mendolia, I.; Contino, S.; Perricone, U.; Pirrone, R.; Ardizzone, E. A Convolutional Neural Network for Virtual Screening of Molecular Fingerprints. **2019**, 399–409.

15. Goh, G. B.; Siegel, C. M.; Vishnu, A.; Hodas, N. O. ChemNet: A Transferable and Generalizable Deep Neural Network for Small-Molecule Property Prediction. **2017**,
16. Iovanac, N. C.; Savoie, B. M. Improved Chemical Prediction from Scarce Data Sets via Latent Space Enrichment. *The Journal of Physical Chemistry A* **2019**, *123*, 4295–4302.
17. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. Attention is All You Need. **2017**,
18. Morris, P.; Clair, R. S.; Hahn, W. E.; Barenholtz, E. Predicting Binding from Screening Assays with Transformer Network Embeddings. *Journal of Chemical Information and Modeling* **2020**, *60*, 4191–4199.
19. Chen, L.; Cruz, A.; Ramsey, S.; Dickson, C. J.; Duca, J. S.; Hornak, V.; Koes, D. R.; Kurtzman, T. Hidden bias in the DUD-E dataset leads to misleading performance of deep learning in structure-based virtual screening. *PLOS ONE* **2019**, *14*, 1–22.
20. Sieg, J.; Flachsenberg, F.; Rarey, M. In Need of Bias Control: Evaluating Chemical Data for Machine Learning in Structure-Based Virtual Screening. *Journal of Chemical Information and Modeling* **2019**, *59*, 947–961.
21. Imrie, F.; Bradley, A. R.; van der Schaar, M.; Deane, C. M. Protein Family-Specific Models Using Deep Neural Networks and Transfer Learning Improve Virtual Screening and Highlight the Need for More Data. *Journal of Chemical Information and Modeling* **2018**, *58*, 2319–2330, PMID: 30273487.
22. Scantlebury, J.; Brown, N.; Delft, F. V.; Deane, C. M. Dataset Augmentation Allows Deep Learning-Based Virtual Screening To Better Generalize To Unseen Target Classes, And Highlight Important Binding Interactions. **2020**,
23. Landrum, G. RDKit: Open-source cheminformatics.
24. Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Modeling* **1988**, *28*, 31–36.
25. Bjerrum, E. J. SMILES Enumeration as Data Augmentation for Neural Network Modeling of Molecules. *CoRR* **2017**, *abs/1703.07076*.
26. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. **2014**, cite arxiv:1409.0473Comment: Accepted at ICLR 2015 as oral presentation.
27. McAteer, M. Getting started with Attention for Classification. *matthewmcateer.me* **2018**,
28. Chithrananda, S.; Grand, G.; Ramsundar, B. ChemBERTa: Large-Scale Self-Supervised Pretraining for Molecular Property Prediction. 2020.
29. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* **2019**, *abs/1907.11692*.
30. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* **2018**,

abs/1810.04805.

31. Rajapakse, T. C. Simple Transformers. **https://github.com/ThilinaRajapakse/ simpletransformers, 2019**,
32. Wolf, T. et al. HuggingFace's Transformers: State-of-the-art Natural Language Processing. **2020**,
33. Jayaraj, P.; Ajay, M. K.; Nufail, M.; Gopakumar, G.; Jaleel, U. A. GPURF-SCREEN: a GPU based virtual screening tool using random forest classifier. *Journal of cheminformatics* **2016**, *8(1)*, 1–10.