ISSN 2278-3091
Volume 9, No.1, January – February 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse29912020.pdf
https://doi.org/10.30534/ijatcse/2020/29912020

Long Short-Term Memory Recurrent Neural Network Architectures for Prediction of HIV-1 Protease Cleavage Sites

Anwar RHEMIMET[1], Said RAGHAY[2], Omar BENCHAREF[3], Younes CHIHAB[4]

[1]Department of Computer Sciences, Faculty of Sciences and Techniques, Cadi Ayyad University, Morocco, rhemimet@gmail.com
[2]Department of Computer Sciences, Faculty of Sciences and Techniques, Cadi Ayyad University, Morocco, s.raghay@uca.ac.ma
[3]Department of Computer Sciences, Faculty of Sciences and Techniques, Cadi Ayyad University, Morocco, o.bencharef@uca.ma
[4]Department of Computer Sciences, Superior School of Technology, Ibn Toufail University, Kenitra, Morocco, y.chihab@uca.ac.ma

ABSTRACT

Proteases of human pathogens are becoming increasingly important drug targets [1]. Especially, the human immunodeficiency virus type 1 (HIV-1) aspartic protease which is an important enzyme owing to its imperative part in viral development and a causative agent of the deadliest disease known as acquired immune deficiency syndrome (AIDS). Hence, it is necessary to understand the substrate specificity and to interpret this knowledge in practical and useful ways [2]. Therefore, a rational design of an efficient inhibitor requires a good understanding of the HIV-1 protease specificity, i.e., knowing which amino acid sequences are cleaved by the protease and which are not [3]. This is, however, difficult since it cleaves at several different sites that have little or no sequence similarity.

Experimental methods of identification of HIV-1 protease cleavage sites are generally time-consuming and labor-intensive, that's why using machine learning methods to predict cleavage sites and optimize results has become highly desirable [4].

Key words: HIV-1 protease, Cleavage sites, Recurrent Neural Networks, Long Short-Term Memory, Machine learning, Pseudo amino acid composition.

1. INTRODUCTION

Proteases of human pathogens are becoming increasingly important drug targets, hence it is necessary to understand their substrate specificity and to interpret this knowledge in practically useful ways. New methods are being developed that produce large amounts of cleavage information for individual proteases and some have been applied to extract cleavage rules from data. However, the hitherto proposed methods for extracting rules have been neither easy to understand nor very accurate. To be practically useful, cleavage rules should be accurate, compact, and expressed in an easily understandable way [5].

HIV-1 protease is the principle etiologic agent of AIDS discovered by Gallo and coworkers in 1984 [6]. It is able to infect and destroy the human immune system, and allows life threating infection. HIV-1 PR, a homodimeric enzyme belonging to aspartate family also known as aspartyl retropepsin, plays a crucial role in viral maturation [7]. HIV constructs many of its protein in one long piece consisting of several tandemly linked proteins. HIV-1 PR has a responsibility to cleave Gag and Gag-Pol polyproteins into their component proteins responsible for the maturation of new virions, which can then infect new cells [4]. Thus, an HIV-specific protease is necessary for the HIV to make more functional viruses. Without HIV-1 PR, it is not possible for HIV to replicate due to unavailability of infectious virion and it remains uninfected.

Currently, researchers have partially succeeded to develop HIV protease inhibitors that are accessible for HIV treatment. However, they have conditional drawbacks such as poor bioavailability and excruciating infectiousness [8] that lead researchers to proceed with their endeavors to create novel and more potent compounds. Also, due to the tremendous amount of potential peptides, it is difficult to discover inhibitors by ordinary ways to deal with testing various types of peptides one by one, which is more labor-intensive and time-consuming.

The purpose of our study is to understand the substrate specificity of human immunodeficiency virus (HIV)-1 protease because it is important when designing effective HIV-1 protease inhibitors. Furthermore, characterizing and predicting the cleavage profile of HIV-1 protease is essential to generate and test hypotheses of how HIV-1 affects proteins of the human host.

The extraction of HIV-1 protease cleavage has two motives: one is to describe the experimental data available to prove the power of the algorithm that will be used, the other is to design a method of predict new cleavage sites. In the latter case, which is by far the most common motivation, the test data must be different from the data used to form the algorithm. A correct evaluation of the methods must be done on test data that have not been involved in the formation of the algorithms. This is not usually done especially that the currently available computer methods for predicting protease cleavage of HIV-1 can be improved.

2. BACKGROUND

To help medicine and find solutions to solve the problem of HIV protease identification, researchers tend to adopt in silico approaches to predict HIV-1 protease cleavage sites [10]. In recent years, there are studies that have incorporated biological features related to this case to machine learning algorithms and that have provided better predictive performance compared to traditional approaches.

Machine learning algorithm is one that can learn from experience with respect to some class of tasks and a performance measure. Machine learning methods are suitable for molecular biology data due to the learning algorithm's ability to construct classifiers / hypotheses that can explain complex relationships in the data. Recently, several works have approached the HIV-1protease specificity problem by applying techniques learning machine. In [21], [22] the authors used a standard feedforward multilayer perceptron (MLP) to solve this problem, achieving an error rate of 12%. In [4] the authors confirm the result of [23], [22] using the same data and the same MLP architecture, showing that a decision tree was not able to predict the cleavage as well as MLP. In [12] the authors showed thatHIV-1 protease cleavage is a linear problem and that thebest classifier for this problem is linear-SVM (L-SVM).

You et al. [11] incorporated machine learning algorithms including artificial neural network (ANN) and support vector machine (SVM) to examine the specificity of an HIV-1 protease for the discovery and development of effective protease inhibitors. Kontijevkis et al. [4] used an extensive dataset collected from HIV proteome research, and designed a rule-based predictive model on rough sets to analyze the specificity of HIV-1 protease.

Several bioinformatics researchers have attacked this problem using a diversity of methods [9]. It was early on claimed that the problem required non-linear methods. However, it was demonstrated that the relatively few experimental data (362 octamers at the time) did not support a non-linear model [12]. Few years later, when more experimental data were available, linear methods [linear support vector machines (LSVMs)] still performed better than non-linear ones [5]. This was when the methods were evaluated through out-of-sample testing on a large dataset from human proteins [20].

So It was speculated that linearity could be a characteristic for the HIV-1 protease cleavage problem [9]. (Note that linear and non-linear relate to when the standard orthogonal encoding is used).

Other studies were used with less precision on the results obtained: In the article [25], authors presented a web server for predicting cleavage by many different proteases, using support vector regression together with many different features. Features were encoded with bi-profile Bayesian feature extraction and selected using a Gini score. They used the larger dataset from Schilling and Overall plus other published data on cleavage of full proteins. Niu et al. (2013) in the article [26] used a correlation-based feature subset selection method combined with genetic algorithms to search for the best subset in a large set of features. This gave better performance than the standard methods when evaluated with cross-validation. Other authors [27] used a sequence representation and introduced a feature selection method (that removed features). They reported improved prediction results with this when tested with cross-validation on a small dataset with only cleaved octamers.

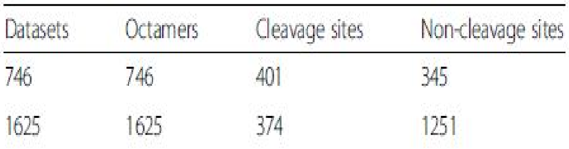
3. METHODOLOGY AND EXPERIMENTS

3.1 Datasets

The datasets used, which are the most important part, are available at the UCI Machine Learning Repository. The tools used are all standard and easily available.

In the present study, two benchmark datasets were used in our proposed method. The benchmark datasets are collections of octamers containing cleavage and non-cleavage sites as shown in Table 1. The 746 and 1625 datasets contain 746 (401 cleaved and 345 non-cleaved) and 1625 (374 cleaved and 1251 non-cleaved).

Table 1: Two benchmark datasets for HIV-1 cleavage site prediction.



Datasets	Octamers	Cleavage sites	Non-cleavage sites
746	746	401	345
1625	1625	374	1251

Amino acids are the essential components of peptides and proteins, and each of 20 amino acids has unique but different properties. The combination of the properties of various residues within a protein can influence diversification and characteristics of the protein structure and function. The aim of the study is to develop a better prediction model using various combinations of features that can predict the HIV-1 protease cleavage sites.

3.2 Methods

In [12] authors assert that neural networks are not to be used for classifying this database. Traditional neural networks assume that all inputs (and outputs) are independent of each other. But for many tasks that's a very bad idea. As in the case of this database. RNNs are called recurrent (Figure 1.) because they perform the same task for every element of a sequence, with the output being depended on the previous computations. Another way to think about RNNs is that they have a "memory" which captures information about what has been calculated so far.

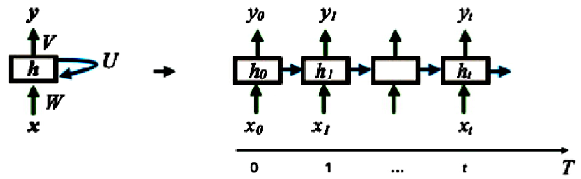


Figure 1: Operation and logic of RNNs

LSTM (Figure 2.) replaces the normal RNN cell and uses an input, forget, and output gate. As well as a cell state.

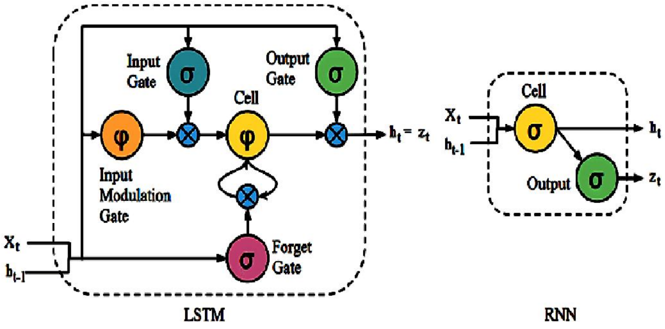


Figure 2: Operation and logic of LSTM (A single memory block is shown for clarity)

These gates each have their own set of weight values. The whole thing is differentiable (meaning we compute gradients and update the weights using them) so we can backprop through it.

We want our model to be able to know what to forget, what to remember. So when new a input comes in, the model first forgets any long-term information it decides it no longer needs. Then it learns which parts of the new input are worth using, and saves them into its long-term memory. And instead of using the full long-term memory all the time, it learns which parts to focus on instead. Basically, we need mechanisms for forgetting, remembering, and attention. That's what the LSTM cell provides us.

By this method, we want to predict if an octamer (a sequence of 8 amino acids) will cleave based on the sequence composed of 8 letters representing amino acids. There are 20 possible amino acids in each seqeunce (A, C, D, E , F ,…). So one-hot encode our sequences by putting 1 in the corresponding letter and 0 otherwise. This means each sequence will be represented by a (8 x 20) dimensional matrix.

We will the use character sequences which make up each case as our X variable, with Y variable as 1/-1 indicating if the case will cleave or not. We use a stacked LSTM model and a final dense layer with softmax activation (many-to-one setup). Categorical cross-entropy loss is used with adam optimizer. A 20% dropout layer is added for regularization to avoid over-fitting.

3.3 LSTM Network Architectures

The LSTM contains special units called memory blocks in the recurrent hidden layer. The memory blocks contain memory cells with self-connections storing the temporal state of the network in addition to special multiplicative units called gates to control the flow of information. Each memory block in the original architecture contained an input gate and an output gate.

The input gate controls the flow of input activations into the memory cell. The output gate controls the output flow of cell activations into the rest of the network. Later, the forget gate was added to the memory block [17]. This addressed a weakness of LSTM models preventing them from processing continuous input streams that are not segmented into subsequences. The forget gate scales the internal state of the cell before adding it as input to the cell through the self-recurrent connection of the cell, therefore adaptively forgetting or resetting the cell's memory. In addition, the modern LSTM architecture contains peephole connections from its internal cells to the gates in the same cell to learn precise timing of the outputs [18].

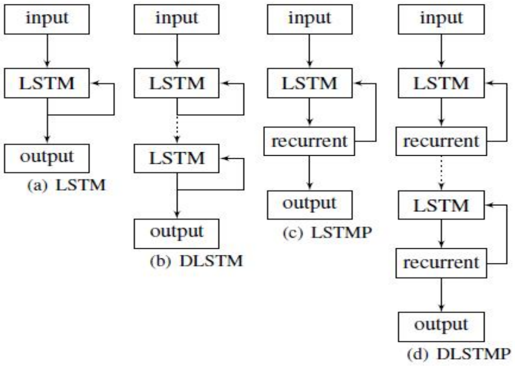


Figure 3: LSTM RNN architectures

The goal of the LSTM is to estimate the conditional probability p(y1, . . . , yT′ |x1, . . . , xT) where (x1, . . . , xT) is an input sequence and y1, . . . , yT′ is its corresponding output sequence whose length T′ may differ from T. The LSTM computes this conditional probability by first obtaining the fixed-dimensional representation v of the input sequence (x1, . . . , xT) given by the last hidden state of the LSTM, and then computing the probability of y1, . . . , yT′ with a standard LSTM-LM formulation whose initial hidden state is set to the representation v of x1, . . . , xT :

$$p(y_1, \ldots, y_{T'} | x_1, \ldots, x_T) = \prod_{t=1}^{T'} p(y_t | v, y_1, \ldots, y_{t-1})$$

Figure 4: Equation LSTM

In this equation, each p(yt|v, y1, . . . , yt−1) distribution is represented with a softmax over all the words in the vocabulary. We use the LSTM formulation from Graves [19].

4. EXPERIMENTS AND RESULTS

In this study, we propose a prediction method based on a Recurrent Neural Network architecture to improve and optimize the results of HIV-1 protease cleavage sites.

The data contains lists of octamers (8 amino acids) and a flag (-1 or 1) depending on whether HIV-1 protease will cleave in the central position (between amino acids 4 and 5).

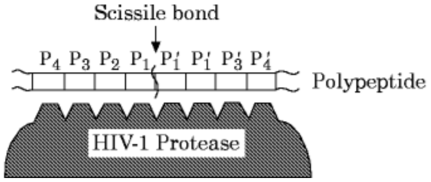


Figure 5: Schematic representation of the HIV-1 protease and substrate subsists. The scissile bond is located between the P’1 and the P1 subsists.

The octamer is represented using one-hot encoding, where each amino acid is encoded to a 20-bit vector, with 19 bits set to zero and one bit set to one. This maps each octamer to an 8 by 20 binary matrix.

Two benchmark datasets were used in our experiments [13]: one containing 746 octamers and another 1625 (The datasets are available at the UCI Machine Learning Repository).

In the proposed approach, we use an LSTM (Long Short-Term Memory) based (RNN) Recurrent Neural Network architecture [14]. Unlike feedforward neural networks, RNNs take as their input not just the current amino acid, but also every other one they perceived previously in time [12]. Moreover, LSTM which is a particular type of Recurrent Neural Networks contains special units called memory blocks in the recurrent hidden layer. The memory blocks contain memory cells with self-connections storing the temporal state of the network. This way, our built model learns the long-term context and dependencies between the amino acids composing each octamer input sequence.

In addition, LSTMs have multiplicative units called gates which control the flow of information into the memory cell and out of the cell to the rest of the network [15]. The gates block or pass on information based on its stregth and significance which they filter using their own set of weights. These weights are adjusted via the recurrent networks learning process which we carefully optimize [16]. Namely, the cells learn when to allow data to enter, leave or be deleted through the iterative process of making predictions, backpropagating error, and adjusting weights via iterative optimization [11].

We use the character sequences which make up each octamer case as our input variable, while the output variable is 1/-1 indicating if the case will cleave or not. We use a stacked LSTM model, a final dense layer with Softmax activation (many-to-one setup) and categorical cross-entropy loss is used with ADAM optimizer. A 20% dropout layer is added for regularization to avoid over-fitting. In addition, we use stratfied cross-validation to evaluate the objective performance of cleavage site prediction. And calculated the specificity, sensitivity and AUC (Area Under the ROC Curve) to quantify how good the model does on the test data. We implemented our deep learning model using Keras, a high level neural network API, with Tensorflow as a backend.

We used the algorithm stacked LSTM model and a final dense layer with softmax activation (many-to-one setup) on all data. Categorical cross-entropy loss is used with adam optimizer. A 20% dropout layer is added for regularization to avoid over-fitting.

Compared to SVM, RNN shows a clear improvement in cross validation (Figure 5). In other words, in almost all results deep recurrent networks show improvement compared to SVM. To overcome the lower results of linear SVM classifier on dataset operated by Rögnvaldsson et al. (2015) in [13], he has trained another nonlinear SVM classifier and has shown that 746 and 1625 dataset are linear while Schilling and Impens are nonlinear.

On the contrary, our approach is to train a single nonlinear classifier (RNN). There is a single reason for this. The application hasn’t changed and the datasets are only samples of a whole population. While we accept that 746 and 1625 are more likely to be linear, in practice one doesn’t know if a completely new sample would behave in accordance to linear models. So we can conclude that a properly regularised RNN architecture can be appropriate for all datasets. In other words, a single model should suffice.

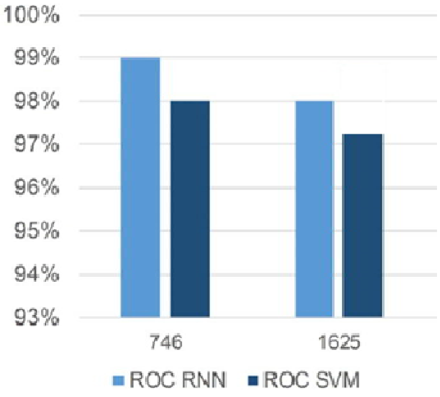


Figure 5. Performance comparison of Recurrent Neural Network and SVM [13] on the two separate databases.

Table 2 shows our approach alongside Oğul [28]. To the best of our knowledge, Oğul has achieved the best results thus far on 1625 dataset. It is to be noted that, compared to VCMC, deep Recurrent Neural Network (RNN) classifier has almost the same performance on 1625 dataset. But, while Oğul has focused its attention to a single dataset and had only reported VCMC’s accuracy on 1625 dataset, LSTM results are tested on two datasets. It is easy to tune the classifier for a single dataset and boost the performance on it while losing the performance on other datasets.

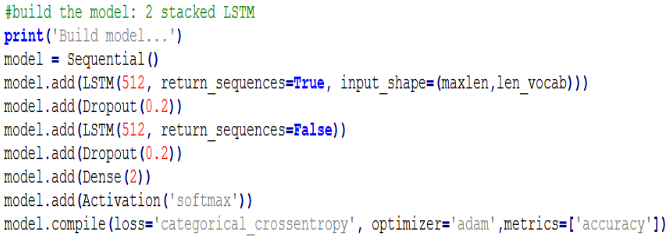
Therefore, we recommend testing on multiple datasets. Thus, it can be argued that RNN’s results are more comprehensive and the resulting classifier is more generalised.

Table 2 : Accuracy results for RNN and VCMC.

	746 (%)	1625 (%)
LSTM	96,4	97,09
VCMC	NA	97

N/A represents results that are not available and are not reported by the original authors (Oğul, Ref [28])

Thus, below the results obtained on the global tests (dataset) with extracts of source code:

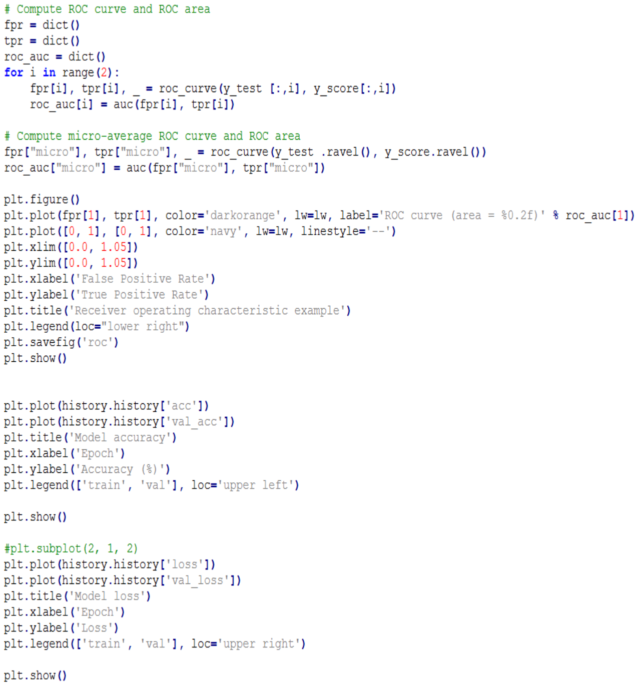


```
#build the model: 2 stacked LSTM
print('Build model...')
model = Sequential()
model.add(LSTM(512, return_sequences=True, input_shape=(maxlen,len_vocab)))
model.add(Dropout(0.2))
model.add(LSTM(512, return_sequences=False))
model.add(Dropout(0.2))
model.add(Dense(2))
model.add(Activation('softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam',metrics=['accuracy'])
```

Table 3: Results that show the accuracy rate

Test accuracy : **0,96717**			
	Precision	*Recall*	*F1-score*
0	0,92	0,99	0,95
1	0,99	0,96	0,97
AVG	0,97	0,97	0,97

Table 3: Results that show the accuracy rate



```
# Compute ROC curve and ROC area
fpr = dict()
tpr = dict()
roc_auc = dict()
for i in range(2):
    fpr[i], tpr[i], _ = roc_curve(y_test [:,i], y_score[:,i])
    roc_auc[i] = auc(fpr[i], tpr[i])

# Compute micro-average ROC curve and ROC area
fpr["micro"], tpr["micro"], _ = roc_curve(y_test .ravel(), y_score.ravel())
roc_auc["micro"] = auc(fpr["micro"], tpr["micro"])

plt.figure()
plt.plot(fpr[1], tpr[1], color='darkorange', lw=lw, label='ROC curve (area = %0.2f)' % roc_auc[1])
plt.plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')
plt.xlim([0.0, 1.05])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic example')
plt.legend(loc="lower right")
plt.savefig('roc')
plt.show()


plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('Model accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy (%)')
plt.legend(['train', 'val'], loc='upper left')

plt.show()

#plt.subplot(2, 1, 2)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend(['train', 'val'], loc='upper right')

plt.show()
```

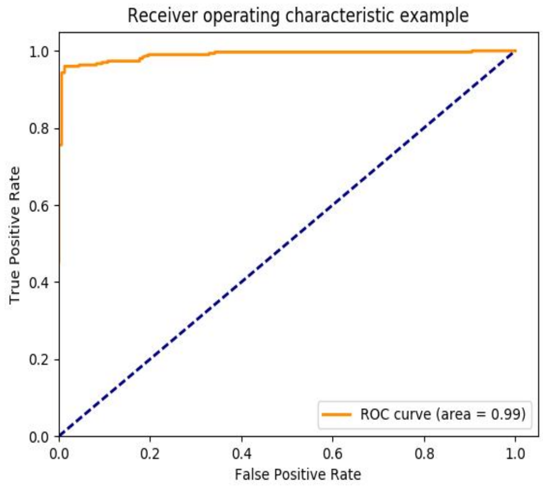


Figure 6. ROC curve

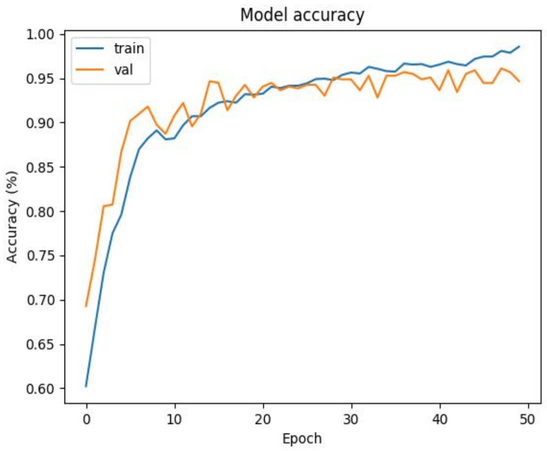


Figure 7. Model accuracy

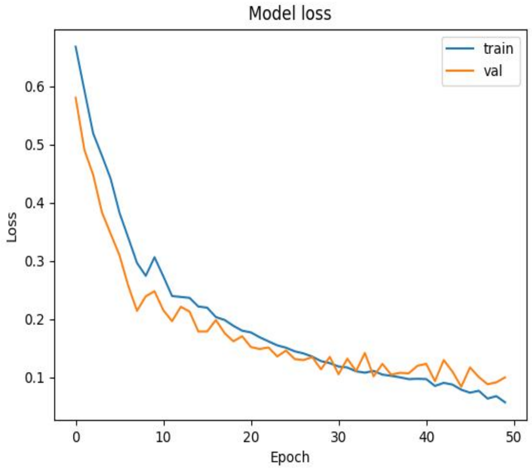


Figure 8. Model loss

5. CONCLUSION

Many of the drugs that are used by HIV infected humans can be catalogued as protease inhibitors. These drugs mainly restrict the protease activity and therefore reduce the formation of mature proteins. By studying the protease and predicting its cleavage sites, one can hope to achieve better drugs. Predicting HIV-1 protease cleavage problem has been addressed by many machine learning approaches but its classification false positive and the lack of an approach with good generalisation remains a challenge. Furthermore, the experiments demonstrate that the proposed approach shows a clear improvement in predicting cleavage by HIV-1 protease. They also show that the size of training data and the choice of the hyper-parameters are very important factors when it comes to performance. In addition, the results obtained indicate a better precision with a very low error rate. We have tested this approach on several datasets and achieved 96,7% accuracy.

REFERENCES

[1] De Clercq, E.2004. Antiviral drugs in current clinical use. J. Clin. Virol. 30:115–133. https://doi.org/10.1016/j.jcv.2004.02.009

[2] Betts, M. J., and R. B. Russell. 2003. Amino acid properties and consequences of substitutions, p. 289–316. In M. R. Barnes and I. C. Gray (ed.), Bioinformatics for geneticists, vol. I. Wiley, Chichester, United Kingdom.

[3] Beck, Z. Q., L. Hervio, P. E. Dawson, J. H. Elder, and E. L. Madison. 2000. Identification of efficiently cleaved substrates for HIV-1 protease using a phage display library and use in inhibitor development. Virology 274:391–401. https://doi.org/10.1006/viro.2000.0420

[4] Kontijevskis,A. et al. (2007) Computational proteomics analysis of HIV-1 protease interactome. Proteins, 68, 305–312. https://doi.org/10.1002/prot.21415

[5] Rögnvaldsson T.et al. . (2009) How to find simple and accurate rules for viral protease cleavage specificities. BMC Bioinformatics , 10, 149.

[6] Gallo RC, Montagnier L. The discovery of HIV as the cause of AIDS. N Engl J Med. 2003;349(24):2283–5. https://doi.org/10.2147/HIV.S79956

[7] Verkhivker GM. Coarse-Grained Modeling of the HIV-1 Protease Binding Mechanisms: II. Folding Inhibition. In: Computational Intelligence Methods for Bioinformatics and Biostatistics. Springer Berlin Heidelberg; 2009. p. 13-24.

[8] Lv Z, Chu Y, Wang Y. HIV protease inhibitors: a review of molecular selectivity and toxicity. HIV AIDS (Auckl). 2015;7:95–104.

[9] Rögnvaldsson,T. et al. (2007) Bioinformatic approaches for modeling the substrate specificity of HIV-1 protease: an overview. Expert Rev. Mol. Diagn., 7, 435–451.

[10] Chou KC. Prediction of human immunodeficiency virus protease cleavage sites in proteins. Anal Biochem. 1996;233(1):1–14.

https://doi.org/10.1006/abio.1996.0001

[11] You L, Garwicz D, Rognvaldsson T. Comprehensive bioinformatic analysis of the specificity of human immunodeficiency virus type 1 protease. J Virol. 2005;79(19):12477–86.1702–1709.

[12] Rögnvaldsson,T. and You,L. (2004) Why neural networks should not be used for HIV-1 protease cleavage site prediction. Bioinformatics, 20,1702–1709.

[13] Rögnvaldsson T, You L, Garwicz D. State of the art prediction of HIV-1 protease cleavage sites. Bioinformatics. 2015;31(8):1204–10.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory,"Neural Computation, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[15] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term de-pendencies with gradient escent is difficult,"Neural Networks,IEEE Transactions on, vol. 5, no. 2, pp. 157–166, 1994.
https://doi.org/10.1109/72.279181

[16] M. Sundermeyer, R. Schluter, and H. Ney, "Lstm neural networks for language modeling." In INTERSPEECH, 2012, pp. 194–197.

[17] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," Neural Computation, vol. 12, no. 10, pp. 2451–2471, 2000.

[18] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," Journal of Machine Learning Research, vol. 3, pp. 115–143, Mar. 2003.

[19] A. Graves. Generating sequences with recurrent neural networks. In Arxiv preprint arXiv:1308.0850, 2013.

[20] Schilling,C. and Overall,C.M. (2008) Proteome-derived, database-searchable peptide libraries for identifying protease cleavage sites. Nat. Biotechnol.,26, 685–694.
https://doi.org/10.1038/nbt1408

[21] Cai, Y.D., Yu, H., Chou, K.C., 1998. Using neural network for prediction of HIV protease cleavage sites in proteins. J. Protein Chem. 17, 607– 615.

[22] Thompson, T.B., Chou, K.C., Zheng, C., 1995. Neural network prediction of the HIV-1 protease cleavage sites. J. Theoret. Biol. 177, 369–379.
https://doi.org/10.1006/jtbi.1995.0254

[23] Narayanan, A., Wu, X., Yang, Z., 2002. Mining viral protease data to extract cleavage knowledge. Bioinformatics 18, S5–S13.

[24] Cai, Y.D., Chou, K.C., 1998. Artificial neural network model for predicting HIV protease cleavage sites in protein. Adv. Eng. Software 29, 119–128.

[25] Song,J. et al. (2012) PROSPER: an integrated feature-based tool for predicting protease substrate cleavage sites. PLoS One, 7, e50300.
https://doi.org/10.1371/journal.pone.0050300

[26] Niu,B. et al. (2013) HIV-1 protease cleavage site prediction based on twostage feature selection method. Protein Pept. Lett., 20, 290–298.

[27] Ozturk,O. et al. (2013) A consistency-based feature selection method allied with linear SVMs for HIV-1 protease cleavage site prediction. PLoS One, 8, e63145.

[28] Oğul, H. (2009) 'Variable context Markov chains for HIV protease cleavage site prediction', BioSystems, Vol. 96, No. 3, pp.246–250.