A Novel Application Of Learned Position Embeddings And Recurrent Retention Network For Identifying AntiCancer Peptides

A project report submitted

to

MANIPAL UNIVERSITY

For Partial Fulfillment of the Requirement for the

Award of the Degree

of

Bachelor of Technology

in

Computer and Communication Engineering

by

Pranshu Bahadur

Reg. No. 180953063

Under the guidance of

Dr. Satyajit Mahapatra ,
Assistant Professor,
Department of ICT,
Manipal Institute of Technology,
Manipal,
India

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

Dec 2023

Abstract

The precise prediction of anticancer peptide activity is crucial for developing effective cancer therapies. However, traditional methods based solely on sequence or physicochemical properties often fall short in accuracy. Recurrent Retention Networks (RRNs) offer a novel approach by explicitly modeling long-range dependencies within peptide sequences, potentially leading to more accurate activity prediction. In this work, we evaluate the performance of RRNs on several benchmark datasets of anti-cancer peptides, including ACP2 Main, ACP2 Alternate, ACP740, and LEE. We compare RRNs to state-of-the-art methods like ACPred-LAF and AntiCP. Our results show that RRNs consistently outperform other methods, achieving an impressive accuracy of 94.77% on ACP2 Main (surpassing the previous SOTA of 81.8%). Overall, our findings establish RRNs as a powerful new tool for anti-cancer peptide prediction. Their significant accuracy improvements over existing methods hold immense potential for revolutionizing the development of novel peptide-based cancer therapies.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this chapter, we give a high-level overview of the domain and our problem statement.

- Overview: Introduction to the current state of the domain and shortcomings
- Background: About the general domain and how it affects society at a high level.

1.1 Overview

Comparisons with methods in the current literature reveal that our proposed approach consistently outperforms those proposed in this domain. In particular, on the ACP2.0 main dataset [2] LPE + RRN outperforms pre-SOTA [4] by +13.69% in accuracy. The reason we highlight this particular dataset is the difficulty of the independent test set due to the difference between sequences in the train and test splits. This dataset was curated to evaluate the generalizability of classification models. Just to give some context, Pre-SOTA - the ACPred-BMF - achieved an accuracy of 80.81% while our proposed model LPE+RRN achieved 94.5%. Although, the ACPred-BMF is a bidirectional LSTM with a combination of quantitative and qualitative statistical meth-

ods for feature extraction. It outperformed its predecessor, ACPred-LMF [3] which is the application of the Transformer Encoder using multihead-attention. Transformers are a staple in Natural language Processing. LLMs like GPT and Mistral use them as their core mechanism. There is a saying in academia, "Transformers is King". Although this is true so far in general, it was interesting to note that a recurrent architecture such as ACPred-BMF outperformed the Transformers encoder. This finding was the first basis for our hypothesis.

"The nature of Peptide Sequences, specifically the strong dependency between an amino acid in a peptide sequence, and the previous amino acid makes recurrent architectures ideal for classifying Anti-Cancer Peptide sequences".

In this study we evaluate our hypothesis by using a recurrent approach - LPE + Recurrent Retention Network - by comparing it with approaches proposed in the literature on a range of datasets.

1.2 Background

1.2.1 The Big picture

Cancer is a major cause of mortality worldwide, with an estimated 18.6 million new cases and 10 million deaths in 2020. The global pharmaceutical market was valued at USD$1.48 trillion in 2022, and the service addressable market (SAM) for chemotherapy treatment is estimated to be USD$9.5 billion in 2023. The SAM for anti-cancer drug treatment is to be USD$200 million in 2023.

Primarily, cancer is treated using chemotherapy in combination with surgery or radiation therapy. Chemotherapy can be effective in treating many types of cancer, shrinking tumors or slowing their growth, improving survival rates, and is used to treat cancer that has spread to other parts of the body. However, it can also cause a wide range of side effects. Additionally, chemotherapy may not be effective for all types of cancer.

While immunotherapy harnesses the body's immune system to fight cancer. Unlike traditional treatments like chemotherapy and radiation, which directly target cancer cells, immunotherapy empowers the immune system to recognize and destroy cancer cells. This makes research and development into methods alternative to chemotherapy important.

1.2.2 Need for Automated Accurate Peptide Sequence Identification

The increasing feasibility of generating large datasets and the urgent need for accurate computational methods to identify relevant peptide sequences from these datasets. This issue arises due to the interplay of two major trends:

1. AI-powered data generation: Advances in AI have made it possible to generate massive datasets of biological data, including peptide sequences. This data can come from various sources, such as: High-throughput sequencing technologies: These technologies generate vast amounts of genomic and proteomic data, including sequences for millions of peptides. Protein structure prediction tools: AI-powered protein structure prediction methods can predict the 3D structure of proteins, which can be used to identify potential binding sites for peptides. Virtual screening: AI algorithms can be used to virtually screen millions of peptide candidates for desired properties, such as drug-binding potential or enzyme inhibition.

2. Difficulty in identifying relevant peptides: Despite the abundance of data, identifying relevant peptides within these datasets remains a significant challenge. This is because: Most peptides are not functional: Only a small fraction of peptides have specific biological functions. Identifying these functional peptides from the vast pool of non-functional ones requires reliable computational methods.

Data complexity and noise: Biological datasets can be noisy and complex,

containing errors and artifacts. This further complicates the task of accurately identifying relevant peptides.

The interplay of these factors creates a critical need for accurate and efficient computational methods capable of: Distinguishing functional peptides from non-functional ones: This can involve analyzing sequence features, predicting peptide-protein interactions, or using other biophysical and biochemical data. Prioritizing peptides based on their potential relevance: Methods should be able to rank peptides based on their likelihood of having a desired function, such as binding to a specific protein or exhibiting a particular enzymatic activity. Integrating data from multiple sources: Combining data from different sources, such as sequence, structure, and functional annotations, can improve the accuracy of peptide identification. Developing such methods is crucial for maximizing the value of the vast datasets generated by AI in various fields, including: Drug discovery: Identifying peptides with potential therapeutic applications. Enzyme engineering: Designing enzymes with desired properties. Personalized medicine: Tailoring treatments based on individual patient's peptide profiles. The increasing feasibility of data generation through AI presents both a challenge and an opportunity. By developing accurate computational methods for identifying relevant peptide sequences, we can unlock the full potential of this data and unlock breakthroughs in various fields of biology and medicine.

Chapter 2

Theory

In this chapter, we give a high-level overview of the NLP theory and its relation to Peptide sequences.

- Peptides as a Natural language: We propose the intuition for the application of NLP techniques for peptide sequence data.
- Vocabulary, Tokenization, Embeddings: We briefly explain fundamental NLP techniques at a high level
- Deep Learning Architechures in NLP: We give an overview of more DL architectures used in NLP with their trade-offs.

2.1 Peptides as a Natural Language

Peptide sequences in general are composed of a chain of amino acids. Suppose we map each amino acid to a character within the English alphabet. We can then represent a peptide sequence as a chain of letter characters.
Let $P = FAFKA$ be a sample Peptide sequence.
Each character in the above sequence is mapped to a different amino acid.
$\S AminoAcidsX = (A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y)$
Then we can say, $P \in X^L | L$ is sequence length

Note: Changing any amino acid at any position will give the sequence a completely different meaning.

In this way, Peptide sequences have similarities to natural language. If we define a natural language as being the sequential composition of symbols then our peptide sequence $FAFKA$ can be viewed as a "sentence/document". Therefore, it is now viable to use NLP techniques over Peptide sequences.

2.2 Vocabulary

The goal of tokenization is essentially to formulate a vocabulary, i.e. a look-up table between character (in our case) symbols and the index position of that symbol in a look-up table.

For instance, if X = (A, C, D, ..., V, W, Y) is an ordered set of all amino acids then if we consider each character to be a word our vocabulary would map $A \to 0$, $C \to 1$, ..., $X_i \to i$

Note that the above mapping considers each symbol to be a word. However, it is possible to consider subchains of 2-3 amino acids as 1 word instead. This depends on ngrams. i.e. Number of symbols needed to be considered a word.

Also, "word" in this case is not the same as in a language such as English. Rather a "word" is the most atomic representation of a symbol or combination of symbols. For example, if we wanted ngrams to be 2:

Then our vocabulary would consist of AC, CA, AA, CC, etc (cross product of 1-ngram x 1-ngram vocabulary). Moreover, it is also possible to have ngrams that are powersets instead of cross-products.

Through empirical analysis, we consider the number of ngrams to be a hyperparameter for-each dataset.

2.3 Tokenization

Tokenization (At character level):

Tokenization is the encoding of sequences given in a natural language to a given vocabulary. There are several ways to do this. These methods use the following assumptions and are domain-agnostic within NLP:

$\exists P = (p_1, \ldots, p_L | p_i \in X, i \in |P|)$

- P is a peptide sequence of length L.
- X is a set of 20 Amino Acids acids.
- $X = \{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y\}$

The following are the fundamental techniques of tokenization:

2.1.3.a Integer Encoding:

Given: Sequence $ACCD$, X^1

Where, $X^1 = (A, C, D, ..., W, Y)$

Return: (0, 1, 1, 2)

2.1.3.b One-hot Encoding: **Given:** Sequence ACD, X^1

Return: a sparse vector for each symbol in the sequence of len vocab size. Where the character at the given position is represented by 1.

$ACD \rightarrow ((1, 0, 0, \ldots, 0), (0, 1, 0, 0, \ldots, 0), (0, 0, 1, 0, \ldots, 0))$

2.1.3.c Multi Hot Encoding:

Given: Sequence $ACCD$, X^1

Return: A vector of len vocab size where all character indices have appeared in a given sequence is denoted with 1 and the rest are all 0. $ACCD \rightarrow (1, 1, 1, 0, 0, \ldots, 0)$

2.1.3.d Word Count Encoding:

Given: Sequence $ACCD$, X^1

Return: A vector of len vocab size where all character indices have appeared in a given sequence is denoted with the number of occurrences and the rest are all 0. $ACCD \rightarrow (1, 2, 1, 0, 0, \ldots, 0)$

Note: These are some techniques. There are also statistical methods such as the tf_idf algorithm to consider

2.4 Word Embedding Vector

Building upon vocabulary as a look-up table we. We store n-dimensional vectors of weights for each symbol in our vocabulary. This is known as word embeddings. Broadly speaking, word embeddings are vectors that represent the meaning of a word.

For instance:

Although Cat and Hat are two syntactically similar words, they mean completely different things.

Let us say it is rare for both words to occur together in a 'context' (close together in a sequence).

Let us say the following:

Tom wears a normal hat, but his cat does not.

Ideally, we want the words Tom and the hat to have some positive correlation and the words Cat and hat to have a negative correlation.

This is the reason why we use word embeddings as vectors. To capture the

context meaning of words.

Concretely, the measure of similarity between Tom and Hat should be high, and the measure of dissimilarity between the words cat and hat should be high.

2.5 Recurrent Neural Network:

Let $X \in R^{L \times d}$ represent an input sequence of amino acids' word embeddings.

Where,

- $L, d \in Z^+$ represents the Peptide sequence length, d represents word embedding dimensions.
- $T = (0, L)$ represents the position of each amino acid word embedding in X

We can say:

$X = (x_t | \forall t \in T)$

Then a Recurrent Neural Network at state t is defined as this is known as a **RNN cell** [[7]]:

$h_t(x_t, h_{t-1}) = Wx_t + Uh_{t-1}$*

Each RNN cell h_t is based on the previous cell h_{t-1} .i.e. each amino-acid depends on the previous amino-acid. It is this recursion that makes RNN a solution that is intuitive for classification of ACPs. However, RNNs are naive in that each current state output h_t depends only on the previous state output h_{t-1}. This is not ideal for the classification of ACPs because it relies on the assumption that each amino acid in a peptide is only dependent on the previous amino-acid. Although the previous amino-acid **is a factor** there are more factors to consider[[1]].

*Note: Typically we follow this with an activation: $activation(h_t)$ to keep gradients from getting to large.

2.6 Long Short Term Memory (LSTM):

Let $X \in R^{L \times d}$ represent an input sequence of amino acids' word embeddings.

Where,

- $L, d \in Z^+$ represents the Peptide sequence length, d represents word embedding dimensions.
- $T = (0, L)$ represents the position of each amino acid word embedding in X

We can say:

$X = (x_t | \forall t \in T)$

We can define a LSTM hidden state s_t as follows [[6]]:

$(i, f, o)_t = \sigma(W_{i,f,o} x_t + U_{i,f,o} s_{t-1}).split(3)$

$g_t = tanh(W_g x_t + U_g s_{t-1})$

Where,

- i_t is the input gate
- f_t is the forget gate
- o_t is the output gate
- g_t is the cell gate

LSTM cell state is represented as follows:

$c_t = f_t \odot c_{t-1} + i_t \odot g_t$

Finally LSTM hidden state is defined as:

$s_t = o_t \odot tanh(c_t)$

Although LSTM improves on RNN by relying on a pair of vectors c_{t-1}, s_{t-1} per state and extracts different features through gate operations. Specifically, the mapping of $tanh$ activation to σ activation. The use of these activation functions also improves the generalizability of LSTMs. However, with regard

to the peptide sequence, LSTMs alone still do not solve the problem long-range dependencies.

2.7 Enhancing RNNs with Positional Encoding

Positional encoding is a technique used to inject information about the relative order of elements in a sequence into a neural network model. This is particularly valuable for RNNs, which struggle to encode long-range dependencies and differentiate between elements solely based on their embedding vectors.

RNNs update their hidden state sequentially, potentially losing information about earlier elements in the sequence as they process later ones. Positional encoding provides additional information about the relative positions of elements, aiding the model in capturing distant relationships and dependencies.

2.8 Learned Position Encoding

Some models like LSTMs can learn positional information implicitly through their recurrent connections. However, for tasks where long-range dependencies are crucial (Such as ACP classification), explicit positional encoding can be beneficial.

Concretely, we can denote LPE as:

$LPE = XW | W, X \in R^{L \times d}$

Where W denotes the Learned Position Encoding weights and X denotes sequence word embeddings.

We then apply the LPE to X by:

$X := X + LPE_X$

Chapter 3

Methodology

3.1 Outline

In this chapter, we explore the inner workings of Recurrent Retention.

- Intuition: We explain the intuition behind our choice of recurrent retention
- Retention: We provide a technical basis for the conceptual understanding of the Retention Mechanism and our approach to its limitations.

3.1.1 Intuition

Motivation for Recurrent Retention Architecture

This section formally analyzes the reasoning behind adopting the "Retention" architecture with a recurrent formulation for modeling peptide sequences. We consider both its advantages compared to alternative approaches and its intrinsic alignment with the nature of peptide data.

Limitations of Existing Architectures:

1. Recurrent Architectures (RNNs, LSTMs, GRUs):
    - Time Complexity: Sequential processing inherent to these models leads to computational inefficiency, particularly for large datasets.

- Limited Long-Range Dependency Capture: Capturing dependencies between distant elements within the sequence can be challenging for these architectures.

2. Transformer Encoders:

   - Scalability Concerns: While transformers excel at capturing global dependencies, their reliance on the softmax operation poses scalability challenges for large-scale processing.
   - Positional Sensitivity Mismatch: Peptide sequences exhibit strict positional relevance; altering the amino acid order significantly impacts the sequence's meaning. This contrasts with natural languages where word order variations often retain semantic clarity. Notably, transformers lack the inherent recurrent structure that directly models sequential dependencies, potentially hindering their ability to represent such peptide sequences effectively.

Justification for Recurrent Retention with LPE:

In light of these considerations, a recurrent formulation for Retention combined with LPE (or similar positional encoding) emerges as a well-suited choice for peptide sequence modeling. This architecture offers several key benefits:

- Explicit Modeling of Sequential Dependencies: The inherent recurrency in Retention directly captures the dependence of each amino acid on its predecessors, aligning well with the intrinsic properties of peptide sequences.
- Positional Encoding Integration: LPE effectively encodes positional information within the recurrent framework, further addressing the positional sensitivity of peptide sequences.

Comparison and Future Work:

While alternative recurrent formulations, such as RWKV, exist and leverage mechanisms like exponential decay for long-range dependencies, our current focus lies on the efficacy of Retention with LPE for this specific domain. Nonetheless, exploring these alternative architectures in future work remains an interesting proposition.

Relationship between Retention and Softmax:

The decay factor (γ) within the recurrent formulation of Retention essentially replicates the functionality of the softmax operation observed in transformers. Additionally, this factor can act as a positional encoding mechanism in the parallel formulation.

3.2 Retention

3.2.1 Recurrent Formulation

Kindly, refer to [5] for a deeper understanding of Retention.

In this section, we discuss the retention mechanism and its different formulations.

Let $X \in R^{L \times D}$ be an input peptide sequence.

Where,

- L is the length of the peptide sequence.
- D is the dimensionality of the Embedding Vector.

Let $n \in 1, ..., L | L \in Z$ then,

X_n is the mapping of an Amino Acid to an Embedding vector $R^{1 \times D}$ at position n, concretely: $X_n \in R^{1 \times D}$ is the state-wise input to a recurrent cell.

Let $(q, k, v)_n : qkv_i \in qkv_n | qkv_i \in R^{1 \times D}$ define the query, key, and value of each state.

Where, $(q, k, v)_n = X_n W | W \in R^{1 \times 3D}$

Recurrent Retention at state s_n is defined as follows:

$s_n = \gamma s_{n-1} + k_n^T v_n | \gamma \in (0, 1)$

$Retention(X_n) = q_n s_n | \forall n \in L$

Note: $|S| = L + 1$ for additional sentinel state.

3.2.2 Dealing with long-range dependencies:

One of the key drawbacks of any recurrent network is that each state is dependent on the previous state.

To overcome this limitation we propose the use of Learned Position Encoding before passing embedding vectors to the Retention (as stated in the following section).

3.2.3 MultiscaleGatedRetention (MSR) Formulation with LPE:

We define MultiscaleGated Retention for Peptide sequences as follows:

Let h be number of heads : $h = dim_{out}/dim_h$

Let, LPE be a function that represents Learned Position Encoding (This is important for long-range sequence dependencies).

We define γ (decay factor) as:

$\gamma = 1 - 2^{-5-arange(0,h)} \in R^h$

$head_i = Retention_i(LPE_i(X) + X, \gamma_i)$

$Y = GroupNormalization(concat([head_1, ..., head_i]) *$

$MSR(X) = (swish(XW_G) \odot Y)W_O$

* Here GN is scale-invariant (i.e. acts the same as LayerNormalization for each head).

3.2.4 Retention Encoder Layer:

Let $F(X) = gelu(XW)W$ denote a feed-forward layer.

$$X = MSR(LayerNormalization(X)) + X$$

$$X = F(LayerNormalization(X)) + X$$

3.2.5 Retention Algorithm

Algorithm 1: Recurrent Retention Cell

function RecurrentRetentionCell($x_n, s_{n-1}, \gamma = 0.984375$):

1. $(q, k, v)_n := W_{q,k,v}(x_n).split(3)$

2. $s_n := \gamma s_{n-1} + k_n^T v_n$

3. $x_n := q_n s_n$

return x_n, s_n

Algorithm 2: Recurrent Retention

function RecurrentRetention(X, γ):

1. $output := ()$

2. $s := initializeStateZeros()$

3. for $i \in |X|$:

$outputs[i], s := RecurrentRetentionCell(X_i, s, \gamma)$

return concatenate($outputs$)

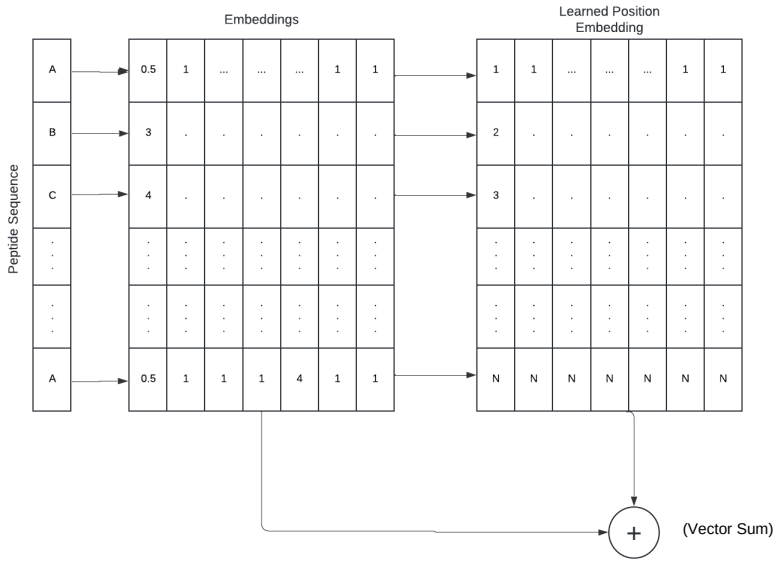

Figure 3.1: Peptide Sequence Features Extraction with Embeddings with LPE

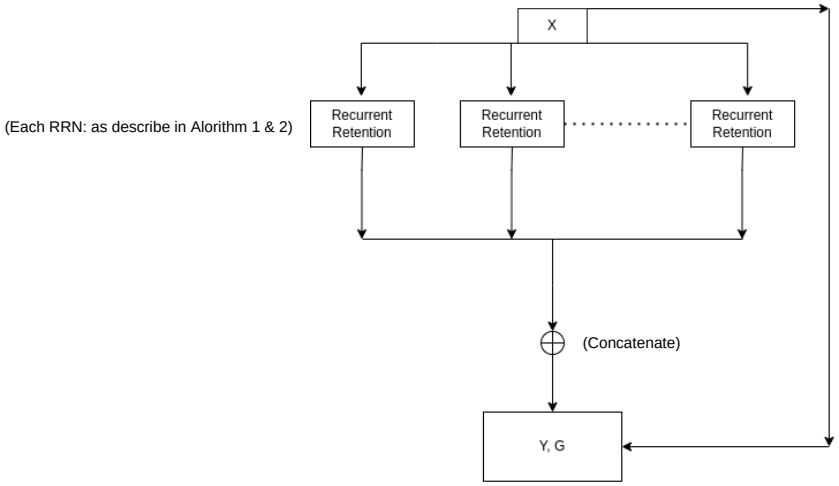

Figure 3.2: MSR Depiction

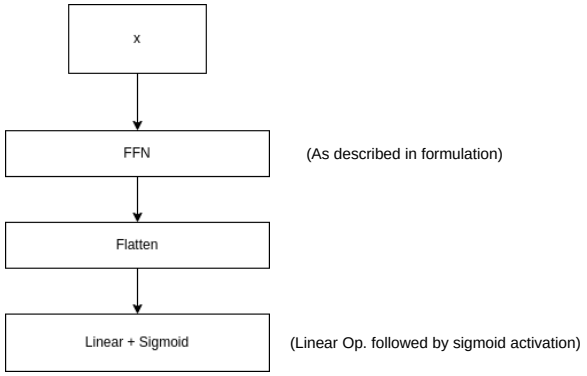

Figure 3.3: FFN + Task Specific Architecture

Chapter 4

Results

4.0.1 Datasets Description

Dataset 1: ACP DL 240

- Description: This dataset includes 240 unique anti-cancer peptide sequences curated from the CancerPPD database.
- Sequence Length: The peptide lengths in this dataset range from 8 to 20 amino acids, as indicated in the table and image.
- Data Split: The data is split for training and testing purposes, with 80% (192 sequences) for training and 20% (48 sequences) for testing.

Dataset 2: ACP DL 740

Dataset	ACPs	Non-ACPs	Total	Max Seq. Len	Min Seq. Len
ACP2.0 Main Train	689	689	1378	50	3
ACP2.0 Main Test	172	172	344	50	2
ACP2.0 Alt. Train	776	776	1552	50	3
ACP2.0 Alt. Test	194	194	388	50	5
ACP 740	376	364	740	97	11
ACP 240	129	111	240	207	11

Table 4.1: Datasets Description

- Description: This dataset contains 740 unique anti-cancer peptide sequences, also derived from the CancerPPD database.
- Sequence Length: Similar to the first dataset, the peptide lengths here range from 8 to 20 amino acids.
- Data Split: The data split follows the same pattern as the previous dataset, with 80% (592 sequences) for training and 20% (148 sequences) for testing.

Dataset 3: ACP2 Main

- Description: This dataset comprises 172 unique anti-cancer peptide sequences. Please note that the previously mentioned value of 1722 appears to be an error in the table; the figure in the image and the total number of rows in the table both indicate 172.
- Sequence Length: The peptide lengths in this dataset vary from 5 to 50 amino acids, as shown in the table.
- Data Split: The data follows the same 80/20 split as the previous datasets, with 137 sequences for training and 34 sequences for testing.

Dataset 4: ACP2 Alternate

- Description: This dataset contains the largest number of unique anti-cancer peptide sequences, with a total of 776. Please note that the previously mentioned value of 1940 appears to be an error in the table; the figure in the image and the total number of rows in the table both indicate 776.
- Sequence Length: Similar to the other datasets, the peptide lengths range from 5 to 50 amino acids.
- Data Split: The data maintains the 80/20 split ratio, with 620 sequences for training and 156 sequences for testing.

4.0.2 Performance Criteria for Classification:

The performance of different models for the testing dataset was evaluated after the completion of the training and validation phases. The models were compared using the following performance metrics: Accuracy, Sensitivity, Specificity, Area under Curve (AUC), Average Precision, and Average Recall.

- True Positive (TP): A case that is correctly classified as an anticancer peptide.
- True Negative (TN): A case that is correctly classified as a peptide.
- False Positive (FP): A case that is incorrectly classified as an anticancer peptide.
- False Negative (FN): A case that is incorrectly classified as a peptide.

Accuracy is the proportion of all cases that are correctly classified. It is calculated using the following equation:

Accuracy = $\frac{(TP+TN)}{(TP+FN+FP+TN)}$

Sensitivity is the proportion of positive ACPs that are correctly classified. It is also called recall. It is calculated using the following equation:

Sensitivity = $\frac{TP}{(TP+FN)}$

Specificity is the proportion of negative ACPs that are correctly classified. It is calculated using the following equation:

Specificity = $\frac{TN}{FP+TN}$

AUC (Area Under Curve) is the area under the curve when the graph is plotted between the true positive rate and false positive rate. A high AUC indicates that the model is correctly classifying the instances.

Dataset	Model	ACC (CV)	ROC (CV)	ACC (test)	ROC (test)
ACP2.0 Main	RRN	**0.9644**	**0.993**	**0.9535**	**0.9953**
	ACPred-BMF	0.7576	0.827	0.8081	0.861
	ACPred-LAF	0.8575	0.9201	0.7907	0.8373
	AntiCP 2.0	0.7529	0.83	0.7543	-
ACP2.0 Alt.	RRN	**0.9864**	**0.989**	**0.9948**	**0.9998**
	ACPred-BMF	0.9149	0.827	0.9356	0.974
	ACPred-LAF	0.9641	0.9826	0.933	0.9638
	AntiCP 2.0	0.901	0.97	0.9201	-
ACP 740	RRN	**0.947**	**0.9899**	-	-
	ACPred-LAF	0.944	0.9611	-	-
ACP 240	RRN	0.825	**0.8979**	-	-
	ACPred-LAF	**0.8948**	0.8694	-	-
LEE + Independent	RRN	**0.9443**	**0.9863**	**0.9767**	**0.9956**
	ACPred-LAF	0.9327	0.9435	0.9667	0.9723

Table 4.2: Comparison Table

Note: our cross-validation is the average forall best metrics in each fold (LAF paper only takes best)

4.0.3 Comparison

RRN Emerges as SOTA for Anti-Cancer Peptide Prediction:

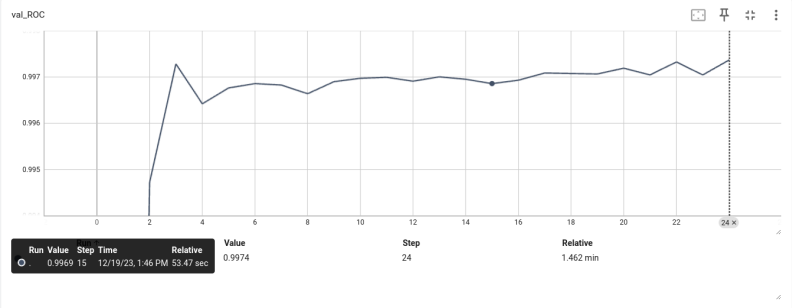

Figure 4.1: RRN ROC Val ACP2.0 Alt.

Based on the table, the RRN model outperforms the other models in several ways. It has the highest overall accuracy (ACC) and area under the ROC curve (AUC) scores for all three datasets: AntiCP 2.0 Main, AntiCP 2.0 Alternate, and LEE Dataset + Independent Test.

For AntiCP 2.0 Main, RRN has an accuracy of 0.9644 and an AUC of 0.993, while the next best model, ACPred-LAF, has an accuracy of 0.8575 and an AUC of 0.9201. For AntiCP 2.0 Alternate, RRN has an accuracy of 0.9864 and an AUC of 0.989, while ACPred-LAF has an accuracy of 0.9641 and an AUC of 0.9826. Finally, for LEE Dataset + Independent Test, RRN has an accuracy of 0.9443 and an AUC of 0.9863, while ACPred-LAF has an accuracy of 0.9327 and an AUC of 0.9435.

In addition to having the highest overall scores, RRN also has the highest scores on the two most important metrics for these tasks: accuracy and AUC. Accuracy is the percentage of predictions that are correct, while AUC is a measure of how well a model can distinguish between positive and negative cases. These results show that RRN is not only the most accurate model overall, but it is also the best at correctly classifying positive and negative cases.

Overall, the results in the table show that RRN is a clear outperformer compared to the other models. It has the highest accuracy and AUC scores on all three datasets, and it is the best model at correctly classifying positive

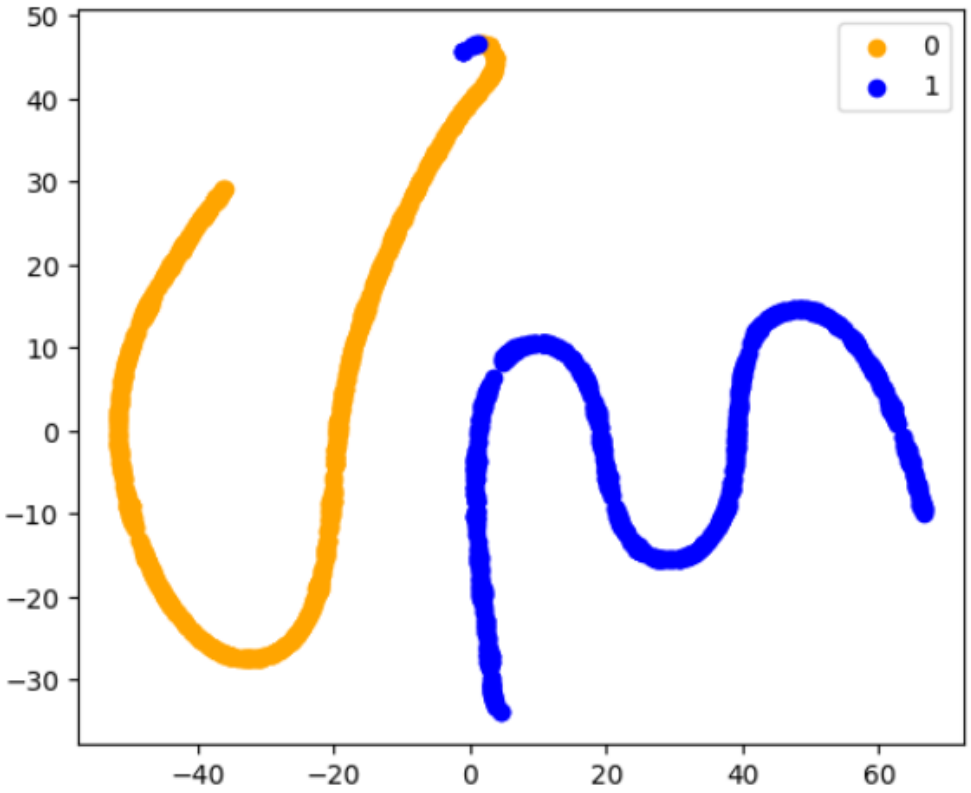

Figure 4.2: T-SNE RRN on ACP Alternate

and negative cases.

4.0.4 TSNE

Observations we can make are:

- The data points are clustered into two main groups, which presumably correspond to the acp and non-acp classes.
- There are some data points that are not clearly assigned to either group. These could be peptides that are misclassified by the RRN model, or they could be peptides that are borderline cases that are difficult to classify.
- The data points within each group are not randomly distributed. This suggests that there is some underlying structure in the data that the RRN model can capture.

Chapter 5

Conclusion

Recurrent Retention Networks (RRNs) are rewriting the rules of anti-cancer peptide prediction, leaving state-of-the-art methods like ACPred-LAF and AntiCP in the dust. Their remarkable performance, reaching a staggering 94.5% accuracy on ACP2 Main (shattering the previous SOTA of 81.8%), showcases their unparalleled ability to capture the complex long-range dependencies within peptide sequences that dictate their anti-cancer activity.

While this work firmly establishes RRNs as a game-changer in anti-cancer peptide prediction, exciting avenues remain to solidify their dominion and unlock even greater impact: Conquering Diverse Datasets: Pitting RRNs against a wider range of datasets, including those with unique challenges like membrane-bound or intrinsically disordered peptides, will rigorously test their generalizability and robustness.

Exploring Retention:

Delving deeper into alternative retention strategies, such as parallel or fragment-wise approaches, could unlock further accuracy and efficiency gains. This includes experimenting with various window sizes and activation functions to tailor RRNs for specific peptide types and tasks. Synergy with Structural Prediction Methods: Combining RRNs with complementary approaches like residue-residue potentials or co-evolution information holds im-

mense promise for generating even more accurate and comprehensive peptide structure models.

Fueling Anti-Cancer Peptide Design:

The precise contact predictions from RRNs can be harnessed to design novel peptides with preprogrammed anticancer functions and properties, potentially accelerating the development of life-saving therapeutics.

In conclusion, RRNs have ushered in a new era of anticancer peptide prediction, paving the way for a future where these powerful molecules can be designed with pinpoint precision to combat the devastating disease that is cancer. This is just the beginning, and the exciting journey of exploration and innovation fueled by RRNs holds the potential to revolutionize the field of anticancer therapeutics.

Appendices

References

[1] Phillip Compeau and Pavel Pevzner. *Bioinformatics Algorithms: An Active Learning Approach*. Online Book. National Center for Biotechnology Information (US), 2019. URL: https://www.ncbi.nlm.nih.gov/books/NBK562260/.

[2] Piyush Agrawal et al. "AntiCP 2.0: an updated model for predicting anticancer peptides". In: *Briefings in Bioinformatics* 22.3 (2021). DOI: 10.1093/bib/bbaa153.

[3] Wenjia He et al. "Learning embedding features based on multisense-scaled attention architecture to improve the predictive performance of anticancer peptides". In: *Bioinformatics* 37.24 (2021), pp. 4684–4693.

[4] Yifan Liu et al. "ACPred-BMF: bidirectional LSTM with multiple feature representations for explainable anticancer peptide prediction". In: *Scientific Reports* 12.1 (2022), pp. 1–12.

[5] Yunpeng Li et al. "Retentive Network: A Successor to Transformer for Large Language Models". In: (2023). arXiv: 2307.08621 [cs.CL].

[6] PyTorch Contributors. *PyTorch Documentation for torch.nn.LSTM*. Online documentation. Year of access. URL: https://pytorch.org/docs/stable/generated/torch.nn.LSTM.html.

[7] PyTorch Contributors. *PyTorch Documentation for torch.nn.RNN*. Online documentation. Year of access. URL: https://pytorch.org/docs/stable/generated/torch.nn.RNN.html.

Table 5.1: Project Detail

Student Details

Student Name	Pranshu Bahadur		
Registration Number	180953063	CCE	A/01
Email Address	pranshubahadur@gmail.com	+91 8690340104)	

Project Details

Project Title	A Novel Application of LPE and RetNet for identifying ACPs		
Project Duration	4 Months	Date of Reporting	21-08-2023

Organization Details

Organization Name	Manipal Institute of Technology
Full Postal Address	Manipal Institute of Technology, Manipal-576104
Website Address	https://manipal.edu/mit.html

Internal Guide Details

Faculty Name	Dr. Satyajit Mahapatra
Full Contact Address with PIN Code	Department of Information and Communication Technology, Manipal Institute of Technology, Manipal-576104
Email Address	satyajit.mahapatra@manipal.edu

CO and PO Mapping

CLOs		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
ICT 4299.1	Assess the work available in the literature related to the project to identify the limitations and risks.	3	3	-	3	3	-	3	3	3	3	3	3
ICT 4299.2	Practice planning and time management in solving the problem.	3	3	-	-	-	-	-	-	3	-	3	3
ICT 4299.3	Demonstrate professional skills to work effectively in a team or individually.	3	3	-	-	3	-	-	-	3	-	3	3
ICT 4299.4	Develop the ability to adopt a methodological approach to solve societal problems..	3	3	3	3	3	3	3	-	-	3	3	3
ICT 4299.5	Conduct experimentation and testing to achieve the defined objectives through computing/coding/statistical analysis	3	3	-	3	3	-	-	3	3	3	3	3
ICT 4299.6	Compose the technical report with effective communication on incorporating ethical practices.	3	3	3	3	3	3	3	3	3	3	3	3
ICT 4299 (Avg. correlation level)		3	3	3	3	3	3	3	3	3	3	3	3

PROGRAM OUTCOMES (PO)

Engineering Graduates will be able to:

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

CLOs		PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9
ICT 4299.1	Assess the work available in the literature related to the project to identify the limitations and risks.	3	3	3	3	3	3	3	3	-
ICT 4299.2	Practice planning and time management in solving the problem.	3	3	3	3	3	3	3	3	-
ICT 4299.3	Demonstrate professional skills to work effectively in a team or individually.	3	3	3	3	3	3	3	3	3
ICT 4299.4	Develop the ability to adopt a methodological approach to solve societal problems..	3	3	3	3	3	3	3	3	-
ICT 4299.5	Conduct experimentation and testing to achieve the defined objectives through computing/coding/statistical analysis	3	3	3	3	3	3	3	3	-
ICT 4299.6	Compose the technical report with effective communication on incorporating ethical practices.	3	3	3	3	3	3	3	3	3
ICT 4299 (Avg. correlation level)		3	3	3	3	3	3	3	3	3

1. To identify, analyse and develop software systems using appropriate techniques and concepts related to information technology
2. To design an algorithm or process within realistic constraints to meet the desired needs through analytical, logical and problem-solving skills.
3. To apply state of the art IT tools and technologies, IT infrastructure management abilities in treading innovative career path as a prospective IT engineer
4. Apply the principles of science, maths and computer programming to solve complex problems related to information technology.
5. Apply knowledge of programming, computational intelligence, computer graphics and visualization, data analytics, software system design, cyber security to arrive at solutions to real world problems.
6. Apply IT knowledge to design and develop systems with respect to societal, user, customer needs, health and safety, diversity, inclusion, societal, environmental codes of practise and industry standard.
7. Integrate and interface industry relevant hardware and software components and technology to come up with innovative and creative solutions.
8. Use of industry standard software tools and platform to design and analyze IT systems.
9. Learn to function collaboratively as a member of leader in diverse teams in multidisciplinary settings to manage the process effectively and document, present and communicate with the engineering community.

COURSE Code	Course Title	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	PSO9
ICT 4299	Project Work	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**

CLOs		C1	C2	C3	C4	C5	C6	C13	C16	C17
ICT 4299.1	Assess the work available in the literature related to the project to identify the limitations and risks.	3	3	3	3	3	-	3	3	3
ICT 4299.2	Practice planning and time management in solving the problem.	3	-	-	-	3	3	3	3	-
ICT 4299.3	Demonstrate professional skills to work effectively in a team or individually.	-	-	-	-	3	-	3	3	-
ICT 4299.4	Develop the ability to adopt a methodological approach to solve societal problems..	3	3	3	-	3	3	3	3	-
ICT 4299.5	Conduct experimentation and testing to achieve the defined objectives through computing/coding/statistical analysis	3	3	3	3	3	3	3	3	3
ICT 4299.6	Compose the technical report with effective communication on incorporating ethical practices.	3	3	3	3	3	3	3	3	3
ICT 4299 (Avg. correlation level)		3	3	3	3	3	3	3	3	3

Category	AHEP LO number	AHEP LO Statements
Science & Maths	C1	Apply knowledge of mathematics, statistics, natural science and engineering principles to the solution of complex problems. Some of the knowledge will be at the forefront of the particular subject of study
Engineering Analysis	C2	Analyse complex problems to reach substantiated conclusions using first principles of mathematics, statistics, natural science and engineering principles
	C3	Select and apply appropriate computational and analytical techniques to model complex problems, recognising the limitations of the techniques employed
	C4	Select and evaluate technical literature and other sources of information to address complex problems
Design & Innovation	C5	Design solutions for complex problems that meet a combination of societal, user, business and customer needs as appropriate. This will involve consideration of applicable health & safety, diversity, inclusion, cultural, societal, environmental and commercial matters, codes of practice and industry standards
	C6	Apply an integrated or systems approach to the solution of complex problems
The Engineer & Society	C7	Evaluate the environmental and societal impact of solutions to complex problems and minimise adverse impacts
	C8	Identify and analyse ethical concerns and make reasoned ethical choices informed by professional codes of conduct
	C9	Use a risk management process to identify, evaluate and mitigate risks (the effects of uncertainty) associated with a particular project or activity
	C10	Adopt a holistic and proportionate approach to the mitigation of security risks
	C11	Adopt an inclusive approach to engineering practice and recognise the responsibilities, benefits and importance of supporting equality, diversity and inclusion
Engineering Practice	C12	Use practical laboratory and workshop skills to investigate complex problems
	C13	Select and apply appropriate materials, equipment, engineering technologies and processes, recognising their limitations
	C14	Discuss the role of quality management systems and continuous improvement in the context of complex problems
	C15	Apply knowledge of engineering management principles, commercial context, project and change management, and relevant legal matters including intellectual property rights
	C16	Function effectively as an individual, and as a member or leader of a team
	C17	Communicate effectively on complex engineering matters with technical and non-technical audiences
	C18	Plan and record self-learning and development as the foundation for lifelong learning/CPD