S.No	Step	Method	AI/ML Integration
1	Peptide Generation	240 peptides predicted using **RFDiffusion**, **ProteinMPNN**, **AlphaFold2**	Use GANs or RL-based models to generate novel peptides with optimized properties
2	Molecular Docking	Docking with Class A β-lactamases (KPC-2, CTX-M-1, SHV-1, SME-1, TEM-1)	Train an ML model (Random Forest/XGBoost) on docking scores to predict binding affinity for new peptides
3	Molecular Dynamics (MD) Simulation	Ran simulations for all docked peptides	Automated the steps of pre-processing to run the simulation for 20ns
4	Peptide Classification	To be performed	Use CNN/LSTM/Transformers (AMP-CLIP-like models) to classify peptides as β-lactamase inhibitors vs. non-inhibitors
5	MIC Regression Model	To be performed	Use XGBoost/Hybrid Regression to predict MIC values and refine peptide selection
6	Peptide Optimization	To be performed	Apply Reinforcement Learning (RL-AMPs) + Bayesian Optimization to improve peptide sequences
7	Explainability	To be performed	Use SHAP, DeepProtein Editing (DPE) to identify key residues for inhibition
8	Experimental Validation	Final step (wet lab validation)	Select peptides with lowest MIC & best docking scores for in vitro testing

Brief description of predicted peptides

Target	Number of peptides generated
BLIP-I	10
Omega Loop	110
Class A beta lactamase	120
Total	240

Number of peptides generated for each target.

MIC regression model

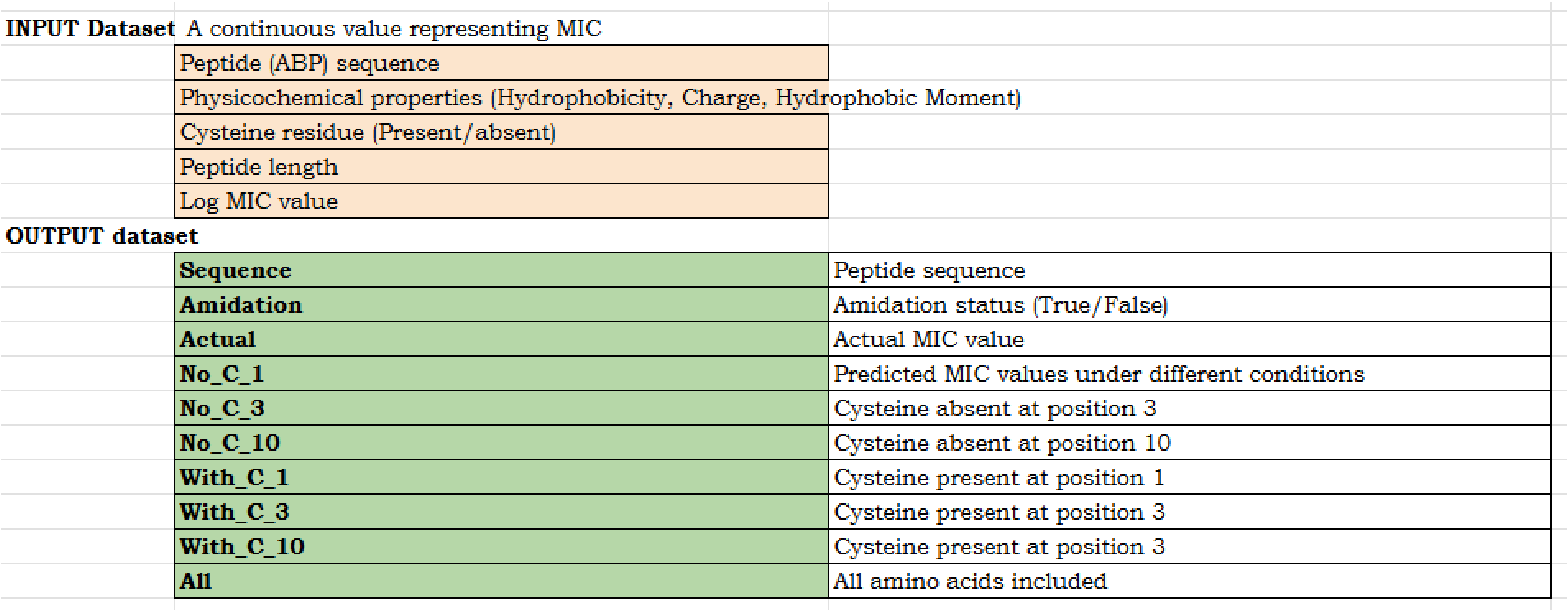
Paper: https://www.biorxiv.org/content/10.1101/692681v1.full

Github: https://github.com/zswitten/Antimicrobial-Peptides

Dataset: GRAMPA: Giant Repository of AMP Activity
(https://github.com/zswitten/Antimicrobial-Peptides)

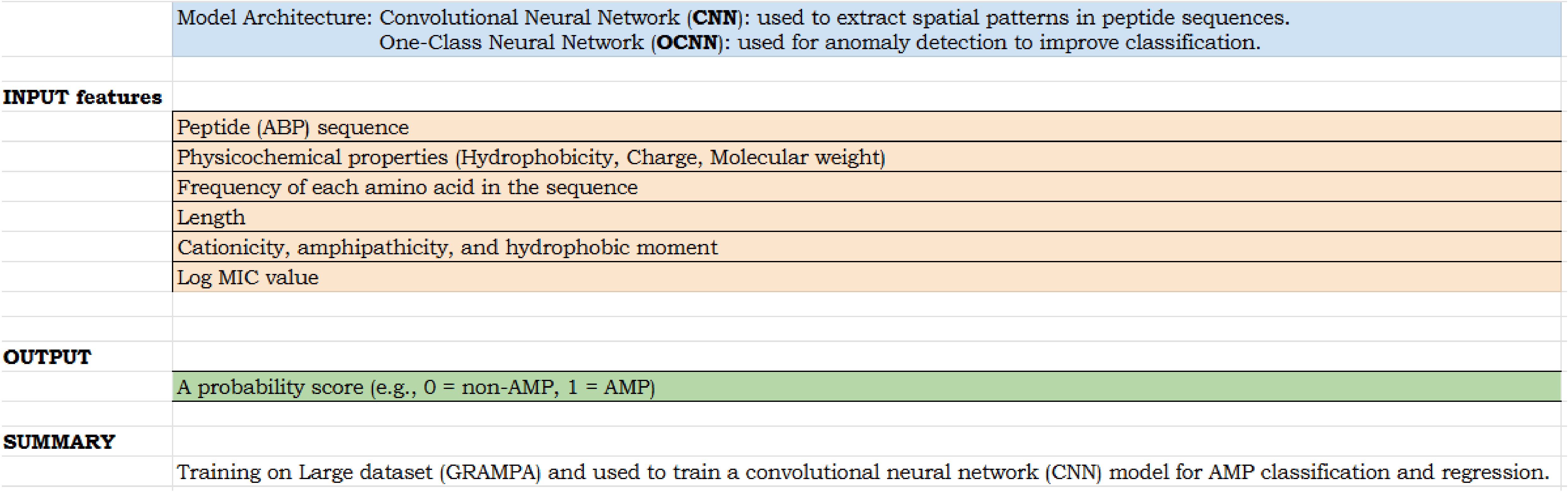
```
###############__Installation__###############
sudo apt update
sudo apt install build-essential libssl-dev libffi-dev python3-dev
git clone https://github.com/zswitten/Antimicrobial-Peptides.git
cd Antimicrobial-Peptides
conda create --name AMP_env python=3.8
conda activate AMP_env
pip install -r requirements.txt
python src/train_model.py --negatives=1 --bacterium='E. coli' --epochs=60
```

MIC regression model



INPUT Dataset	A continuous value representing MIC	
	Peptide (ABP) sequence	
	Physicochemical properties (Hydrophobicity, Charge, Hydrophobic Moment)	
	Cysteine residue (Present/absent)	
	Peptide length	
	Log MIC value	
OUTPUT dataset		
	Sequence	Peptide sequence
	Amidation	Amidation status (True/False)
	Actual	Actual MIC value
	No_C_1	Predicted MIC values under different conditions
	No_C_3	Cysteine absent at position 3
	No_C_10	Cysteine absent at position 10
	With_C_1	Cysteine present at position 1
	With_C_3	Cysteine present at position 3
	With_C_10	Cysteine present at position 3
	All	All amino acids included

AMP and Non-AMP classification model



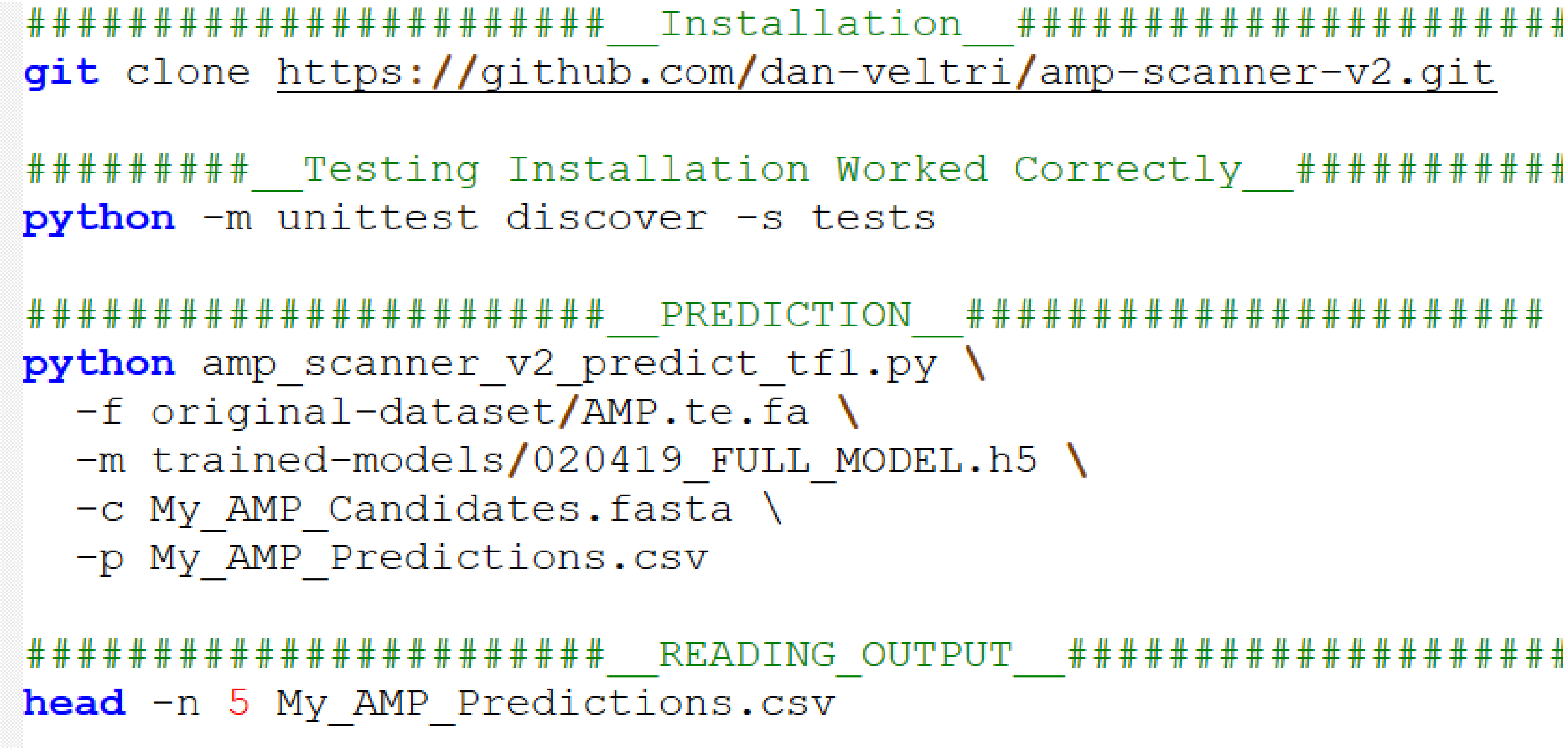
	Model Architecture: Convolutional Neural Network (**CNN**): used to extract spatial patterns in peptide sequences. One-Class Neural Network (**OCNN**): used for anomaly detection to improve classification.
INPUT features	
	Peptide (ABP) sequence
	Physicochemical properties (Hydrophobicity, Charge, Molecular weight)
	Frequency of each amino acid in the sequence
	Length
	Cationicity, amphipathicity, and hydrophobic moment
	Log MIC value
OUTPUT	
	A probability score (e.g., 0 = non-AMP, 1 = AMP)
SUMMARY	
	Training on Large dataset (GRAMPA) and used to train a convolutional neural network (CNN) model for AMP classification and regression.

Case study2: AMP scanner v2

Paper: https://academic.oup.com/bioinformatics/article/34/16/2740/4953367?login=true

Github: https://github.com/dan-veltri/amp-scanner-v2

Dataset: Antimicrobial Peptide Database (**APD**) version 3 (https://aps.unmc.edu/downloads)



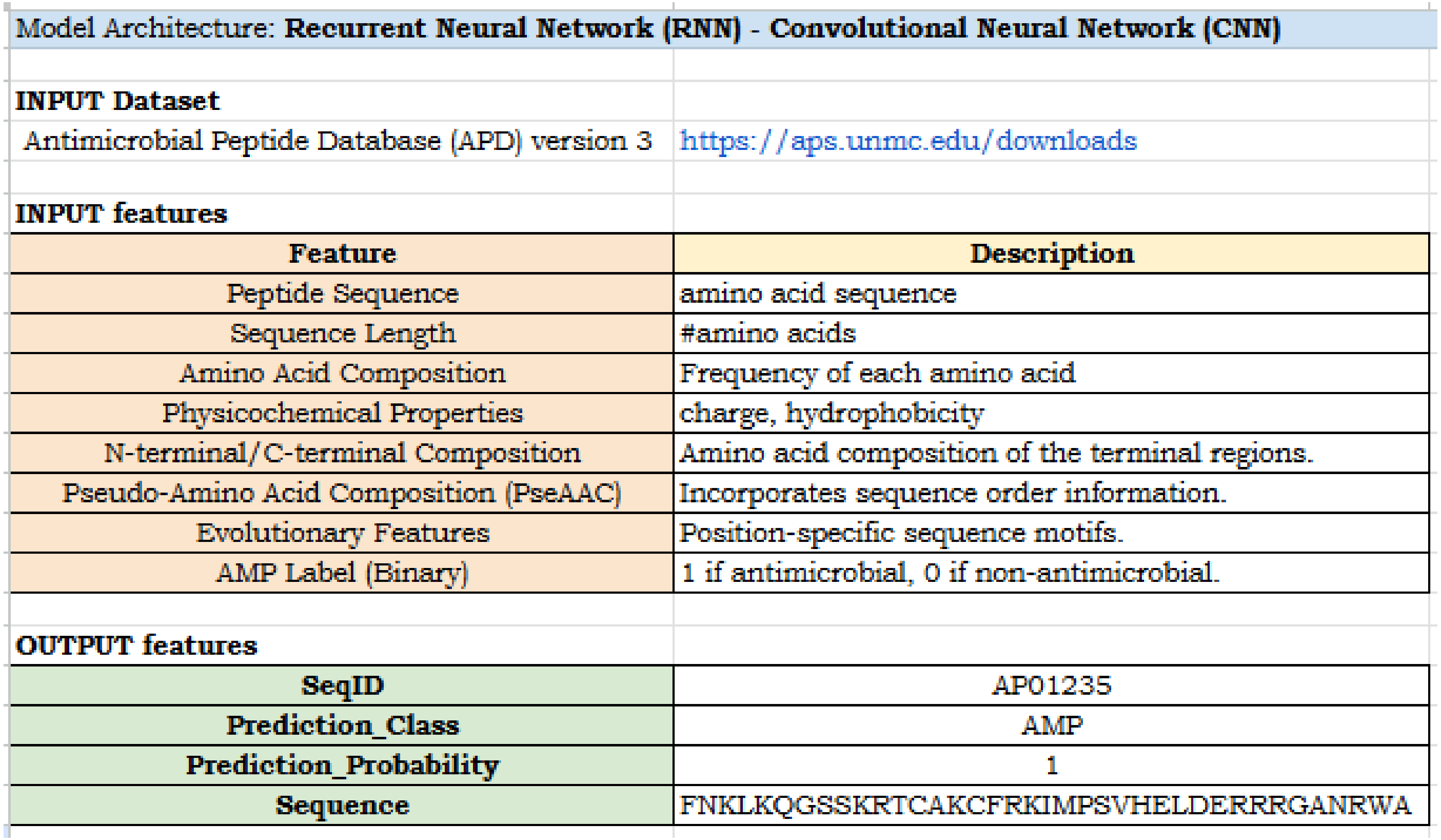
```
#######################__Installation__######################
git clone https://github.com/dan-veltri/amp-scanner-v2.git

#########__Testing Installation Worked Correctly__###########
python -m unittest discover -s tests

#######################__PREDICTION__#######################
python amp_scanner_v2_predict_tf1.py \
  -f original-dataset/AMP.te.fa \
  -m trained-models/020419_FULL_MODEL.h5 \
  -c My_AMP_Candidates.fasta \
  -p My_AMP_Predictions.csv

#######################__READING_OUTPUT__####################
head -n 5 My_AMP_Predictions.csv
```

AMP scanner v2



Model Architecture: **Recurrent Neural Network (RNN) - Convolutional Neural Network (CNN)**

INPUT Dataset

Antimicrobial Peptide Database (APD) version 3	https://aps.unmc.edu/downloads

INPUT features

Feature	Description
Peptide Sequence	amino acid sequence
Sequence Length	#amino acids
Amino Acid Composition	Frequency of each amino acid
Physicochemical Properties	charge, hydrophobicity
N-terminal/C-terminal Composition	Amino acid composition of the terminal regions.
Pseudo-Amino Acid Composition (PseAAC)	Incorporates sequence order information.
Evolutionary Features	Position-specific sequence motifs.
AMP Label (Binary)	1 if antimicrobial, 0 if non-antimicrobial.

OUTPUT features

SeqID	AP01235
Prediction_Class	AMP
Prediction_Probability	1
Sequence	FNKLKQGSSKRTCAKCFRKIMPSVHELDERRRGANRWA

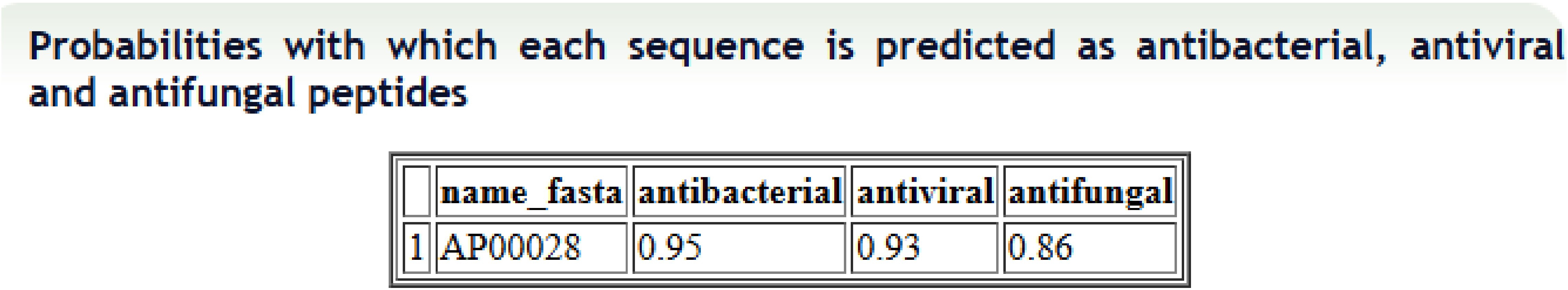
Case study 3: iAMPpred

Paper: https://www.nature.com/articles/srep42362#Sec2

Online accessible server: http://cabgrid.res.in:8080/amppred/

The model aims to classify peptides as **antibacterial, antiviral, or antifungal**

Output on server:



Probabilities with which each sequence is predicted as antibacterial, antiviral and antifungal peptides

	name_fasta	antibacterial	antiviral	antifungal
1	AP00028	0.95	0.93	0.86

AMP scanner v2

Input sequence from APD3 database:

>AP00028

CLGIGSCNDFAGCGYAVVCFW

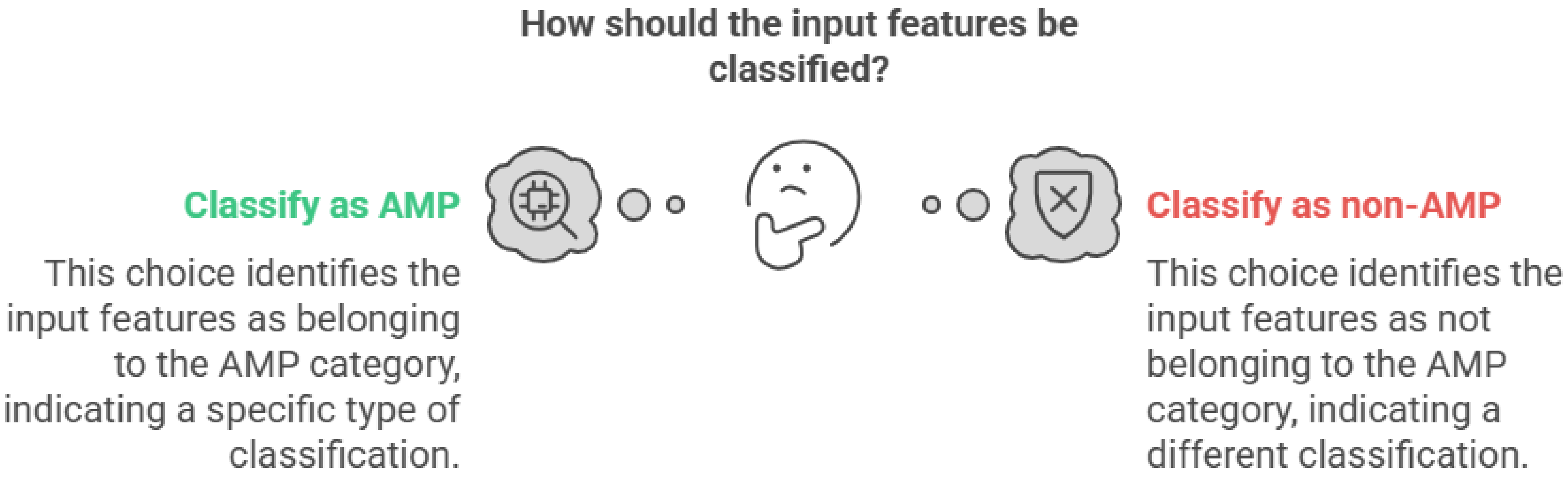
INPUT FEATURES:

compositional and PHYC features were computed by using the “Peptide” package of R-software, whereas the STRL features were computed by using the TANGO software available at http://tango.crg.es/.

Feature category	Features in each category	#Features
Compositional	Amino acid composition (AAC)	20
	Normalized AAC (NAAC)	20
Structural (STRL)	Pseudo AAC (PAAC)	20
	α-helix propensity	1
	β-sheet propensity	1
	Turn propensity	1
Physico-chemical (PHYC)	Iso-electric point	1
	Hydrophobicity	1
	Net-charge	1

Model Architecture

- Support Vector Machine
- **RBF Kernel**: Captured non-linear patterns in AMPs.
- Accuracy: 94.4%
- The model was trained for high-dimensional data a total of 66 features.



How should the input features be classified?

Classify as AMP

This choice identifies the input features as belonging to the AMP category, indicating a specific type of classification.

Classify as non-AMP

This choice identifies the input features as not belonging to the AMP category, indicating a different classification.

- The **model does not give a single label** but rather a probability distribution, meaning a peptide can belong to multiple categories with different confidence levels.
- **Multi-functional peptides** can have high probabilities in multiple categories, indicating **broad-spectrum antimicrobial activity**.

Dataset:

The value inside bracket {} is the number of sequences collected in that category.

Dataset	Bacterial	Viral	Fungal
Positive	CAMP, APD3, AntiBP2 {3417}	CAMP, APD3, LAMP, AVPpred {739}	CAMP, LAMP, APD3 {1496}
Negative	AntiBP2 {984}	AVPpred {893}	AntiBP2, AVPpred {1384}

Input Features for AMP Classification Model

Feature	**Description**
Peptide Sequence	The amino acid sequence of the peptide.
Sequence Length	The total number of amino acids in the sequence.
Amino Acid Composition	Frequency of each amino acid in the sequence.
Physicochemical Properties	Includes charge, hydrophobicity, isoelectric point, etc.
N-terminal/C-terminal Composition	Amino acid composition of the terminal regions.
Pseudo-Amino Acid Composition (PseAAC)	Incorporates sequence order information.
Evolutionary Features	Position-specific sequence motifs.
AMP Label (Binary)	1 if antimicrobial, 0 if non-antimicrobial.