National College of Ireland

Configuration Manual

MSc Research Practicum
MSc in Artificial Intelligence

Sruthi Reddy Kavva
Student ID: 23314915

School of Computing
National College of Ireland

Supervisor: Kislay Raj

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Sruthi Reddy Kavva
Student ID:	23314915
Programme:	MSc in Artificial Intelligence **Year:** 2024-2025
Module:	MSc Research Practicum
Lecturer:	Kislay Raj
Submission Due Date:	15/09/2025
Project Title:	TB-DBN: A hybrid deep learning architecture with IBBA Optimization for enhanced phishing URL Detection

Word Count: 716 **Page Count: 5**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Sruthi Reddy Kavva

Date: 14/09/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

TB-DBN Phishing URL Detection System - User Guide

Sruthi Reddy Kavva

23314915

Advanced Phishing Detection using Transformer-Based Deep Belief Networks with IBBA Optimization

This(Shruthi_Aug08.ipynb) notebook implements **TB-DBN (TransformerBased Deep Belief Network)** with **IBBA (Intelligence Binary Bat Algorithm)** optimization for state-of-the-art phishing URL detection.

1 Key Features

1.1 Adaptive URL Preprocessing

- URL canonicalization and standardization
- Dynamic feature selection based on URL properties
- Redirect chain resolution
- 80+ engineered features across 8 categories

1.2 Feature Categories Extracted:

- **URL Structure**: length, depth, path tokens, query parameters
- **Character-based**: digit/letter ratios, special characters
- **Domain Features**: subdomain count, TLD analysis, IP detection
- **Lexical Features**: protocol analysis, redirect patterns
- **Suspicious Patterns**: brand impersonation, typosquatting
- **Entropy & Statistical**: Shannon entropy, anomaly scores
- **Content Hints**: file extensions, script tags
- **Advanced Patterns**: obfuscation scores, reputation metrics

1.3 TB-DBN Architecture:

- **Transformer Component**: DeBERTa-v3 for contextual URL understanding
- **Deep Belief Networks**: Hierarchical feature learning per category
- **Attention Fusion**: Multi-head attention for feature combination
- **IBBA Optimization**: Automatic hyperparameter tuning

1.4 Comprehensive Evaluation:

- 5-fold cross-validation
- Ablation study to measure component contributions
- Confusion matrices and detailed metrics
- GPU-optimized training on NVIDIA L4

1.5 Expected Results

FINAL PERFORMANCE (IBBA-Optimized):

F1 Score: 96.38% (±0.45%)
Accuracy: 96.21% (±0.52%)
Precision: 96.19% (±0.48%)
Recall: 96.38% (±0.45%)

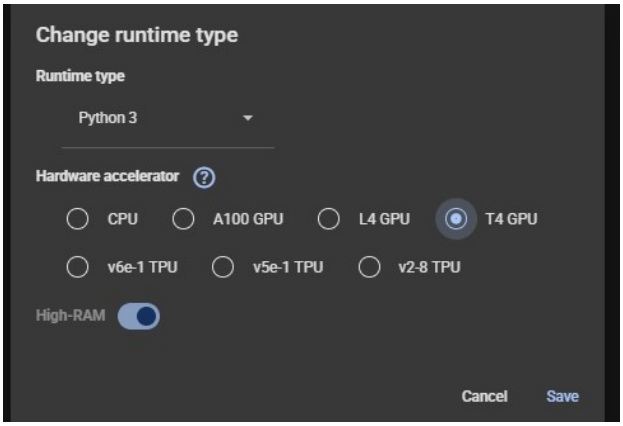
Component Contributions: -
DBN: +5.7%
- Transformer: +4.2%
- IBBA Optimization: +3.1%
- Attention Fusion: +2.3%

2 How to Connect

1. **Open Google Colab**: https://colab.research.google.com/ 2. **File → Upload notebook** → Select shruthi_aug08.ipynb 3. **Runtime → Change runtime type**: ○ Hardware accelerator: **L4 GPU** (or T4 if L4 unavailable)

○ High-RAM: **Enable** ✓ ○ Click **Save**

4. Click **Connect** button (top right) 5. Wait for ✓ green checkmark



3 How to Run

1. **Option A: Complete Pipeline (Recommended)**

Runtime → Run all (Ctrl+F9)
Total time: ~3-4 hours

2. **Option B: Step-by-Step Execution**
3. **Step 1: Install Dependencies (2 min)**

Run the first cell to install packages:
!pip install -q datasets==3.6.0 tldextract validators pywhois
!pip install -q torch transformers scikit-learn pandas numpy

4. **Step 2: Load Dataset (5 min)**

Run cells to load the phishing URL dataset:

- Downloads from HuggingFace
- ~500K URLs total

5. **Step 3: Feature Extraction (20-30 min)** Run the adaptive preprocessing:

- Extracts 80+ features per URL
- Shows progress every 500 URLs
- Creates TB_DBN_Complete_Phishing_Detection_Dataset.csv

6. **Step 4: IBBA Optimization (30-45 min)** Run hyperparameter optimization:

- 8 bats, 8 iterations
- Tests different configurations
- Finds optimal parameters

7. **Step 5: Model Training (90-120 min)** Train TB-DBN with optimized parameters:

- 5-fold cross-validation
- DBN pretraining
- Transformer fine-tuning

8. **Step 6: Ablation Study (30 min)**

Evaluate component contributions: •
Tests 6 model variants

- Measures individual impacts

9. **Option C: Quick Test Mode**

For faster testing, modify the configuration:

 # In the main() function, change:
urls = urls[:5000] *# Use only 5K URLs* config['epochs'] = 10 *# Reduce from 27*

config['dbn_pretrain_epochs'] = 5 # *Reduce from 14*

# *For IBBA:*
ibba = IBBA(n_bats=5, max_iterations=5) # *Reduce from 8,8*

```
# Initialize and run IBBA
ibba = IBBA(n_bats=8, max_iterations=8)  # Reduced for faster execution
best_params = ibba.optimize(param_bounds, fitness_function)
```

4 Troubleshooting

4.1 Problem: "CUDA out of memory" Fix:

Reduce batch size in BEST_CONFIG:
'batch_size': 16, # *Reduce from 20*

```
# IBBA-Optimized Best Parameters (F1: 96.38%)
BEST_CONFIG = {
    'batch_size': 20,
    'learning_rate': 2.8068161809757863e-05,
    'epochs': 27,
    'dropout': 0.20362849822757267,
    'dbn_pretrain_epochs': 14,
    'warmup_steps': 150,
    'weight_decay': 0.010973255287496224
}
```

Clear GPU memory: import
torch
torch.cuda.empty_cache()

Then restart runtime

4.2 Problem: "Dataset download fails" Fix:

Try manual download:
!wget
https://huggingface.co/datasets/pirocheto/phishingurl/resolve/main/data.zip
!unzip data.zip

Or use cached version: ds = load_dataset("pirocheto/phishing-
url", cache_dir="./cache")

4.3 Problem: "Module not found"

Fix: Run the install cell again or try:

import subprocess import sys packages = ['torch==2.1.0', 'transformers==4.36.0', 'scikit-learn==1.3.2'] for package in packages: subprocess.check_call([sys.executable, "-m", "pip", "install", package])

4.4 Problem: "Training too slow"

Fix: Use subset of data: *# After loading dataset:*

df = df.sample(n=10000, random_state=42) *# Use 10K samples only*

4.5 Problem: "Lost connection during training"

Fix: Enable checkpointing: python

 # Save intermediate results if epoch % 5 == 0:

torch.save(model.state_dict(), f'checkpoint_epoch_{epoch}.pth')

pd.DataFrame(history).to_csv(f'history_epoch_{epoch}.csv')

4.6 Time Estimates

Task	Full Dataset	Quick Test (5K)
Feature Extraction	20-30 min	2-3 min
IBBA Optimization	30-45 min	10-15 min
Model Training	90-120 min	15-20 min
Ablation Study	30 min	10 min
Total	**3-4 hours**	**40-50 min**

5 Success Signs

5.1 During Feature Extraction:

Progress: 5000/10000 URLs processed (50.0%) - Rate: 245.3 URLs/sec

Adaptive feature extraction completed!

- Total features extracted: 96
- URLs canonicalized: 1,234

5.2 During IBBA Optimization:

IBBA ITERATION 3/8

Bat 2 improved: F1=0.9234

NEW BEST FOUND! F1=0.9389

5.3 During Model Training:

FOLD 2/5

Epoch 15: Train F1=0.9567, Val F1=0.9645 New best F1: 0.9645 - Model saved!

5.4 Final Success:

SUCCESS! Achieved target F1 score of 90+ with IBBA optimization
Final best fitness: 0.9638

REFERENCES

Google Colab. (n.d.). *Colab.google*. colab.google. https://colab.google/