Project Report: Contributor Verification using Robust Deep Neural Networks

January 8, 2026

Abstract

This report details the development of machine learning models for the Contributor Verification challenge, a critical binary classification task. We systematically explored and critically analyzed multiple Deep Neural Network (DNN) architectures and an Ensemble method. The final selected model, **Final_model_Robust**, was evolved with regularization techniques to maximize generalization. This model secured a Public Leaderboard **Performance Score of 0.87500** and a Private Leaderboard **Performance Score of 0.86071**, outperforming alternative approaches by integrating sparse categorical features with complex textual representations while minimizing overfitting.

1 Introduction

1.1 Problem Context and Objective

The big idea behind this project was simple: **We had to build a system that could tell us, for sure, if a person actually helped write an article.** In the world of online publishing, verifying who did what is critically important. So, our job was to solve a **Yes/No problem** (a "supervised binary classification task"): Is this candidate a contributor? Yes or No. We started with a dataset containing **25,000 articles** for training and had to test them on $\mathbf{2,000}$ brand-new articles. Our main goal was to get the highest **Classification Performance Score** on the final, secret test set (the Private Score). That's why we ultimately chose the **Final_model_Robust**—it was the best at handling the unexpected and giving us reliable predictions.

1.2 Data Preprocessing and Sample Strategy

The article-centric data was transformed into **(article, candidate, label)** samples. This involved generating positive and negative training instances:

- **Positive Samples ($Y = 1$):** Every actual contributor pairing, totaling **44,170 samples**.
- **Negative Samples ($Y = 0$):** An equal number of non-contributor candidates were randomly sampled, resulting in a final balanced training set size of **88,340 pairs**. This balanced approach was key to stabilizing training.

1.3 Feature Engineering

Feature processing utilized the Keras Functional API, with specific layers for each data type:

The text sequence was summarized using **Global Average Pooling (GAP)** after embedding, converting the variable-length textual input into a fixed-size vector.

Table 1: Feature Processing and Intrinsic Dimensions

Feature	Processing Method	Intrinsic Detail
Candidate ID	Embedding Layer	Vocabulary size: **2,302 unique candidates**
Text	Embedding + Global Average Pooling	Sequence length up to **500** tokens
Article Year	Input Layer (Normalized)	Single scalar input (Year is normalized)

2 Methodology: Deep Neural Network Architecture

2.1 Final_model_Robust DNN

The Final_model_Robust is a multi-branch DNN with aggressive regularization hyperparameters to minimize generalization error.

2.1.1 Model Structure and Hyperparameters

The architecture consisted of a concatenated feature vector passed through three deep hidden layers:

- **Hidden Layer 1:** Dense layer with **256** neurons, ReLU activation, subjected to **L2 Regularization** ($\lambda \approx 0.001$).
- **Hidden Layer 2:** Dense layer with **128** neurons, ReLU activation, followed by a **Dropout rate of 0.3**.
- **Hidden Layer 3:** Dense layer with **64** neurons, ReLU activation, followed by a **Dropout rate of 0.2**.

The model was optimized using the **Adam optimizer** and trained with **Binary Cross-Entropy** loss, employing **Early Stopping** based on validation loss.

2.2 Alternative Architecture: Ensemble DNN_XGBoost

This approach attempted to combine the predictions of a DNN with an XGBoost Classifier, utilizing specific parameters for the tree-based component: $n\_estimators = \mathbf{100}$ and $max\_depth = \mathbf{5}$.

- **Blending Method:** The ensemble utilized a simple, unweighted arithmetic average of the predicted probabilities:

$$\mathrm{P}_{\text{ensemble}} = \frac{\mathrm{P}_{\text{DNN}} + \mathrm{P}_{\text{XGBoost}}}{2}$$

  This **0.5/0.5** weighting scheme proved to be a critical limitation.

3 Comparative Analysis and Conclusion

3.1 Model Performance Summary

Performance was evaluated based on Validation Accuracy and the final leaderboard scores.

Table 2: Model Performance Comparison (Leaderboard Scores)

Model Name	Val Acc	Public Score	Private Score	Drop (Val Acc to Private Score)
last	**0.8982**	0.86660	0.84785	**−5.60%**
DNN_Baseline	0.8870	0.87000	0.85571	−3.53%
Ensemble	0.8783	NA	NA	(Lowest Validation)
Robust DNN	0.8822	**0.87500**	**0.86071**	**−2.43%**

3.2 Critical Analysis and Rationale for Selection

The **Final_model_Robust** was selected based on its superior **Private Score (0.86071)** and the minimal generalization gap ($-$**2.43**%).

- **Failure of Overfitting (The `last` Model):** The high Validation Accuracy of **0.8982** for the `last` model masked its severe **overfitting**, resulting in a **5.60**% performance drop on the final test set.
- **Success of Robustness:** The Final_model_Robust's specific regularization (Dropout **0.3**/**0.2** and L2) successfully minimized the generalization gap. This proved its superior capacity to learn true, stable contribution patterns.
- **Ensemble Limitation:** The Ensemble approach failed due to its low Validation Accuracy (**0.8783**) and the basic **0.5**/**0.5** averaging scheme, which was not effective in combining the predictions of the DNN and the shallow (**max_depth** $=$ **5**) XGBoost model.

3.2.1 Strengths and Weaknesses

- **Strengths:** The targeted regularization successfully improved generalization. The balanced sampling helped the regularized model fight potential class imbalance effectively.
- **Weaknesses:** Training the deep multi-input DNN was computationally expensive, taking approximately **11** seconds per epoch. This high training time limited advanced hyperparameter tuning.

3.3 Conclusion

The Contributor Verification project successfully completed a specialized Deep Neural Network. The **Final_model_Robust** effectively synthesized complex multi-modal features and, through targeted regularization, achieved the highest Private Score of **0.86071**.