BRAUDE
College of Engineering, Karmiel

Software Engineering Department

Capstone Project Phase B – 61999

Analysis of Medieval Arabic Creations

25-1-R-19

https://github.com/YosraDaso/CapstoneProject.git

User Guide

Maintenance Guide

Yosra Fhamne	204138879	yosra.fhamne@e.braude.ac.il
Omar Saleh	323004895	omar.saleh@e.braude.ac.il

Supervisors:

Prof. Zeev Volkovich

Dr. Renata Avros

Table of Contents

1. Introduction

Authorship attribution in medieval Arabic texts presents unique linguistic and computational challenges. Rich morphological structure, stylistic variation, and non-standard orthography in historical manuscripts make it difficult to determine whether a given work was genuinely authored by the individual to whom it is attributed. These issues are especially critical in the case of Abu Hamid Al-Ghazali, a central figure in Islamic thought, whose corpus includes numerous texts of disputed authenticity.

Traditional methods for authorship verification, relying on manual stylistic analysis, lack the objectivity and scalability required for consistent evaluation across large, diverse collections. This has created a need for automated tools that can offer reproducible, data-driven insight into textual authorship.

The goal of this project is to design a computational framework capable of distinguishing authentic from pseudo-attributed texts, even in the absence of labeled training data. The system must be scalable to large historical corpora, adaptable to low-resource, morphologically rich languages like classical Arabic, and interpretable enough to support scholarly insight. Although the project focuses on Al-Ghazali's writings, the broader aim is to develop a methodology that generalizes across authors and eras.

1.1 Scope of This Document

This document presents the second phase of the project, focused on the practical implementation and evaluation of the proposed authorship attribution framework. It builds upon the theoretical foundation established in Phase A and details the process of translating the conceptual model into a functioning system.

The following chapters describe the datasets used, the core algorithmic pipeline, experimental procedures, and results. While the primary case study centers on Al-Ghazali’s corpus, the system was built to serve as a general tool for computational authorship analysis.

This work is intended for scholars, researchers, and practitioners in the fields of digital humanities, Arabic literary studies, and computational linguistics. It offers a replicable methodology and insights for those engaged in the analysis, preservation, or attribution of historical texts.

2. Proposed Solution

This section presents the authorship verification approach developed for this project, which is based on deep neural models, Arabic-specific preprocessing, and unsupervised stylistic signal comparison. The system operates in two main phases: training and classification. During training, a Siamese neural network is trained on pairs of texts from a diverse impostor group using contrastive loss to learn stylistic similarity. In the classification phase, the model is used to compute similarity-based signals for each test text, which are then analyzed to detect stylistic anomalies.

The proposed solution follows a modular architecture and is composed of several key stages:

- Text preprocessing tailored to Arabic language characteristics.
- Embedding generation using AraBERT, a pre-trained transformer model for Arabic.
- A Siamese neural network combining CNN and BiLSTM layers to model pairwise stylistic distance.

- Signal generation and aggregation.
- Anomaly detection using Dynamic Time Warping (DTW) and Isolation Forest.
- Clustering based on anomaly vectors to identify stylistic groupings.

Figure 1 presents an overview of the complete algorithmic workflow. It highlights the transition from raw Arabic texts to signal-based clustering through preprocessing, embedding, similarity scoring, and anomaly analysis.

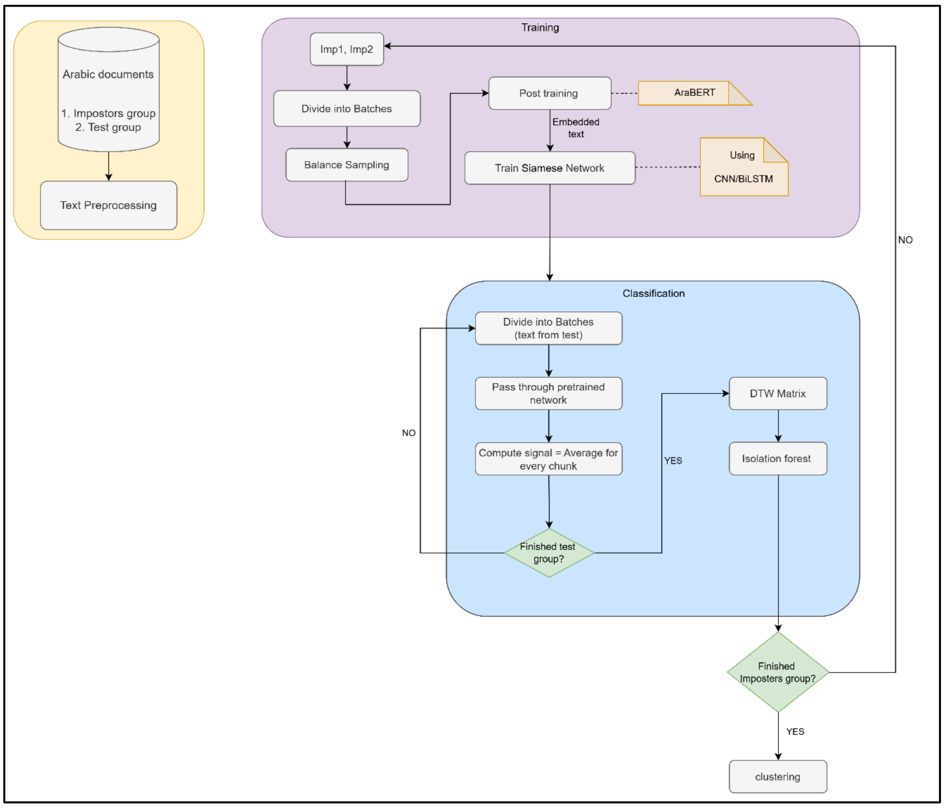

Figure 1. Algorithm explanation diagram.

To train the model to distinguish between writing styles, we employ a Siamese neural network at the core of the training pipeline. This architecture is specifically designed for pairwise comparison and enables the system to learn a representation of stylistic similarity. By processing pairs of embedded texts through identical subnetworks with shared weights, the Siamese model learns to minimize the distance between similar texts and maximize it for dissimilar ones using contrastive loss [1].

The Siamese Network used in this project combines convolutional and recurrent components to extract meaningful stylistic features from Arabic text. The convolutional (CNN) layers identify local stylistic cues within the embedded text, such as lexical patterns or phrase-level features. These are then passed to a bidirectional LSTM (BiLSTM) layer, which captures sequential dependencies and broader contextual relationships by processing the information in both forward and backward directions. This hybrid architecture allows the model to encode both fine-grained and holistic stylistic characteristics of the text.

Figure 2 presents the architecture of the Siamese Network and the role of contrastive loss in training the model to compare text pairs effectively.

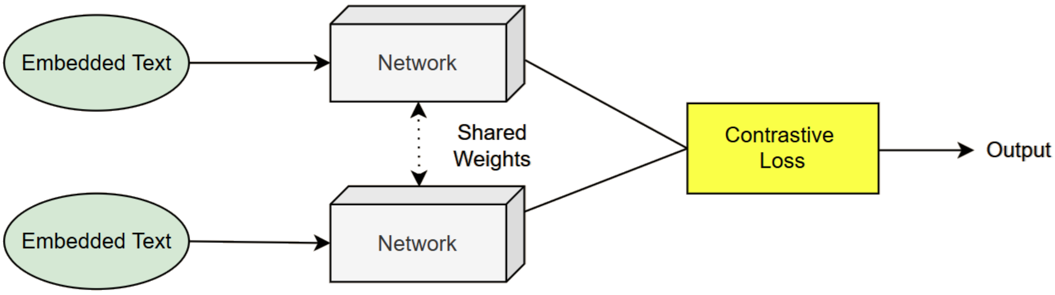

Figure 2. Siamese Neural Network Architecture.

To complement this high-level overview, Figure 3 provides pseudocode for the full process. This includes training with impostor pairs, signal extraction for the test group, and clustering of final anomaly vectors.

```
Algorithm 1 Impostor-Based Authorship Verification
Input: Impostor_Group, Test_Group
Output: clusters of authentic vs. pseudo-Ghazali texts
(Impostor_Group, Test_Group) ← preprocess_all(Impostor_Group, Test_Group)
Impostor_Group ← balance(Impostor_Group)
impostor_pairs ← all unique pairs from Impostor_Group
test_signals ← ∅
for each (imp1, imp2) in impostor_pairs do
    imp1.batches ← batch(imp1)
    imp2.batches ← batch(imp2)
    emb1 ← AraBERT(imp1.bal)
    emb2 ← AraBERT(imp2.bal)
    model ← Siamese(CNN + BiLSTM)
    train(model, emb1, emb2, contrastive_loss)
    for each test_text in Test_Texts do
        batches ← batch(test_text)
        emb ← AraBERT(batches)
        scores ← ∅
        for each chunk in chunkify(emb) do
            score ← model.similarity(mean(chunk))
            scores ← scores ∪ {score}
        end for
        test_signals[test_text] ← scores
    end for
end for
dtw_matrix ← DTW(test_signals)
outliers ← IsolationForest(dtw_matrix)
clusters ← KMeans(test_signals)
return clusters
```

Figure 3. Pseudocode of the impostor-based authorship verification pipeline.

The following subsections describe the experiment setup and dataset used for implementation and evaluation.

2.1 Experiment Setup

The entire system was implemented in Python 3.7.6, using the PyTorch deep learning framework for model construction and training. AraBERT v2 (aubmindlab/bert-base-arabertv2), accessed via the Transformers library (v4.28.1), served as the embedding model to capture contextual linguistic features tailored to Arabic.

Preprocessing steps, including tokenization and light stemming, were performed using CAMeL Tools (v1.5.5), a library specifically designed for Arabic NLP. Additional libraries such as scikit-learn, fastdtw, and datasets were used for signal generation, anomaly detection, clustering, and data management.

Development and experimentation were primarily conducted in Google Colab Pro+ using an A100 GPU with high-RAM configuration. This setup enabled smooth execution of memory-intensive embedding and signal operations. For the training of the Siamese network, which required more computational power and extended runtime, we transitioned to Lambda Cloud, leveraging an NVIDIA GH200 GPU for high-throughput performance.

Table 1 summarizes the key configuration parameters and model architecture choices, including convolutional kernel sizes, hidden dimensions, dropout rates, and clustering strategy. The system was trained over 300 randomized iterations, and all texts were processed in standardized segment and chunk sizes to ensure consistency and comparability.

Table 1. Experimental Configuration and Model Parameters.

Component	Parameter
General Setup	Number of Iterations = 300
	Impostor Group Size = 25
	Test Group Size = 113
	Word Embedding Model = AraBERTv2 (aubmindlab/bert-base-arabertv2)
	Max Token Length = 128
Siamese Network	Number of Kernels = 3
	1D Convolution Kernel Sizes = (3, 6, 12)
	Stride Size = 1
	Number of Feature Maps (Filters) = 300
	Fully Connected Layer Neurons = 600
	LSTM Layers = 2
	LSTM Hidden Size = 300
	LSTM Dropout = 0.25
	Output Size = 128-dimensional embeddings
	Activation Function = ReLU
	Optimizer = Adam
	Metric = Accuracy
	Learning Rate = 0.00001
	Number of Epochs = 5
	Cross-Validation Split = 0.25
Signal Processing	Segment Size = 50 words
	Chunk Size = 8 segments (i.e., 400 words)
Clustering	K = 2

2.2 Dataset

This study utilizes classical Arabic texts obtained from open-access digital libraries, including both works attributed to Al-Ghazali and texts by other historical figures. All materials were collected from reliable sources such as [2-3].

- Impostors' Group:

To construct the impostor group, we selected a diverse collection of authors from the same historical period as Al-Ghazali. The selection includes scholars who wrote in overlapping genres such as theology, philosophy, and ethics, as well as authors from other disciplines to introduce stylistic variability. Additionally, we included texts authored by Al-Ghazali’s students or intellectual contemporaries, as many misattributed works are believed to originate from this circle.

The impostor dataset consists of 25 books from the following authors:

- Abū Bakr al-Ṭurṭūshī (2 books)
- Abū Bakr ibn al-ʿArabī (2 books)
- Ibn Taymiyyah (3 books)
- Ibn Rushd (grandfather) (3 books)
- Ibn Sīnā (2 books)
- al-Juwaynī Abū al-Maʿālī (3 books)
- al-Fārābī (2 books)
- Ibn Khaldūn (3 books)
- Ibn Qudāmah al-Maqdisī (2 books)
- Ibn Rushd (grandson) (3 books)

Each book was segmented into chunks of approximately 250 KB to standardize input size and reduce bias related to text length. This dataset was used to generate 300 training iterations of impostor pairs for the Siamese network.

- Test Group:

The test set comprises 32 books attributed to Al-Ghazali, including both confirmed and disputed works (commonly referred to as Pseudo-Ghazali). Each book was segmented into parts of roughly 200 KB, resulting in a total of 113 test samples. These samples were passed through the trained model to compute similarity-based signals and evaluate authorship patterns.

3. Research Process

The research process began with collecting and preparing the dataset. We manually searched for classical Arabic books written by Al-Ghazali and a diverse group of historical authors from the same period. Our focus was on identifying sources that provided machine-readable .txt files, with sufficient variety in style and genre to support authorship attribution experiments. We then organized the books into two groups: test texts attributed to Al-Ghazali, and impostor texts from other authors.

Once the data was collected, we defined the architecture of our system. The proposed framework consists of five main stages:

1. Preprocessing the text data using Arabic-specific tools
2. Generating contextual embeddings using AraBERT
3. Comparing text segments using a Siamese CNN-BiLSTM network

4. Converting the output into signal form
5. Applying anomaly detection and clustering for final analysis

Each stage was implemented as a separate module in Python to allow flexibility and reusability. We selected Google Colab Pro+ as our primary development environment due to its GPU access and integration with Google Drive, which we used to manage datasets and intermediate outputs. To improve runtime efficiency, the system was designed to save and reuse embeddings and signals, minimizing repeated computation during experiments.

We implemented the Siamese model using PyTorch, relying on documentation and research articles to guide the structure of contrastive learning and embedding comparison. Throughout the development, we tested the system on small examples to validate each component before scaling up.

During experimentation, we adjusted key parameters, such as kernel sizes, learning rate, and hidden dimensions, based on validation feedback. We also tested different segment sizes to find the best input configuration for generating clean, informative signals.

Once the pipeline was integrated, we conducted multiple full runs, each producing a set of anomaly vectors used for clustering. The outputs were analyzed both visually (using plots and heatmaps) and quantitatively, to assess how well the system distinguished authentic from pseudo-attributed works.

3.1 Challenges

Throughout the development of the system, we encountered several practical and analytical challenges that required creative problem-solving and adaptation at different stages of the project:

- A major challenge was data acquisition. We needed access to high-quality, machine-readable Arabic texts, both for works attributed to Al-Ghazali and for impostor texts from the same historical context. Many sources were either unavailable in plain text or locked in non-editable formats such as scanned PDFs. After extensive searching, we assembled a dataset of usable TXT files, ensuring coverage across different genres and authorships.

- During training, we encountered computational bottlenecks. Initially using Google Colab Pro and later Colab Pro+, we found these environments insufficient for the heavy demands of Siamese network training. After consulting with professionals, we migrated the training and signal generation phases to Lambda Cloud, which provided more powerful GPUs. This switch significantly accelerated our progress and allowed for deeper experimentation.

- After executing the full pipeline, we noticed unexpected clustering behavior. The model grouped certain books together based primarily on file size rather than content. Upon investigation, we discovered that large test files were dominating the signal patterns due to their scale, while smaller books were underrepresented. To resolve this, we partitioned each book into equal-sized segments (200 KB) and performed a new round of experiments. This adjustment eliminated the size bias and enabled more balanced comparison across texts.

4. Results

As a first step in our analysis, we examine the output signals generated by the trained Siamese network for selected texts in the test group. These signals represent the learned similarity patterns between each book and a pair of impostor texts, processed through the network after feature extraction.

Figures 4 and 5 present representative examples: the signal of *Iḥyāʾ ʿUlūm al-Dīn*, a widely recognized work attributed to al-Ghazali, and the signal of *Mishkāt al-Anwār*, whose authorship is more disputed. These examples illustrate the type of signals obtained from the network, which serve as the basis for further interpretation and clustering.

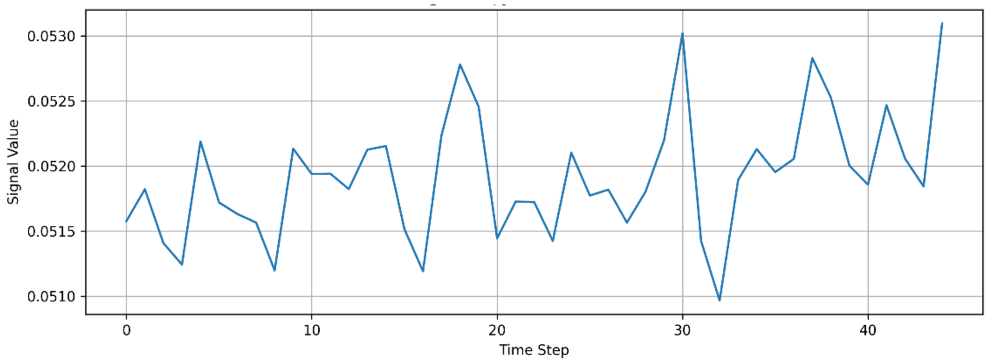

Figure 4. Signal Output for *Iḥyāʾ ʿUlūm al-Dīn*.

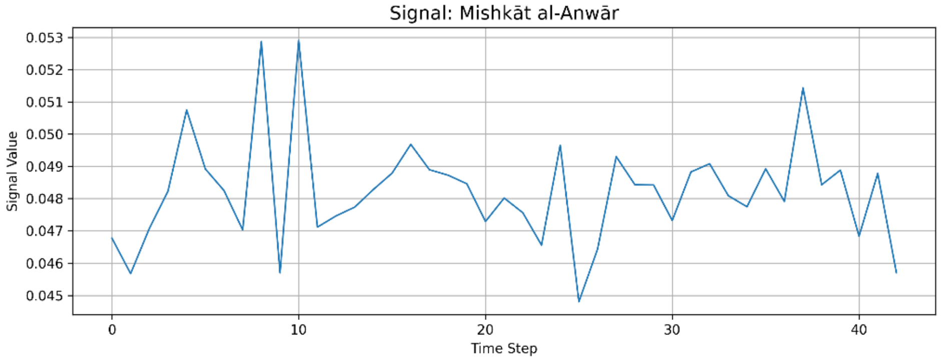

Figure 5. Signal Output for *Mishkāt al-Anwār*.

The signal of *Iḥyāʾ ʿUlūm al-Dīn* displays relatively smooth transitions and moderate variation across time steps, indicating a stable representation that aligns with the network's learned notion of textual coherence typical of al-Ghazālī's style. In contrast, the signal of *Mishkāt al-Anwār* exhibits sharper local fluctuations and higher-frequency oscillations. These irregularities may suggest structural or stylistic divergence from the reference profile, reinforcing doubts about its authorship. While both signals operate within a similar value range, the smoother trajectory of *Iḥyāʾ* reflects stronger alignment with known authentic features, whereas the more erratic pattern of *Mishkāt* suggests potential deviation, supporting its contested status in the literature.

Having obtained the per-sample signals, we proceeded to quantify stylistic divergence using Dynamic Time Warping (DTW). For each iteration, we constructed a DTW distance matrix comparing the signals of all test books. Each row and column correspond to a book, and the (i, j) entry reflects the temporal alignment cost between book *i* and book *j*; the diagonal entries are zero since each book is compared with itself.

For example, in one iteration, the DTW distance between *Iḥyāʾ ʿUlūm al-Dīn* and *Mishkāt al-Anwār* was 32.42, while its distance from *Al-Iqtiṣād fī al-Iʿtiqād* was 30.34. These values illustrate relative stylistic proximity and are used in subsequent anomaly detection and clustering stages.

To quantify how atypical each book's signal is compared to the impostor patterns, we applied Isolation Forest to the DTW matrices from each run. This yielded a normalized anomaly score per book per iteration. As a result, we obtained a matrix where rows represent the test books and columns represent the anomaly scores across 300 randomized impostor configurations.

To consolidate these scores at the book level, we averaged the anomaly values across all segments belonging to the same original text. These aggregate scores served as input for clustering and further interpretation.

Given the scale and complexity of the anomaly score matrix, direct interpretation of individual values was impractical. We needed a method to summarize the overall structure of the data and extract meaningful groupings that could reflect patterns of stylistic similarity and deviation.

To identify this underlying structure, we applied K-Means clustering to the aggregated anomaly scores. Before selecting the number of clusters, we evaluated the Silhouette score, a standard internal validation metric that measures how well each book fits within its assigned cluster compared to others [4]. Based on this criterion, k = 2 provided the best separation among tested values and was therefore chosen for the final clustering, shown in Figure 6.

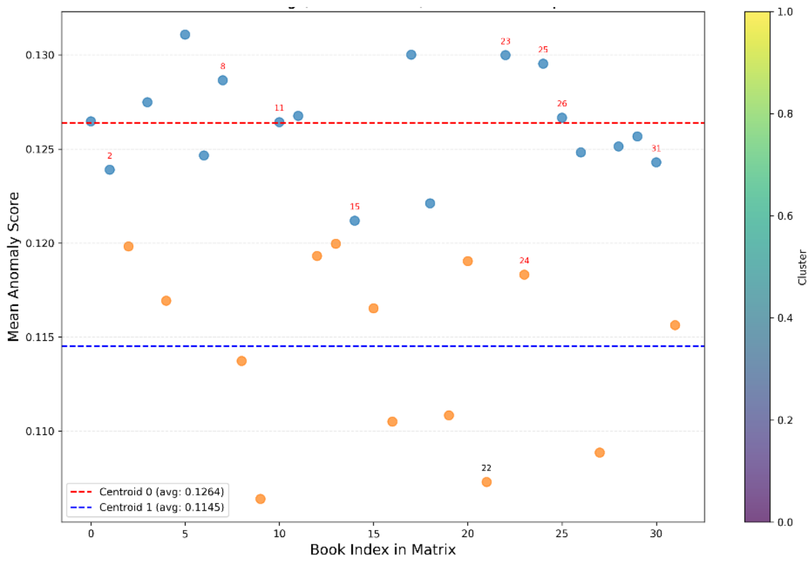

Figure 6. K-Means Clustering of Test Books Based on Anomaly Scores (k = 2).

To interpret the clustering results meaningfully, we classified the books based on established references and scholarly analyses provided in [5-7]. These sources categorize the test group texts into three groups: those confirmed not to be authored by al-Ghazali (marked in red), those of uncertain or disputed authorship (marked in green), and those generally accepted as authentic (unmarked). Tables 2 and 3 present the distribution of these categories across the two clusters, providing a basis for evaluating the alignment between the clustering outcome and historical attribution research.

Table 2. Cluster 0: Books Grouped Separately from *Iḥyāʾ ʿUlūm al-Dīn*.

Book	Index
Al-Adab fi al-Din	1
Al-Ajwiba al-Ghazaliyya fi al-Masa'il al-Akhrawiyya al-Madnun al-Saghir	2
Al-Hikma fi Makhluqat Allah	4
Iljam al-'Awam 'an 'Ilm al-Kalam	6
Al-Kashf wa al-Tabyin fi Ghurur al-Khalq Ajma'in	7
Al-Madnun bihi 'ala Ghayr Ahlihi	8
Al-Mawa'idh fi al-Ahadith al-Qudsiyya	11
Al-Munqidh min al-Dalal	12
Al-Risala al-Ladunniyya	15
Ayyuha al-Walad	18
Bidayat al-Hidaya	19
Khulasa al-Tasanif fi al-Tasawwuf	23
Minhaj al-'Arifin	25
Mishkat al-Anwar	26
Mizan al-'Amal	27
Qanun al-Ta'wil	29
Rawdat al-Talibin wa-'Umdat al-Salikin	30
Risalat al-Tayr Dhikr al-'Anqa'	31

Table 3. Cluster 1: Books Grouped with *Iḥyāʾ ʿUlūm al-Dīn*.

Book	Index
Al-Durra al-Fakhira fi Kashf 'Ulum al-Akhira	3
Al-Iqtisad fi al-I'tiqad	5
Al-Mankhul min Ta'aliqat al-Usul	9
Al-Maqsad al-Asna	10
Al-Qawa'id al-'Ashr	13
Al-Qistas al-Mustaqim	14
Al-Risala al-Wa'zhiya	16
Al-Wasit fi al-Madhhab	17
Fadaih al-Batiniyya	20
Faysal al-Tafriqa bayn al-Islam wa al-Zandaqa	21
Ihya' 'Ulum al-Din	22
Ma'arij al-Quds fi Madarij Ma'rifat al-Nafs	24
Qawa'id al-'Aqa'id	28
Tahafut al-Falasifa	32

Using *Iḥyāʾ ʿUlūm al-Dīn* as our reference point, widely regarded as a core, confirmed work of Al-Ghazali, we interpret Cluster 1, which includes *Iḥyāʾ*, as the group most likely representing authentic authorship. Based on this assumption, we calculated several evaluation metrics to better understand the alignment between our clustering results and external authorship assessments.
Figure 7 illustrates the resulting confusion matrix, which summarizes the classification outcomes based on this interpretation. It provides a breakdown of how many texts were correctly or incorrectly grouped as authentic or non-authentic, serving as the basis for our precision, recall, and purity calculations.

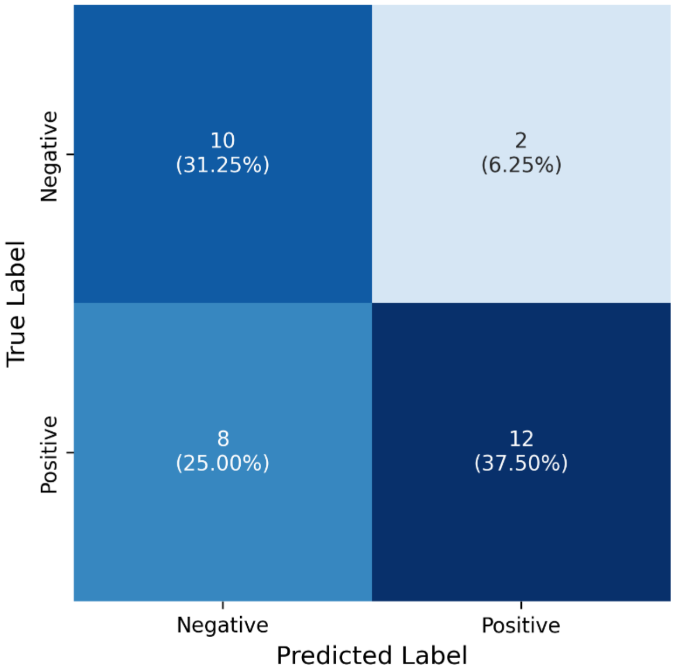

Figure 7. Confusion Matrix Based on Cluster Assignment and External Authorship Labels.

In this evaluation, True Positives (TP) represent 37.5% of the dataset, authentic books correctly grouped in Cluster 1. True Negatives (TN) account for 31.25%, non-authentic books correctly grouped in Cluster 0. False Positives (FP) are 6.25%, and False Negatives (FN) make up 25% of the books. These values are based on a total of 32 test texts and form the basis for the evaluation metrics presented below.

- Purity (Overall):

This metric evaluates the overall clustering quality by checking how many books are in the correct cluster relative to their ground truth label. A higher purity indicates more accurate grouping.

$$\text{Purity} = \frac{\text{Total correctly grouped books}}{\text{Total number of books}} = \frac{20}{32}\text{x}100 = 62.5\%$$

- Purity (Cluster 1):

This checks the internal consistency of the cluster containing Iḥyāʾ by measuring how many books in it truly belong to al-Ghazali. It helps us understand how "clean" the cluster is.

$$\text{Purity}_{\text{Cluster 1}} = \frac{\text{Total correctly grouped books}}{\text{Total number of books}} = \frac{12}{14}\text{x}100 = 85.7\%$$

- Precision (Cluster 1):

Precision reflects the proportion of true al-Ghazali books within all books in Cluster 1. A high precision means fewer unrelated books were grouped with Iḥyāʾ.

$$\text{Precision} = \frac{11}{13}\text{x}100 = 84.6\%$$

(Excludes the book marked "uncertain" from true positives)

- Recall (Recovered Authenticated Works):

Recall measures how many of the truly authenticated works were successfully captured in the Iḥyāʾ cluster. It reflects the ability of clustering to recover al-Ghazali's actual texts.

$$\text{Recall} = \frac{11}{19}\text{x}100 = 57.9\%$$

To better understand the internal distribution of anomaly scores, we generated a Cluster Centroid Visualization. As shown in Figure 8, the plot illustrates the average score profile for each cluster centroid across the 300 runs. Each line represents the centroid of a cluster, capturing the general pattern of normalized anomaly scores associated with the books grouped into that cluster.

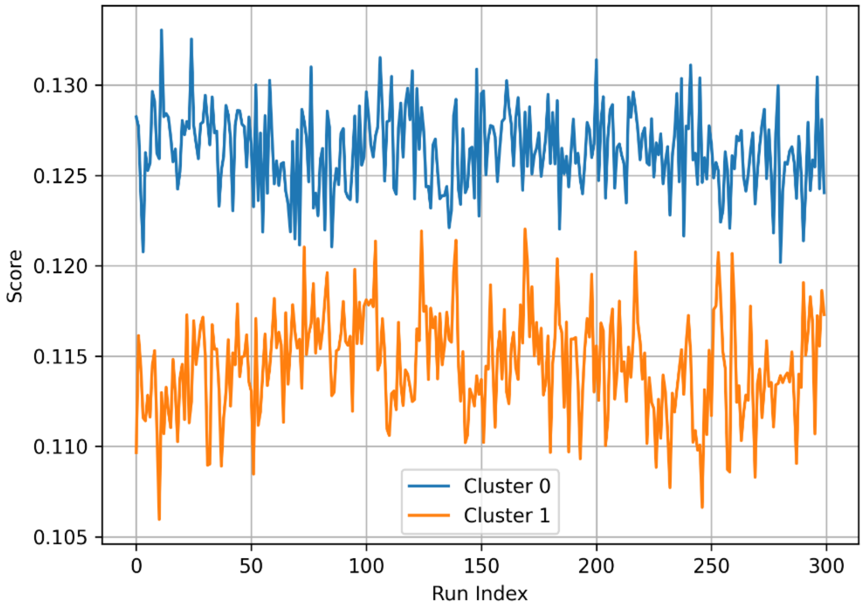

Figure 8. Cluster centroid anomaly score profiles across 300 randomized impostor configurations.

The centroid curve of Cluster 1 exhibits consistently lower anomaly scores compared to Cluster 0 across all 300 runs. This suggests that the books grouped in Cluster 1, including *Iḥyāʾ ʿUlūm al-Dīn*, exhibit higher internal coherence and lower deviation relative to the impostor pairs. In contrast, the higher scores in Cluster 0 indicate stronger deviations and possibly lower stylistic compatibility with the reference structure learned during training. This pattern aligns with our assumption that Cluster 1 is more likely to contain authentic works of al-Ghazali, while Cluster 0 may include misattributed or stylistically divergent texts.

5. Conclusion

This study explored a computational approach to analyzing the authorship of medieval Arabic texts attributed to Al-Ghazali. By converting each book into a signal that captures its similarity to randomized impostor pairs, we generated a structured and interpretable representation for comparing texts. Aggregating these signals across 300 runs allowed us to compute anomaly scores and uncover patterns of stylistic consistency and divergence.

The clustering results reflected a meaningful separation: Iḥyāʾ ʿUlūm al-Dīn, a widely accepted authentic work, was grouped with other texts previously considered reliable in scholarly literature. Quantitative evaluation supported this alignment: True Positive and True Negative classifications covered 69% of the dataset, and overall purity reached 62.5%. Within Cluster 1, interpreted as the authentic group, we achieved 85.7% purity, 84% precision, and a recall of 57.9%, demonstrating the system's ability to recover coherent authorial signatures despite the absence of definitive labels.

Taken together, these outcomes indicate that the project met its goals of building a scalable, reproducible framework for authorship verification under low-resource, morphologically complex conditions.

This suggests that our signal-based approach may serve as a valuable complement to traditional textual analysis, offering new avenues for exploring authorship attribution and the transmission of classical works.

6. Reflection and Lessons Learned

Throughout the course of the project, we maintained a modular and iterative workflow that allowed us to adapt to challenges and refine our methodology as needed. The initial design choices, such as using AraBERT for embeddings and a Siamese architecture for stylistic comparison, proved effective for capturing the linguistic nuances of classical Arabic texts. However, certain aspects required rethinking during development. Looking back, we would have benefited from establishing a more robust data collection pipeline early on, as acquiring high-quality, machine-readable Arabic texts consumed significant time and effort. Similarly, limitations in computational resources initially slowed progress, but migrating to Lambda Cloud mid-project was a key turning point that enabled deeper experimentation. While our preprocessing and modeling pipeline ultimately functioned as intended, additional iterations might have improved model tuning and recall rates.

Despite these challenges, we believe we worked efficiently and systematically, and we are satisfied with how the project unfolded. We met our core objectives: the system successfully operated without labeled training data, handled stylistic variation in Arabic, and produced interpretable outputs. Evaluation metrics and clustering results demonstrated meaningful alignment with external scholarly classifications. These outcomes affirm that the project's goals were achieved, and the approach has potential to support further research in digital authorship analysis.

References

[1] De Rosa, G. H., & Papa, J. P. (2022). Learning to weight similarity measures with Siamese networks: A case study on optimum-path forest. In *Optimum-Path Forest*. https://doi.org/10.1016/B978-0-12-822688-9.00015-3

[2] https://ketabonline.com/ar

[3] https://shamela.ws/

[4] https://medium.com/@hazallgultekin/what-is-silhouette-score-f428fb39bf9a

[5] https://www.ghazali.org/biblio/AuthenticityofGhazaliWorks-AR.htm

[6] https://tariq-library.com/%D8%AA%D8%AD%D9%85%D9%8A%D9%84-%D9%83%D8%AA%D8%A7%D8%A8-%D9%85%D8%A4%D9%84%D9%81%D8%A7%D8%AA-%D8%A7%D9%84%D8%BA%D8%B2%D8%A7%D9%84%D9%8A-pdf-%D9%84%D9%80-%D8%B9%D8%A8%D8%AF-%D8%A7%D9%84%D8%B1%D8%AD/

[7] https://ar.m.wikipedia.org/wiki/%D9%82%D8%A7%D8%A6%D9%85%D8%A9_%D9%85%D8%A4%D9%84%D9%81%D8%A7%D8%AA_%D8%A3%D8%A8%D9%8A_%D8%AD%D8%A7%D9%85%D8%AF_%D8%A7%D9%84%D8%BA%D8%B2%D8%A7%D9%84%D9%8A