Relational Inference in Real-World Settings

Learning across Text, Vision, and Behavioral Data

ISBN:
DOI:

Typeset by LaTeX.
Printed by: Ipskamp Printing
Cover design by: Britt Eva van Leeuwen & Janus Jansen

VRIJE UNIVERSITEIT

Relational Inference in Real-World Settings

Learning across Text, Vision, and Behavioral Data

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad Doctor aan
de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof.dr. J.J.G. Geurts,
volgens besluit van de decaan
van de Faculteit der Bètawetenschappen
in het openbaar te verdedigen
op <datum> om <tijd> uur
in de universiteit

door

Britt Eva van Leeuwen

geboren te Velsen

promotoren:	Prof.dr. R.D. van der Mei
	Prof.dr. S. Bhulai
promotiecommissie:	Prof.dr. M. Hoogendoorn
	Prof.dr.ir. R.H.A. Lindelauf
	Prof.dr. A.P.J.M. Siebes
	Dr. I.H.E. Hendrickx
	Dr. M.C. ten Thij

"Ik heb het nog nooit gedaan, dus ik denk dat ik het wel kan."

Contents

1 General Introduction

In recent years, Machine Learning (ML) and Artificial Intelligence (AI) have reshaped the way complex data is analyzed and interpreted. Through the growth of digital communication, surveillance, and social media, large and diverse datasets are produced. Early ML research primarily focused on relatively well-defined classification and prediction tasks operating on independent observations. While these approaches have been highly successful, they largely assume that each data point can be treated in isolation.

In contrast, research increasingly focuses on problems where relationships between individual data points is targeted instead of the individual data points alone. This shift has made relational inference a main theme of current AI research. Relational inference in this dissertation refers to the task of modeling and predicting relationships between observations, rather than assigning labels to individual instances in isolation. These relationships may be defined over pairs, sets, or sequences, depending on the problem setting. While similarity learning and metric learning focus on representing pairwise relations through distances in a learned feature space, they typically assume fixed training conditions and do not explicitly address changing or unseen classes. Open-set recognition, in contrast, explicitly considers the presence of unknown classes, but remains primarily centered on instance-level classification rather than modeling relations themselves. Graph-based inference extends relational modeling to structured settings by capturing dependencies among multiple entities, often assuming an explicit relational structure is available. In this work, relational inference is treated as a broader framework that encompasses these settings while focusing on learning relations directly under real-world conditions such as distribution shift, limited supervision, and heterogeneous data. Examples include verifying whether two texts were writ-

ten by the same author, recognizing familial relationships from pictures of faces, or identifying coordinated behavior among maritime vessels. Addressing these problems requires models capable of detecting patterns and dependencies across observations.

Despite all the progress that has been made, relational inference remains challenging in real-life scenarios. Models often face changes in data distributions across domains, limited or noisy labeled data, and strict interpretability in sensitive applications areas. These problems are pronounced especially in areas like security, forensic analysis, and biometric recognition, where reliability and transparency are crucial. This dissertation tackles these issues through multiple research routes, emphasizing authorship verification (AV), kinship recognition (KR), and maritime anomaly detection. Although these domains differ in data modalities and application context, they share fundamental methodological challenges centered on relational reasoning under uncertainty.

Human identity and behavior leave marks on multiple observable signals. Written language contains stylometric patterns shaped by cognitive processes, habits, and intents. Facial structure holds hereditary information, possibly indicating biological correspondence. Likewise, maritime vessel movements often follow patterns reflecting either legitimate operational behavior or coordinated illegal activity. While these signals provide valuable information, extracting reliable and generalizable structure from them remains a non-trivial task in practice. Just because these patterns exist does not mean that they can be reliably modeled in practice.

Recent advances in ML, particularly deep learning, representation learning, and multimodal modeling, have significantly improved performance across natural language processing (NLP), computer vision, and behavioral analysis. However, many of these successes depend on assumptions like access to large-scale labeled datasets, balanced class distributions, and consistent training and test conditions. These assumptions are particular for a controlled environment, but are often violated in real-world scenarios. Data distributions shift across time, platforms, and environments; annotations may be incomplete or noisy; and observational conditions may vary substantially. Writing styles differ across communication platforms and genres, visual kinship cues may vary across different age groups and demographic backgrounds, and maritime behavior is influenced by environmental factors. As a result, systems optimized mainly for benchmark performance may not generalize well beyond controlled experimental settings. In high-stakes areas such as forensic analysis or surveillance, it is often more important to have robust and adaptable models than to achieve marginal gains in perfect conditions.

Motivated by these challenges, the aim of this dissertation is to develop models

that are robust across diverse and evolving real-world settings. This requires rethinking the usual evaluation tactics, taking into account domain shifts and learning representations that capture relevant patterns across contexts. At the same time, it also calls for emphasizing interpretability, ensuring the transparency of models under changing conditions. Ultimately, to make relational inference both useful in the real world and scientifically sound, it requires building AI systems capable of handling dynamic real-world environments.

To address these challenges in a structured manner, this dissertation investigates relational inference across three complementary domains: (1) maritime security and anomaly detection, (2) AV and linguistic identity, and (3) KR in facial imagery. Each of these sections explores a different modality of relational reasoning, while collectively contributing to a unified perspective on robust and generalizable inference under real-world conditions.

1.1 Maritime Security and Anomaly Detection

Global maritime transportation is crucial for sustaining international trade and economic stability. However, the scale, openness, and limited observability of the maritime domain create easy opportunities for illegal activities like drug trafficking, human smuggling, and illegal fishing. Consequently, monitoring vessel activity across large regions of the ocean is both an operational and analytical challenge, and has been widely studied in the context of maritime anomaly detection [1]. Surveillance systems need to be capable of processing large amounts of positional and behavioral data while also being able to quickly spot suspicious patterns that can support law enforcement intervention.

A key data source for current maritime monitoring is the Automatic Identification System (AIS). AIS messages provide information about a vessel's identity, position, speed, and navigational behavior [2]. However, AIS data is far from perfect in practice, since messages can be noisy, incomplete, or purposefully manipulated. Vessels engaged in illegal activity could disable their transponders, imitate traffic patterns of lawful vessels, or take advantage of regions with limited surveillance coverage.

Within this context, this dissertation explores supervised anomaly detection in AIS data for the detection of illicit at-sea activities. In particular, the focus will be on so-called 'drop-off' events, a method used most commonly in maritime drug trafficking. It involves coordinated interactions between vessels to "drop off" drugs.

Coordinated multi-vessel operations, such as at-sea transshipments of illegal goods,

are particularly challenging. Fixed rule-based methods have trouble capturing the subtle behavioral patterns and precisely timed interactions that these operations often depend on. ML methods offer a promising alternative by learning typical vessel behavior patterns and identifying deviations that may indicate suspicious activity. Nevertheless, there are still challenges regarding extreme class imbalance, evolving criminal strategies, and the scarcity of reliably labeled criminal events.

To address these challenges, this work explores the use of historical AIS trajectories, temporal segmentation techniques, and recurrent neural architectures, such as Long Short-Term Memory (LSTM) models, to demonstrate that supervised learning can be effectively applied to detect the rare behavioral patterns in fishing vessel traffic. This approach enables the modelling of vessel behavior over time and provides a foundation for detecting rare and complex interaction patterns in realistic maritime settings.

In contrast to the pairwise nature of AV and KR, the maritime case study is more closely aligned with behavioral anomaly detection. The model operates on temporal windows of AIS trajectories for individual vessels and classifies patterns as normal or suspicious. The relational aspect in this setting is therefore not explicitly modeled through pairwise inputs, but is reflected in the structure of the behavior itself. Suspicious activities, such as coordinated rendezvous or transfer events, inherently involve interactions between multiple vessels and their environment, even when these are only indirectly observed through individual trajectories. As such, the relational component lies in the interpretation of these temporal patterns as indicators of underlying interactions, rather than in the explicit modeling of relationships between instances.

1.2 Authorship Verification and the Complexity of Linguistic Identity

Authorship analysis has been extensively studied in shared tasks and surveys, most notably in the PAN evaluation campaigns [3]. AV remains a relatively recent subfield where substantial progress can still be made. AV aims to determine whether two or more documents were written by the same author. It plays a vital part in applications such as digital forensics, plagiarism detection, misinformation analysis, and the study of historical texts.Unlike authorship attribution, which operates in a closed-set setting with a predefined set of candidate authors, AV is typically formulated as an open-set problem. As a result, AV models must determine whether two documents are stylistically consistent without relying on prior knowledge of the particular author.

Writing style emerges from a complex interaction between situational context, cultural background, and cognitive habits. Stylometric features such as lexical diversity, syntactic structure, punctuation, and sentence rhythm can be helpful in distinguishing authors. Concurrently, semantic content also influences textual similarity, making it challenging to distinguish topical overlap from actual stylistic components.

Recent advances in NLP have enabled the use of deep contextual models, which are able to capture rich semantic representations of text. Despite their strength, these models do not explicitly highlight stylistic elements that are essential to AV. This has motivated hybrid methods that combine conventional stylometric features with contextual embeddings in order to improve robustness and interpretability.

Within this context, this dissertation explores two complementary lines of work on hybrid AV.

The first investigates hybrid approaches to AV that combine contextual representations with explicit stylometric features. In particular, different model architectures have been investigated for modelling pairwise relationships between texts, enabling verification under open-set conditions.

The second extends this direction to a cross-domain AV setting, where models must generalize across different communication types and writing contexts. Depending on the platform, audience, or communicative purpose, authors may exhibit substantial shifts in writing style. This makes generalisation across datasets and domains a central issue in practical applications of AV. In addition, the generalization across unseen domain combinations is studied.

1.3 Kinship Recognition and the Role of Biometric Similarity

The goal of KR is to identify familial relationships based on biometric data, most commonly facial images. Potential applications include social media analysis, genealogical research, and missing person investigations. KR has been widely studied using benchmark datasets such as FIW, FIW-MM, KAN-AV, and TALKIN-Family [4], [5], [6], [7]. The task is inherently challenging. Unlike facial recognition, which focuses on identifying the same individual across multiple images, KR requires detecting subtle and often ambiguous similarities. These can be influenced by a variety of factors, including age differences, gender, environmental factors, and variations in genetic expression. Consequently, identifying consistent and discriminative kinship cues remains a complex task.

Deep learning has significantly advanced KR by enabling models to learn hierarchical facial representations. In practice, many approaches rely on transfer learning from large-scale face recognition models due to the limited size of kinship datasets. However, challenges remain. Available data is often small, imbalanced, and lacking in diversity, and performance can vary across modalities such as images and video.

Within this context, this dissertation investigates two complementary directions. The first investigates which facial features contribute most to automated KR, using feature representations derived from generative models to analyse how models rely on attributes such as age-related traits and localized facial characteristics. The second examines the role of pre-trained models and learning strategies under limited supervision, comparing different fine-tuning approaches across image-based and video-based settings to better understand how kinship representations are learned.

Finally, interpretability remains an important consideration. Understanding which features drive model decisions is essential for both scientific insight and assessing potential biases, supporting the development of more transparent and reliable KR systems.

1.4 Contributions and Research Scope

Domain shift presents a common challenge in all three of the domains examined in this dissertation: maritime surveillance, AV, and KR. When applied in different environments, models that were trained under one set of conditions frequently perform worse. Model performance can be deteriorated by changes in writing style, variations in biometric capture conditions, or environmental variations in vessel behavior. Representations that capture patterns that are meaningful in a variety of contexts are necessary to address these issues.

Another challenge is the scarcity and uneven distribution of labeled data. In many real-world situations, datasets are frequently unbalanced and heterogeneous. Instances of coordinated illegal maritime activities, authorial matches across diverse texts, or familial kinship pairs are uncommon. Models find it challenging to learn reliable patterns and generalize across contexts as a result of this scarcity. Careful modeling decisions are needed to address this challenge, such as utilizing pre-trained embeddings, combining various feature types, and creating architectures that function well in imbalanced and cross-domain scenarios. This dissertation investigates how to extract meaningful patterns in spite of these constraints by concentrating on supervised learning techniques customized to the available data, enhancing model reliability and applicability in real-world, high-stakes situations.

Within this scope, the dissertation focuses on developing methods that improve robustness and generalization across heterogeneous and data-constrained environments.

1.4.1 Research Questions

The contributions of this dissertation span multiple modalities and application domains, and are guided by a set of overarching research questions:

1. How can relational patterns be effectively modeled in diverse data sources such as vessel trajectories, textual data, and facial imagery?
2. How can meaningful representations be learned in settings with limited and imbalanced labeled data?
3. How can ML models be designed to remain robust under domain shift and varying real-world conditions, particularly in cross-domain textual settings?
4. How can relational inference benefit from integrating complementary feature representations?
5. How can interpretability be incorporated into ML models, particularly in the context of KR, to better understand and validate their predictions in high-stakes applications?

To clarify the relation between the research questions and the empirical chapters, this dissertation provides an overview of how each question is addressed across the different studies. Research question 1, focusing on modeling relational patterns across modalities, is addressed in all empirical chapters through different data types: vessel trajectories in Part I, textual data in Part II, and facial imagery Part III. Research question 2, on learning meaningful representations under limited and imbalanced data, is addressed across Chapters 2, 4 and 6, where each setting involves varying degrees of data scarcity. Research question 3, concerning robustness under domain shift, is primarily investigated in 4 through cross-domain AV experiments. Research question 4, regarding the combination of complementary feature types, is studied in Chapters 3 and 6 through the integration of semantic and stylistic features, as well as visual and pre-trained representations. Research question 5, focusing on interpretability, is mainly explored in Chapter 5 in the context of KR.

1.4.2 Contributions

In response to these questions, the main contributions of this dissertation are as follows:

1. In the domain of maritime surveillance, this work advances anomaly detection by modelling coordinated vessel behavior using AIS data, with particu-

lar attention to challenges such as class imbalance and evolving operational patterns.
2. In AV, it develops and evaluates hybrid approaches that combine semantic representations with stylometric features, with a focus on cross-domain generalization and realistic verification settings.
3. In KR, it investigates both feature-level interpretability and the role of pre-trained models across image- and video-based modalities, providing insight into how kinship-related patterns are learned.
4. Across all domains, this work demonstrates how combining different representation types and modeling strategies can improve robustness in relational inference tasks.
5. Finally, this dissertation examines how models can generalize to unseen domains and new data conditions, highlighting the importance of learning representations that remain stable across variations in data distribution and application context.

When combined, these contributions establish a perspective on relational inference across heterogeneous data modalities. By studying maritime trajectories, textual data, and facial imagery, this dissertation highlights shared challenges such as limited supervision, class imbalance, and domain shift, and investigates how different representation learning strategies can address them in practice.

1.5 Publications

The following publications are part of this dissertation:

B. van Leeuwen and M. Nützel, "Detecting Drug Transfers via the Drop-off Method: A Supervised Model Approach Using AIS Data," *Machine Learning with Applications*, volume 18, October 2024, article 100590.

B. van Leeuwen, S. Bhulai, and R.D. van der Mei, "Combining Style and Semantics for Robust Authorship Verification," *Machine Learning with Applications*, volume 22, December 2025, article 100732.

B. van Leeuwen, S. Bhulai, and R.D. van der Mei, "Cross-Domain Authorship Verification with Feature Interaction Networks: Evaluating No-Holdout and Holdout Protocols." *Under review*, 2026.

B. van Leeuwen, A. Gansekoele, J. Pries, E. van de Bijl, and J. Klein, "Explainable Kinship: A Broader View on the Importance of Facial Features in Kinship Recognition," *International Journal On Advances in Life Sciences*, volume 14, 2022.

B. van Leeuwen, Y. Hartman, and M. Hoogendoorn, "Evaluating Pre-trained Models for Facial Kinship Recognition: A Comparative Study of Video and Image-based Approaches." *Under review*, 2025.

Part I

Maritime Security

2 Detecting Drug Transfers via the Drop-off Method: A Supervised Model Approach Using AIS Data

Contents

Based on
B. van Leeuwen and M. Nützel, "Detecting Drug Transfers via the Drop-off Method: A Supervised Model Approach Using AIS Data," *Machine Learning with Applications*, volume 18, number 1, article 100590, October 2024.

Abstract

This chapter introduces a novel approach for detecting sea-based drug transfers through the utilization of Automatic Identification System (AIS) data. We propose a model for the detection of the 'drop-off' method, a prevalent technique for contraband smuggling at sea. Unlike the prevailing focus on unsupervised methods in existing maritime anomaly detection research, our paper introduces a supervised model tailored to the 'drop-off' method, particularly in the context of fishing vessels. We have developed a machine learning algorithm, employing a Long Short-Term Memory (LSTM) model capable of identifying 'drop-offs'. This model holds the potential for integration into real-time surveillance systems. Furthermore, our model demonstrates generalizability, thereby enhancing maritime security efforts and providing invaluable assistance in countering drug traffic on a global scale.

2.1 Introduction

2.1.1 Motivation

Recently, the BBC published an article about three men who were put under investigation two weeks after their rescue on the coast of Australia [8]. Suspicions were raised after almost 400 kg of cocaine was found at their rescue location. Although the men claimed to have fallen overboard while fishing, the authorities suspected their story may not hold water. In fact, it is believed that the men were attempting to retrieve cocaine from the ocean and transfer it onto their fishing vessel as part of a drug smuggling operation. This transfer of contraband through the ocean is known as the 'drop-off' or 'pick-up' method. Although this BBC article was focused on an incident along the Australian coastline, this particular method of drug trafficking is observed in various parts of the world. Despite the relatively low number of arrests made in relation to drop-offs in the Netherlands, the Dutch police suspect that the 'drop-off' method is being employed with increasing frequency along the country's coast. This suspicion is supported by the growing number of packages discovered in the ocean or washed ashore [9].

In 2000, the International Maritime Organisation (IMO) adopted a requirement that obliged certain vessels to be equipped with an *Automatic Identification System* (AIS). It concerns passenger ships and all vessels larger than 300 tonnage on international voyage, and larger than 500 tonnage not engaged in international voyage [2]. Through AIS, these vessels are obliged to transmit certain dynamic and static information: identity, type, position, course, speed, navigational status, and other safety-related messages. Furthermore, the vessel automatically receives the same data from other vessels on the AIS. The idea behind the implementation of this regulation was to significantly strengthen the safety and security of ships while also facilitating more effective monitoring by the authorities. In 2016, The Dutch government further mandated this regulation for all vessels operating commercially and all pleasure crafts longer than 20 meters [10]. The sea is incredibly busy, 90% of all international transport occurs at sea [11]. Most of these transporters are large cargo ships carrying containers, which are obliged to sail within a designated shipping lane. Additionally, many other vessels sail the coast every day. Unlike cargo ships, these vessels are not bound by the shipping lanes and consequently tend to follow more unpredictable and chaotic paths. Therefore, detecting a pick-up with the current methods in place is possible, but challenging. The concentration of vessels is high, especially in regions around coasts (as shown in Figure 2.1), and the trajectories are complex.

The suspected number of drop-offs occurring is high and both historical and real-time data are available. Utilizing machine learning for automated surveillance

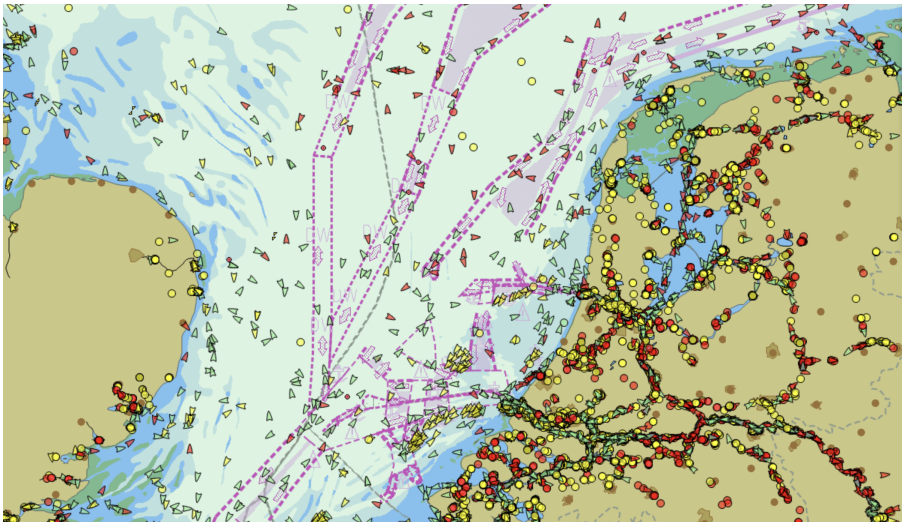


Figure 2.1: Snapshot of all vessels equipped with AIS on the coast of The Netherlands (2023). Every arrow and dot represents a vessel.

holds significant promise. There have already been worldwide implementations of automated anomaly detection using Recurrent Neural Networks (RNNs) or cluster methods. However, these studies primarily focus on unsupervised approaches, which concern general trajectory anomalies. Research on the detection of anomalies at sea using supervised approaches is very limited, especially concerning the detection of the drop-off method. This is a gap in this field that we are aiming to address with this research. The goal of this research is to develop a machine-learning algorithm that can detect the possibility of a drop-off taking place. The algorithm could be integrated into real-time data systems for future use. It would serve as a trigger to alert The Maritime Police (TMP) and Coastguard when there is a high likelihood of a drug transfer occurring.

2.1.2 Modus Operandi

The type of drug transfer this chapter focuses on is those using the drop-off method, performed at sea. It is important to note that while this method is commonly known as the drop-off method, the method actually consists of two actions: (1) a *drop-off* of the contraband into the water, and (2) a *pick-up* of the contraband out of the water. A mother vessel sails through a designated shipping lane outside the coast of The Netherlands. Then at a pre-determined time and location, the mother vessel drops a package of illegal drugs into the water, usually without changing its speed or course. The daughter vessel is often already present in the shipping lane or the area. Following the drop-off, this vessel picks up the con-

traband with either nets or a hook, thereby executing a pick-up. The organizers of the drop-offs are aware that the daughter vessel's live AIS activities are being monitored. For this reason, the contraband is frequently transferred to at least one other vessel before being brought to land. Generally speaking, to execute a pick-up the daughter vessel must maneuver to the location of the package, decrease speed, and, after performing the pick-up, often makes a U-turn. This behavior differs from vessels occupied with normal activities, such as fishing. Since the mother vessel maintains its course and speed during the drop-off, its involvement is very challenging to trace using AIS data. However, the daughter vessel does make several movements that are anomalous in the AIS data. On top of that, fishing vessels are known to be used for the smuggling of migrants, drug trafficking, weapon trafficking, and acts of terrorism [12]. Fishermen are often recruited for their knowledge of the sea and are generally not the masterminds of the criminal act. They are more susceptible to these recruitments due to financial difficulties caused by the declining fish stocks in many regions of the world [9].

Accordingly, the anomaly detection is focused on the daughter vessel. Since this research is limited to anomaly detection related to the daughter vessel, the 'pick-up' action rather than the 'drop-off' action is of relevance. When talking about the method, we will mention 'drop-off' and when talking about the anomaly, we will refer to 'pick-up'.

2.1.3 Related Work

In the context of all types of maritime anomaly detection, there exists a wide variety of classification and prediction methods. In 2022, Wolsing et al. survey on anomaly detection of AIS tracks [1]. To categorize the existing methods, they adopted the same classification as described by Lane et al. [13]: 'route deviation', 'unexpected activity', 'port arrival', 'close approach', and 'zone entry'. From these five categories, the classification of potential pick-ups falls under the 'route deviation' category, although 'close approach' and 'unexpected activity' are also closely related. Accordingly, the remainder of this section will focus on the existing research related to these three categories.

Venscus et al. [14] introduced a method using a LSTM with bootstrapping to predict route deviation anomalies in an unsupervised manner. The resulting model accurately detected abnormal movements such as 'vessel slowdown', 'turn around', 'sharp direction change', and 'unplanned stopping'. It is noted that this works for predictable paths such as cargo vessels, but that it is much more challenging with fishing vessels' complex trajectories. Nyguyen et al. [11] developed GeoTrackNet, an anomaly detection model in route deviations. It uses probabilistic notation of tracks with a Variational Recurrent Neural Network (VRNN) trained

for anomaly detection. Unlike previous models that try to detect route deviations, GeoTrackNet can be applied to any type of vessel, including fishing vessels. The authors do note that their patterns are the most complex of all vessels and thereby pose the greatest challenge for the model to learn their deviations. GeoTrackNet uses a four-hot encoding vector as input to the neural network. This is a concatenation of four one-hot-encoded vectors of the ship's latitude, longitude, Speed Over Ground (SOG), and Course Over Ground (COG). The authors argue that this is more appropriate for a neural network type of model since the encoding better represents the geo-spatial meaning of these variables. Be that as it may, a large memory capacity is required for this application [11].

An interesting research under the 'unexpected activity' anomalies is by Singh et al. [15]. They propose a neural network framework to detect intentional on-and-off switching of the AIS transmitter. It uses the numerical values of latitude, longitude, SOG, and COG as input for their model. The authors test their model by comparison to known power outrages and reach a $99,99\%$ accuracy in detecting whereas the AIS that is transmitted is normal, intentionally turned off, or is due to a power outage, only missing a few instances.

Within the 'close approach' category, the identification of collision of vessels has been researched by Lui et al. [16]. In this research, an RNN is implemented to predict collisions based on interpolating geographical information. While the research shows promising results, it is suggested to use other variables such as SOG and COG in further research. Fang et al. [17] do consider SOG and COG as variables in addition to longitude and latitude and approach the classification with spatial indexing. A grid is defined on the spatial resolution and a close approach is determined by checking pairs of vessels in the same or adjacent squares. The results of the research are compared to official precautionary areas and achieved desirable results in some areas and inaccurate results in others. The authors highlight the application of the model in real-time scenarios for further research. Outside the detection of collisions, close-approach anomalies have been researched to a smaller extent. Most research done in anomaly detection for fishing vessels is regarding illegal fishing activities. While this is not the goal of this research, these models are built with the same trajectory data and so address the issue of complex trajectories. In 2020, Arateh et al. [18] introduced FishNet. A convolutional neural network (CNN) that classifies fishing trajectories. While this model serves as a reliable indicator of whether a vessel is engaged in fishing or not, reaching an F-1 score of 92.35% versus the 90% score of the highest baseline, it lacks the capability to differentiate the found anomalies from other routine activities, such as port arrival or certain route deviation.

As of the time of composing this chapter, no prior research on supervised 'drop-

off' method detection has been identified by the authors. Our research differs from previous research in multiple ways. Of the existing research on illegal activities at sea, most focus on illegal fishing. This concerns mainly forbidden navigational maneuvers. This makes illegal fishing an issue of trajectory, whereas the drop-off method is an issue of behavioral patterns (including, but not limited to trajectory patterns). Other research focuses on anomaly detection, taking the entirety of illegal activities into account. However, due to the lack of labeled data, the models proposed for this are unsupervised, which makes labeling the anomalies quite challenging.

This research aims to bridge this gap by introducing to this field a supervised model, using labeled data. The data is preprocessed in a way that the model is applicable not just within the area of our data, but in many parts of the ocean. Accordingly, we get a model that is trained specifically on the 'drop-off' method that is easily applicable.

2.2 Data

2.2.1 Automatic Identification System

Every Fishing vessel (longer than 15 meters) is obliged to transmit an Automatic Identification System, AIS for short. These transmits consist of static and dynamic information. The static information has been manually entered by the vessel's crew and consists of information such as the Maritime Mobile Service Identity (MMSI) number of the ship, its name, its status, dimensions of the boat, destination, and vessel type. The dynamic information is transmitted automatically and consists of the date and time, the vessel's SOG, its location represented in longitude and latitude coordinates, its COG, its Heading (HDG), and Rate of Turn (ROT). The variables, together with their type and numerical range can be observed in Table 2.1. Unlike the other variables in the table, the range for the SOG is not fixed. AIS data often includes unrealistically high SOG values [19]. To filter out this noise from the dataset during pre-processing of the data, a threshold is used based on research by Greig et al. [20]. Hence, the truncation threshold for fishing vessels is set at 15 knots.

Although AIS data is widely accessible, it has its limitations. The primary issue associated with using AIS data is that it can be unreliable. AIS sends out signals every $2-12$ seconds, depending on the ship's SOG as well as gaps in the reception [21]. This is usually a result of saturation of the system due to a high concentration of vessels in an area or insufficient quality of the transmissions caused by the equipment of the receiver and/or the transmitter [1], [22]. As a result, the

Table 2.1: All the AIS variables, together with their type and numerical range.

Name	Type	Description
Date and UTC Time	Datetime	The date and UTC time of the transmission.
MMSI	Integer	Unique 9 digit identifier number for radio station(s).
Status	Integer $\in [1, 15]$	Status of the vessel by manual input, e.g., 'Fishing'.
Latitude (°)	Float $\in [-90, 90]$	Position coordinate.
Longitude (°)	Float $\in [-180, 180]$	Position coordinate.
SOG (knots)	Float	Speed over Ground, the speed relative to the ground.
HDG (°)	Float $\in [0, 365]$	Heading, the direction in which the vessel is pointing.
COG (°)	Float $\in [0, 365]$	Course over Ground, the direction which the vessel is heading.
ROT (°/min)	Float $\in [-720, 720]$	Rate of Turn, the position of the steering wheel.
IMO Number	String	Unique 7 digit number for propelled sea going merchant ships of more than 100 Gross Tonnage.
Name	String	Name of the vessel.
Call Sign	String	Unique alphanumeric vessel identity.
Length Bow (m)	Integer	Vessel measurement.
Length Stern (m)	Integer	Vessel measurement.
Length Overall (m)	Integer	Vessel measurement.
Width Port (m)	Integer	Vessel measurement.
Width Starboard (m)	Integer	Vessel measurement.
Width Overall (m)	Integer	Vessel measurement.
Draught (m)	Integer	Vessel measurement.
Destination	String	The intended destination.
Vessel Type	String	e.g., 'Vessel'.
Extra info	String	e.g., 'Fishing'.

trajectories often contain significant gaps.

Besides this, the missing information can also be caused by the fishing vessel's

crew, not cooperating with the AIS system. It is possible that a fishing vessel's captain has, intentionally or not, never activated its AIS system or incorrectly entered static information. Furthermore, the AIS transmitter can be turned off. This can be with the goal of (drug) trafficking or more innocent crimes such as the prevention of disclosing a good fishing spot to other fishers in the area [1], [22]. However, this does generally get easily noticed by the authorities.

2.2.2 Data Set

Data Extraction

The dataset used for this research was historical AIS data downloaded from MadeSmart [23]. The negative data points were obtained by downloading all the tracks available in 2022, with two filters. The first filter required the tracks to have occurred in the polygon shown in Figure 2.2. This polygon was determined in collaboration with TMP. Additionally, the polygon encompasses a large shipping lane which cargo ships navigate through. Therefore, a majority of the pick-ups are expected to occur in this lane. The second filter is that on ship type. Only fishing vessels are exported. Since the vessel type in the AIS system relies on manual input from the skipper, it is susceptible to human error. Nonetheless, the filter was necessary to effectively restrict the amount of downloaded data. The resulting negative dataset consists of AIS information from almost 40 million tracks. The positive dataset is built on 18 tracks of known pick-ups. The tracks cover the estimated time of suspicious activity, spanning from 3 hours to 4 days. The whole positive dataset is much smaller, consisting of nearly 200 thousand data points. The majority of the tracks occurred within the past three years, with the earliest one dating back to 2013. For each positive data point, there has either been direct evidence by investigation conducted by TMP or compelling suspicions by experts within the TMP. Using the TMP's knowledge of known pick-ups, the data from that day was extracted from MadeSmart.

Class Imbalance

Due to the small availability of the positive dataset, the dataset suffers from extreme class imbalance. The minority class makes up 0.5% of the total dataset. There is no statistics on the actual percentage of tracks that represent pick-ups per year. It can, however, be assumed that a class imbalance is realistic. Only a small amount of the traffic at sea is en route to a pick-up. To enhance generalizability, the training data should reflect the real world and this imbalance must remain [24]. It is challenging to pinpoint the exact threshold at which the minority class becomes too under-represented. Nonetheless, previous research suggests that, in general, when the minority class constitutes less than 1% of the dataset, it leads to

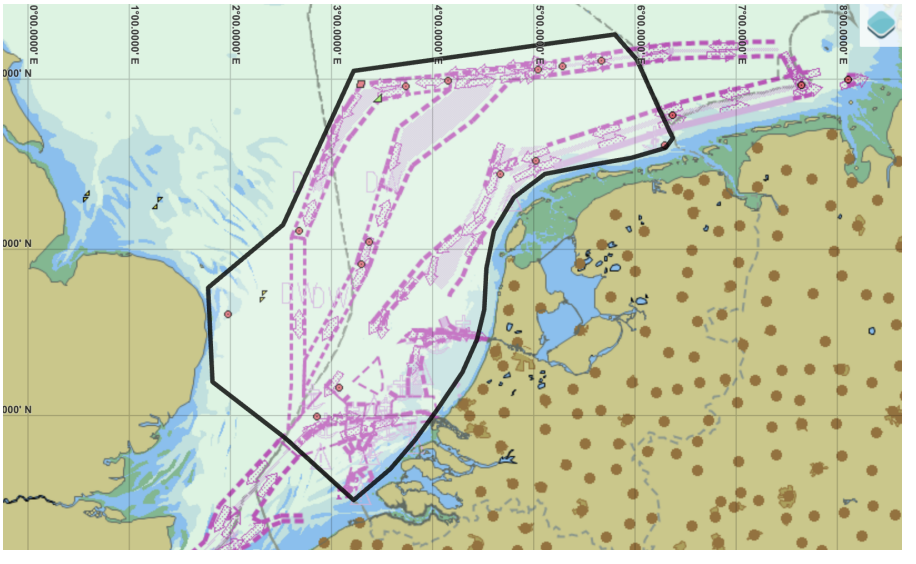


Figure 2.2: The polygon filter on downloaded data for the negative dataset: an area with significant traffic activity.

significant class imbalance issues and can result in poor model performance [25], [26]. We deal with the extreme dataset imbalance by using data augmentation, where the negative class is undersampled resulting in a minority set that makes up 1% of the complete dataset.

2.2.3 Dataset Preprocessing

For the proposed model, the dynamic variables *SOG*, *COG*, *Latitude*, *Longitude*, as well as the static variables *Date and UTC Time*, and *MMSI* are considered. The other AIS variables are removed from the dataset. During pre-processing some general errors should be addressed [27]. This includes incorrect MMSI lengths, duplicate data points, and abnormal latitude, longitude, SOG, and COG values. Possible errors such as large gaps in the reception of latitude and longitude data are not to be removed from the dataset. The large gaps in the data could be an indicator of intentional tampering with the material. By removing these gaps, important information might get lost. The model must focus on finding anomalies in the trajectories. To achieve this, the latitude and longitude values undergo normalization to not contain positional information. This normalization process involves converting latitude and longitude to relative values by subtracting each coordinate from the initial coordinates of the track.

2.3 Model Description

2.3.1 Proposed Model: LSTM

The selected model for this research is a (stacked) LSTM model [28] which is a type of Recurrent Neural Network (RNN). It is an extension of traditional RNNs, aiming to address the vanishing gradient problem. The vanishing gradient problem arises when training on long sequences, as the gradients approach zero over time. It prohibits the model's ability to capture long-term dependencies. The LSTM model aims to improve the long-term dependencies by the incorporation of a memory cell. This enables the model to selectively retain or discard information over long sequences. This capability makes LSTM especially useful in (spatial) time-series analysis. The input to an LSTM model is commonly a sequence of data, where each element in the sequence represents a time step. In the classification of dynamic AIS data, the sequence is (a segment of a) trajectory and the element AIS data at a certain point in time.

The model's architecture was experimented with both single-layer and two-layer configurations. The choice of configuration depends on the complexity of the task. For more complex tasks that require capturing long-term dependencies or intricate patterns in the input sequence, a two-layer LSTM model could be beneficial. The additional layer allows the model to learn more abstract representations and capture higher-level features. This increased depth can enhance the model's capacity to handle complex sequences. If the task is not complex and a two-layer model is used, overfitting is likely to happen. Since the complexity of the data is difficult to determine, the optimal configuration is found using hyperparameter tuning. This was done with Hyperopt [29], which explores the hyperparameter search space by making use of Sequential Model-based Optimization (SMBO), a Bayesian optimization technique. As the method assesses the quality of the hyperparameters using an objective function, it is crucial to select an appropriate objective function like the AUC-PR.

Table 2.2 contains the tuned input for Hyperopt. The $U(a, b)$ indicates the uniform distribution and $dU(a, b)$ the discrete (integer) uniform distribution between the values of a and b. The parameters found through tuning can be observed in Table 2.2.

2.3.2 Sliding Window

The classification model will be trained and tested on historic AIS data. This entails that the LSTM model is trained on previously completed trajectories. However, in further development, the model is meant to be employed as a live classific-

Table 2.2: Hyperparameters for the LSTM model.

Hyperparameter	Parameter Space
Number of layers	$\sim dU(1, 2)$
Number of units layer 1	$\sim dU(32, 256)$
Number of units layer 2	$\sim dU(32, 256)$
Dropout	$\sim U(0.5, 0.65)$
Learning rate	$\sim U(0.0001, 0.01)$
Batch size	$\sim dU(2048, 4096)$

ation model. In that case, the complete trajectories are not yet known and thus can not be used as input for the model. As a solution, a sliding window will determine the input sequence. A sliding window involves moving a fixed-size window over every n^{th} received AIS message.

2.3.3 Adam Optimization

Adam regularization [30] is implemented as an optimization algorithm. Adam is chosen as the preferred optimizer due to its speed and early convergence capabilities. The β_0 and β_1, ϵ, and decay are kept as is standard for Keras [31] 0.9, 0.99, 10^{-8}, and 0 consecutively. The learning rate is optimized with hyperparameter tuning.

2.3.4 Loss Function

Weighted binary cross-entropy (WBCE) is used as a loss function. The binary cross-entropy is a logical choice in binary classification. It is weighted since this approach assigns a higher penalty to samples from the minority class. This causes more accurate detection of these cases. Research shows that using weighted versus non-weighted binary cross-entropy could increase recall by around 10% while not decreasing precision by more than 3% [32]. Given that the minority class represents the cases of suspicious activities, achieving a higher recall would lead to a higher number of pick-up classifications. As the objective of this research is to prioritize finding pick-ups over missing a potential pick-up, the potential increase in recall could significantly enhance performance. The WBCE function is given in Equation (2.1). N is the number of data points, β is a scaling parameter that gives preference for either fewer false positives or fewer false negatives, y is the binary label (1 for suspicious or 0 for normal behavior), and $p(y_i)$ is the predicted probability of the label of point i being suspicious, or belonging to the

positive class, 1.

$$H_p = -\frac{1}{N}\sum_{i=1}^{N} \beta y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)). \tag{2.1}$$

2.3.5 Performance Measure

The data is heavily unbalanced, making accuracy misleading. Therefore, recall and precision are much more informative. For the implementation of the model, it is preferred for the system to give a false alarm rather than ignore a potentially suspicious track. Recall is, therefore, preferred over precision. The Area Under the Precision-Recall Curve (AUC-PR) is a performance measure that can be given a threshold that weighs in favor of either recall or precision. Due to these characteristics, the AUC-PR serves as an optimal objective function for our research. Next to the AUC-PR, other metrics were also chosen for the goal of detecting as many pick-ups as possible, while not disregarding false alarms: Recall, Precision, Balanced Accuracy, and F_β Score (with beta chosen to be 3). In order to measure overall performance, the AUC-PR is computed during training and after testing. AUC-PR traditionally evaluates the performance of a soft classifier, which is a classifier that predicts probabilities instead of discrete classes. However, by setting a threshold, it can be used as a measure for hard classifiers. The threshold is set at 0.35 with the use of grid-search with estimated hyper-parameter values. Since the AUC-PR, Recall, and Precision are not possible to determine for the chosen baseline, only the Balanced Accuracy and F_β are also computed for the test set. Balanced accuracy is an alternative to the classic accuracy measure which takes the arithmetic mean of sensitivity and specificity. By doing so, it aims to balance out the impact of class imbalance. F_β takes the harmonic mean of recall and precision with parameter β. A higher β increases the weight of the recall as opposed to precision.

2.3.6 Baseline

To evaluate our model using a performance score, a baseline must be established. The objective of this research is not to enhance an existing state-of-the-art model but to introduce a model for detecting pick-ups. To the best of the authors' knowledge, comparing this model to any state-of-the-art model would be improper, due to the inherent differences between them. Hence, we employ the Dutch Draw (DD) [33] as our baseline. This research by Van de Bijl et al. proposes a universal baseline for binary classification models. They propose a method to determine a baseline to facilitate cross-paper comparisons.

2.4 Experimental Design

The trajectories are split up with the sliding window approach. Initial hyperparameter tuning through grid search found the ideal length to be 16 and the ideal overlap to be 50%. Since the sliding window is of size 16, all tracks with fewer than 16 data points are removed. The dataset is shuffled and split into 60% train set, 20% validation, and 20% test set. Almost half of the set is put into validation and testing due to the small amount of positive data points. Each set is made sure to have sufficient positive data points. The variables $Latitude$, $Longitude$, SOG, and COG are put into a four-dimensional vector, to be used as input of the LSTM.

The class weights of the WBCE loss function (represented by the β in equation (2.1)) are computed with a function from the Scikit-learn package [34]. With the addition of these weights, the loss function is weighted such that samples from each class carry equal weight. To prevent overfitting, Early Stopping has been implemented. Early Stopping monitors the model's performance using the validation set. If the AUC-PR validation score has stagnated at a maximum, the model stops training. It is implemented with a patience of 10 epochs.

The final parameters for the model can be observed in Table 2.3.

Table 2.3: Parameters LSTM model. The parameters with an asterisk (*) were found with hyperparameter tuning with the hyperopt package.

Parameter	Chosen Value
Number of layers*	1
Number of units in layer 1*	154
Dropout*	0.64
Learning rate*	0.002
Batch size*	2048
Epochs	300
Window size	16
Overlap between windows	50%

2.5 Results

2.5.1 WBCE Loss

After 500 epochs, the model achieves 0.0091 loss on the training set and 0.0612 on the validation set. Figure 2.3 depicts the progression of the BCE over all the epochs. The training and validation loss curves reach a stable and nearly flat state

at approximately epoch 200. During the epochs that follow, there is a minimal difference in the loss values, indicating that the model learns very little additional information. From that point on, the model is more inclined to overfit if trained longer. The AUC-PR was chosen as the metric for early stopping, thereby decreasing the chance of over-fitting by ensuring that if the validation AUC-PR performance stagnates, the model's learning process comes to a halt.

The plot displaying the validation loss consistently remains slightly higher than the training loss without intersecting. This pattern indicates a well-fit model. Since there is no large gap between validation and training loss, the training set does not seem underrepresented. From the graph, it can be seen that both the validation and training loss oscillate while converging. This is to be expected due to mini-batch gradient descent, a method that divides the training data set into small batches to calculate model error and update model coefficients [35].

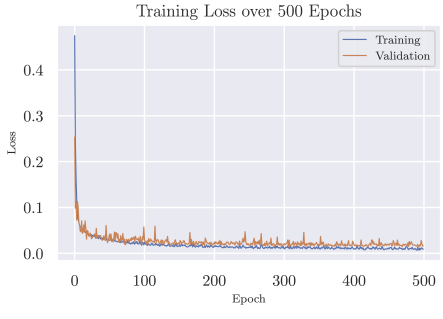


Figure 2.3: Training loss over 500 epochs.

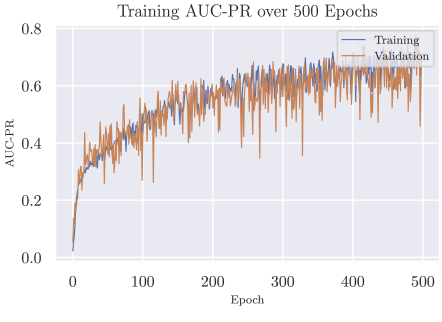


Figure 2.4: AUC-PR score during training over 500 epochs.

2.5.2 Performance Measures

Table 2.4 shows the final values of the performance measures after training and testing. Table 2.5 and Table 2.6 display the confusion matrices after training and testing. A final recall score of 97% is high. It represents the model only missing out on 3% of drop-offs. The final precision score is 71%. It is significantly lower than the recall. This is expected due to the weight in the loss function favoring recall over precision. A precision of 71% indicates that the model classifies 29% movements as pick-ups when the trajectory of the vessel was in reality not normal.

After 500 epochs, the model achieves a ROC-PR score of 71% on the training set and 67% on the test set. Figure 2.4 illustrates the progression of the ROC-PR curve during training. From this curve, it can be seen that the model does not seem to over- or under-fit on the data. The training and validation scores increase but

reach a plateau near the end of the epochs. After testing, the balanced accuracy scores 98% and the F-3 scores 94%.

Table 2.4: Performance of the LSTM model.

Metric	Training	Testing
Recall (TPR)	0.9993	0.9703
Precision (PPV)	0.7391	0.7097
AUC-PR	0.7107	0.7096
Balanced Accuracy	-	0.9832
F-3	-	0.9360

Table 2.5: Confusion matrix after training.

		Predicted	
		Yes	No
Real	Yes	9578	3381
	No	7	956431

Table 2.6: Confusion matrix after testing.

		Predicted	
		Yes	No
Real	Yes	8291	3392
	No	253	849751

2.5.3 Comparison to the Baseline

Table 2.7 presents the results for both the Dutch Draw baseline and the model's score. For the Recall, Specificity, Miss rate, and Fall Out metrics, a direct comparison with the Dutch Draw is not possible, and these are displayed in gray. As can be observed from the table, the model scores higher than the Dutch Draw baseline for all the performance measures.

2.6 Discussion

2.6.1 Major Findings

Four dynamic variables from the historic AIS data have been used. All data from fishing vessels operating off of the Dutch coast in 2022 was used for the neg-

Table 2.7: Results with Dutch Draw scores.

	Dutch Draw		
	argmin$\{\mathbb{E}\}$	argmax$\{\mathbb{E}\}$	Model
Recall	-	-	0.9704
Specificity	-	-	0.9960
Miss Rate	-	-	0.0296
Fall Out	-	-	0.0040
Precision	0.0099	0.0099	0.7097
NPV	0.9901	0.9901	0.9997
FDR	0.9901	0.9901	0.2903
FOR	0.0099	0.0099	0.0003
f_3	0.0910	0.0000	0.9360
J	0.0000	0.0000	0.9664
Markedness	0.0000	0.0000	0.7094
Accuracy	0.9901	0.0099	0.9958
Balanced Accuracy	0.5000	0.5000	0.9832
MCC	0.0000	0.0000	0.8280
Cohen	0.0000	0.0000	0.9958
FM	0.0996	0.0001	0.8298
G-mean 2	0.0000	0.0000	0.9831
Threat Score	0.0099	0.0000	0.6946

ative dataset. The minority positive class contained historic AIS data from 18 known cases. The data was segmented using a sliding window and pre-processed. Subsequently, an LSTM model was tuned and fit to the training segment of the resulting data. The activation function was implemented with a weight to ensure that the model favored detecting pick-ups over missing potential pick-ups. The model achieves a 67% ROC-PR score on the test set, a recall of 97% and a precision of 71%. The model only misses 3% of occurring pick-ups and miss-classifies 29% of normal tracks as suspicious. The model outperforms the baseline model, implying a promising classification model for detecting drop-offs. This result is anticipated and desired. The 29% of normal tracks wrongly classified as suspicious is expected due to the unpredictable behavior of fishing vessels. Furthermore, a human observer will act as a second layer of detection and can disregard the model's positive classification upon further inspection. Despite the model still having a relatively high error score, it significantly reduces the number of data points requiring classification by the observer, thereby enhancing the efficiency of the drop-off detection process.

2.6.2 Limitations

Certain factors must be taken into account when interpreting the model's performance. First, the model is trained on a very small number of negative data points. Given that the model is only trained on 18 examples of suspicious behavior, it is highly probable that a new occurrence may not resemble any previously seen cases. The model could have a significantly lower likelihood of picking up on the anomaly.

Second, the model has been trained on only cases which are known. Since the model is supervised, it will only detect anomalies that resemble previous cases known by police. Therefore, it is possible that unknown instances of successful pick-ups are included in the negative dataset. As a result, the model may have been told certain movements are normal, even though they may actually indicate a pick-up unbeknownst to the police. This is not reflected in the performance measures presented in this research but will become apparent when applied in real life.

Lastly, the model's reliance solely on fishing vessels' data confines its applicability, restricting its use to this specific vessel type. Our aspiration is to develop a more versatile model capable of detecting drop-offs across all types of maritime traffic in the future.

2.7 Conclusion

This research aimed to prove that the implementation of a supervised machine learning model utilizing AIS data will be a significant addition for detecting drop-offs on the coast of The Netherlands. The dynamic variables of AIS data were used and an LSTM model was fit and tuned to this data, giving preference for false alerts over missed pick-up classifications. The model achieved a 67% ROC-PR, 97% recall, and 71% precision score and outperforms the DD baseline.

2.7.1 Future Research

In the application of this model, and in the continuing effort of the police, more positive data points should become available to continue this research. These data points should be used to retrain the model in further research. Due to the decrease in the class imbalance, this has a high chance of increasing the performance of the model. Moreover, the model would become more generalized, with new patterns of other drop-off scenarios added to the data, making it more promising for real-life application.

It would also be quite interesting to look further into the positively classified data points that come from the negative dataset. This might result in finding more positive data points as the model learned that these resembled the known positive data points. Combining the results of our model with an active learning model could make this process of finding more positive data points significantly more efficient.

Another interesting addition to the model could be the use of more features. Existing AIS variables to consider are the HDG and ROT. While these do not serve as the actual heading traveled, they could be indicators of the shipper's intention. Other features to add (outside of AIS features) could be wind direction and current as predictors of expected unusual behavior.

Last of all, the input of the model is an array of the numerical values of the four dynamic AIS variables. Research has shown great potential in using a four-hot encoded vector. This would, however, require a large computational power to handle the amount of memory that this requires.

2.7.2 Concluding

The model developed in this research is a model that is trained for the classification of the 'drop-off' method and that is easily applicable. If implemented as a live model in further application, it could serve as a first layer of a monitoring system, thereby aiding human observers in the detection of illegal activities. The model could also serve as a baseline for future models in detecting pick-ups or, with adequate tuning, be implemented for other illegal route deviation activities such as human trafficking.

Part II

Authorship Verification

3

Combining Style and Semantics for Robust Authorship Verification

Contents

Based on
B. van Leeuwen, S. Bhulai, and R.D. van der Mei, "Combining Style and Semantics for Robust Authorship Verification," *Machine Learning with Applications*, volume 22, December 2025, article 100732.

Abstract

3

Authorship Verification is a key task in Natural Language Processing, essential for applications like plagiarism detection and content authentication. This chapter analyzes the use of deep learning models for Authorship Verification, focusing on combining semantic and style features to enhance model performance. We propose three models: the Feature Interaction Network, Pairwise Concatenation Network, and Siamese Network, which aim to determine if two texts are written by the same author. Each model uses RoBERTa embeddings to capture semantic content and incorporates style features such as sentence length, word frequency, and punctuation to differentiate authors based on writing style. Our results confirm that incorporating style features consistently improves model performance, with the extent of improvement varying by architecture. This demonstrates the value of combining semantic and stylistic information for Authorship Verification. While limitations such as RoBERTa's fixed input length and the use of predefined style features exist, they do not hinder model effectiveness and point to clear opportunities for future enhancement through extended input handling and dynamic style feature extraction. In contrast to prior studies such as [36] and [37], which relied on balanced and homogeneous datasets with consistent topics and well-formed language, our work evaluates models on a more challenging, imbalanced, and stylistically diverse dataset, better reflecting real-world Authorship Verification conditions. Despite the increased difficulty, our models achieve competitive results, underscoring their robustness and practical applicability. These findings support the value of combining semantic and style features for real-world Authorship Verification.

3.1 Introduction

3.1.1 Motivation

Authorship Verification (AV) is a critical task within the field of authorship analysis. Contrary to Authorship Classification (AC), where a model is trained on a predefined set of authors, AV aims to determine whether two texts were written by the same author, even without prior knowledge of that author's writing. Verifying authorship has applications in various fields, including digital forensics [38], literary analysis [39], and copyright infringement detection [40]. Beyond these, AV has also been applied in more specialized domains, such as verifying authorship in song lyrics [41] and conducting cross-lingual AV [42], where stylistic and linguistic variations introduce additional complexity. However, this task remains particularly challenging due to the need for robust generalization beyond known examples, making it an active area of research.

Extensive research has been conducted on AC, as discussed in Subsection 3.1.3 below. The task is framed as a closed-set classification problem, meaning the model is trained and evaluated on a fixed set of known authors. Based on the text, the model recognizes distinct patterns associated with a specific author and uses these style features to learn to differentiate one author from another.

Contrarily, AV involves an open-set problem, where the model encounters authors it has never seen during training. Here lies a considerable challenge: instead of classifying texts into predefined categories, the model must generalize to identify patterns between two texts without having prior knowledge about the authors. This forces AV models to focus on identifying patterns of similarity and divergence between texts, relying heavily on abstract style features. The challenge is magnified when dealing with highly variable writing styles or texts written in different contexts, genres, or time periods.

Despite these hurdles, significant progress has been made in AV through the integration of advanced NLP techniques. One promising approach is to combine deep learning models like RoBERTa [43] with traditional style features, which capture aspects of an author's writing style such as sentence structure, word choice, and punctuation usage. By embedding both semantic content and style features, models can better identify patterns indicative of authorship, even in the absence of direct author-specific training data. This hybrid approach has the potential to bridge the gap between generalizable pattern recognition and fine-grained stylistic analysis, offering a more robust solution to AV.

In this chapter, we propose the combination of RoBERTa-based embeddings with style features to improve the accuracy of AV models. Our goal is to leverage the

strengths of deep learning for semantic understanding while incorporating stylistic information to capture author-specific writing traits. By doing so, we aim to address the core challenge of AV: verifying authorship in a way that generalizes across both known and unseen authors. Through careful experimentation, we demonstrate that combining semantic and style features leads to consistently strong performance across multiple neural architectures. This hybrid approach especially improves recall and F1 scores, showing its effectiveness in accurately identifying same-author pairs, particularly in challenging and imbalanced AV scenarios

3.1.2 Background

Authorship analysis concerns a range of methodologies aimed at understanding and identifying the characteristics of written texts in relation to their authors. This field can be broadly categorized into several sub-disciplines, each regarding different facets of authorship and its implications. Some of these disciplines that are concerned with our task are discussed in this section.

Authorship Classification

AC, in machine learning occasionally also referred to as Authorship Attribution (AA), involves a defined set of authors. The task is to classify a text into one of these known categories. This *closed-set classification* is often more straightforward, as it relies on established patterns and features learned during training. However, it lacks the flexibility required for real-world applications where unknown authors may be encountered.

Authorship Verification

AV focuses on determining whether two texts are written by the same author. Unlike classification, AV deals with an *open-set* scenario, wherein the model must generalize to assess authorship without prior knowledge of the author's specific writing patterns. This task is critical in various fields such as forensic linguistics, where confirming the authorship of a document can have legal implications.

Author Profiling

Author profiling seeks to deduce information about an author based on their writing style and language usage. This can include demographic information such as age, gender, or education level, inferred from textual features. This analysis can be particularly useful in marketing and social-media analysis, where understanding the audience's preferences can drive targeted content creation.

Other

Additionally, the field also touches on phenomena such as obfuscation, plagiarism, multi-author detection [44], and style imitation. These topics delve into how texts can be manipulated or altered, intentionally or otherwise, to obscure their original authorship or to mimic the style of another author. Each of these areas presents unique challenges and necessitates specific analytical approaches.

These various forms of authorship analysis are not mutually exclusive; they often intersect and inform one another. For instance, advancements in AC can enhance AV methodologies, and vice versa. The rapid evolution of computational methods, particularly in NLP and machine learning, has significantly propelled the effectiveness and accuracy of these analyses.

Given the diverse applications and implications of authorship analysis, it is important to understand its different areas. As we direct our approach in AV, we must take advantage of new technologies while also tackling the complexities of language and authorship in today's fast-changing digital world.

3.1.3 Related Work

AV faces several key challenges. AV models focus on consistencies between texts without prior knowledge of the authors. In addition, sufficient writing samples are crucial as writing styles can vary significantly across different contexts, genres, and time periods [45]. Moreover, to prevent overfitting in AV, it is necessary to focus on the author's style and not fixate solely on the genres and topics of the texts.

Recent research has focused on applying NLP techniques to address these challenges, leading to notable advancements in model accuracy, robustness, and generalization across diverse AV tasks

Feature Based Approaches

Traditional AV methods rely on handcrafted linguistic features, such as lexical choices, syntactic structures, and stylistic markers. These features capture distinctive writing habits and have been commonly used in classical machine learning models like Support Vector Machines (SVMs) and Random Forests. However, feature engineering can be time-consuming and may not generalize well across different datasets.

Style techniques have been widely used for AV by analyzing linguistic styles and writing characteristics. While these methods perform well on long texts, they face

significant challenges with short documents, especially in cases involving a large number of authors [46].

Character n-grams are commonly used in both AV and AC [47], [48], [49], [50], [51]. However, such methods can be questionable for AV, because overlapping character sequences are associated with content words. As a result, these methods may focus more on the topic and genre of the text rather than the authors' style, increasing the risk of trained models misclassifying texts with a similar topic to being written by the same author, even though the authors are different.

In AC, a study by [51] investigated various feature extraction techniques, such as Bag-of-Words (BoW), Term Frequency - Inverse Document Frequency (TF-IDF), and n-grams, applied to tweets. Like other studies [52], [53], the TF-IDF method yielded the most desirable performance. However, TF-IDF treats words as independent entities and does not capture the context in which they appear. This is a major drawback for AV, where writing style and contextual patterns are crucial. Moreover, TF-IDF works best with longer texts, whereas AV usually concerns shorter texts. Nonetheless, their findings highlight the value of combining lexical features with contextual representations, supporting the use of deep embeddings and style features in AV.

Deep Learning and Representation Learning

Recent work has increasingly shifted towards deep learning models, particularly those using neural embeddings to capture textual similarities. Transformer-based models such as BERT, RoBERTa, and their variations have been employed to encode texts into dense vector representations [54], [55], [56], [57]. These representations can then be compared using similarity measures or processed through architectures like Siamese Networks (SN). These models leverage contextualized word embeddings to capture subtle stylistic patterns without the need for extensive manual feature extraction, making them well-suited for AV tasks.

[54] adopt a SN initialized with pretrained BERT encoders, with the learning objective designed to map texts by the same author closer in the embedding space and texts by different authors farther apart. This methodology demonstrates the power of pretrained language models in AV, as well as the potential for distance-based loss functions to guide the network towards meaningful representations.

Recent work has explored the combination of deep learning-based embeddings with style-based features for authorship-related tasks. One approach [58] integrates RoBERTa with a Convolutional Neural Network (CNN) to extract contextualized text representations, while also incorporating a writing style module that captures stylistic patterns. These representations are fused and used for AA,

demonstrating the benefits of combining content-based and style-based features. Additionally, a similarity-based approach is employed through cosine distance computations, highlighting the effectiveness of measuring stylistic consistency across texts. These findings suggest that leveraging both linguistic representations and style features can enhance authorship analysis, motivating further exploration in this direction.

Hybrid and Contrastive Learning Approaches

Some studies combine traditional feature-based methods with neural embeddings to enhance model performance. Hybrid models integrate style features with deep learning representations to leverage the strengths of both approaches. Additionally, contrastive learning techniques, which train models to differentiate between similar and dissimilar text pairs, have been explored to improve generalization across unseen authors.

Recent work by [44] proposed style features and RoBERTa for detecting author changes in multi-authored documents, focusing on tasks like single and multiple author-switching detection. While their work aims at document provenance and authentication, our approach leverages similar techniques for AV, demonstrating their versatility in analyzing writing style.

Another study by [59] applied a deep neural network approach to AV, leveraging the T5 language model along with CNNs and an attention mechanism to capture stylistic and semantic features. While their model achieved high accuracy on a manually created test set, its performance on the official PAN dataset highlights the challenge of generalization in AV—an issue our work also aims to address.

PAN Shared Tasks

At PAN 2020, a shared task of AV was tackled [36]. Given two fanfiction texts, the goal was to determine if they were written by the same author. While research in AV with PAN 2020 has achieved high performance metrics [60], [61], [62], these studies rely on well-structured datasets where texts are long, grammatically correct, and often centered around a consistent topic per author. Such datasets can make the AV task easier, as models may partially rely on content-related features such as topic and genre rather than purely on stylistic elements.

Additionally, these studies use artificially balanced datasets with an equal number of positive and negative pairs, not accounting for the real-world challenge where generally positive pairs occur substantially less than negative pairs.

At PAN 2022, another shared task of AV was tackled [37]. Given two texts from

different discourse types, the goal was to determine if they were written by the same author. The dataset is well-structured, consisting only of native English speakers from the same age group, with an equal 50-50 split between positive and negative pairs. PAN 2022 inspired multiple research efforts [59], [63], [64], [65], [66], [67], [68] to tackle the task, with the highest overall performance reaching 0.600 with an F1 score of 0.669 and an AUROC of 0.546. In this shared task, the graph-based approach proved to perform better for longer texts like essays, whereas for shorter texts like emails, business memos, and text messages, the pre-trained language model approaches performed better. The shared task that PAN 2022 provided was challenging, as it involved texts written on different platforms. However, the dataset is well-structured, providing a suitable base to achieve good performance. Still, the overall results of the different methods leave considerable room for improvement. Notably, the results showed that pre-trained language models performed better on shorter texts, suggesting that the assumptions and methods used in PAN 2020 may not generalize well to more diverse and real-world data.

Overall, the PAN 2020 and PAN 2022 shared tasks provide a strong foundation, but show to be not ready for real-life applications yet. A logical next step toward improving the generalizability of AV for real-life scenarios is to combine traditional style models with pre-trained language models.

3.1.4 Our Contribution

The contribution of this chapter is three-fold. First, in contrast to the current state-of-the-art research, this study utilizes a more diverse and noisy dataset of blog texts, where authors write across multiple topics and genres. This setup ensures that our models primarily capture style features rather than topic-driven patterns, making them more robust for real-world applications where genre consistency cannot be assumed.

Second, unlike studies that use artificially balanced datasets with an equal number of positive and negative pairs, we adopt a more realistic 20-80 distribution, where same-author pairs are significantly less frequent. This imbalance better reflects real-world AV scenarios, where the vast majority of comparisons involve texts from different authors. While this setting is inherently more challenging and may lead to lower absolute performance scores, it provides a more reliable evaluation of a model's effectiveness in practical applications.

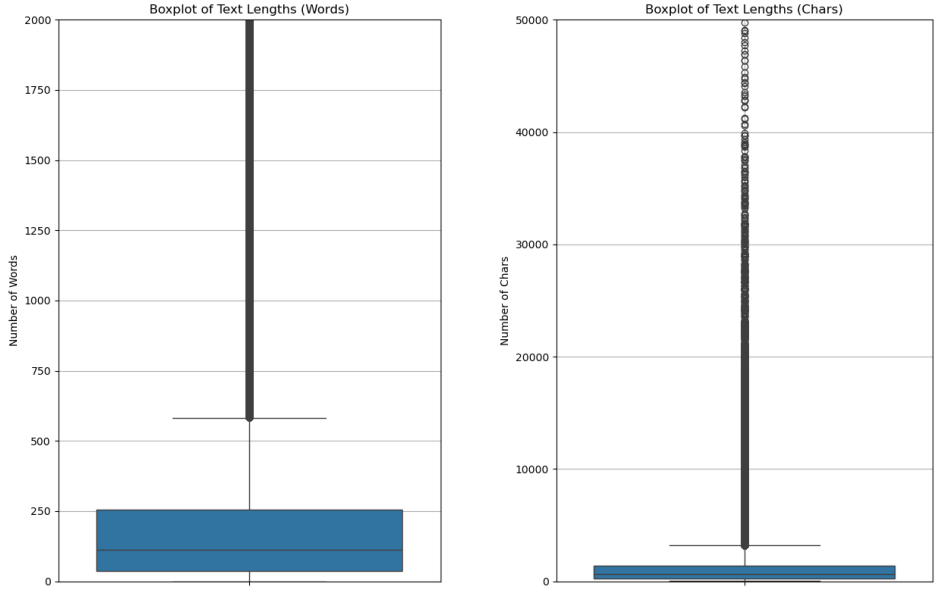
Third, our results demonstrate that incorporating style features significantly improves model performance, reinforcing the importance of linguistic style in authorship analysis beyond content-based cues.

3.2 Data

3.2.1 Data Set

Data Extraction

The dataset used in this study originates from a collection of blogs gathered from Blogger.com in August 2004 by [69]. It contains 681,284 text posts written by 19,320 authors, offering a large-scale and diverse corpus for AV research. Each blog includes all posts from its inception up to the time of collection. The original dataset comprised over 71,000 blogs, filtered to retain only those with at least 200 common English words and available author metadata such as gender and, in some cases, age. To support demographic analyses, a balanced sub-corpus of 37,478 blogs was created, with an equal number of male and female authors across different age groups.



(a) Text length in number of words.

(b) Text length in number of characters.

Figure 3.1: Distribution of text lengths across the dataset measured in words and characters.

The analysis of text lengths across the dataset, as shown in Figures 3.1a and 3.1b, both in terms of words and characters, reveals a highly skewed distribution, with most texts falling within a relatively narrow range and a long tail of extreme outliers. While the majority of documents are under 250 words or 3,000 charac-

ters, there are numerous outliers that significantly exceed these lengths, with the longest text reaching almost 120,000 words. To visualize the distribution of text lengths more effectively, we capped the y-axis at 2,000 words and 50,000 characters, as the presence of extreme outliers distorted the interpretability of the original boxplots.

Challenges

A key challenge in this study arises from the use of RoBERTa, which has a maximum input length of 512 tokens. This constraint can lead to truncation of longer texts, potentially resulting in information loss. While newer transformer models such as Longformer and BigBird address this limitation by supporting longer input sequences, RoBERTa was chosen for its proven performance and compatibility with our architecture. Another significant challenge is the risk of overfitting and data leakage. Since the dataset consists of a limited number of authors, the model could inadvertently learn patterns specific to the training data rather than generalizable style features, reducing its effectiveness on unseen data.

Comparison to Other Data

The BlogAuthorshipCorpus [69] is significantly noisier than state-of-the-art AV datasets such as those from the PAN shared tasks. Unlike PAN datasets, which are carefully curated with balanced author distributions, consistent text lengths, and controlled topical variation, the BlogAuthorshipCorpus consists of informal, user-generated blog posts with high variability in style, topic, and quality. This includes frequent spelling errors, inconsistent punctuation, and wide intra-author stylistic differences. Additionally, the dataset lacks topic control, leading to strong topical signals that may confound stylistic analysis, and includes imbalanced author contributions, further complicating model training and evaluation.

3.2.2 Dataset Preprocessing

To ensure a fair evaluation, we constructed pairs from the blog texts while strictly maintaining author separation between the training and test sets. This guarantees that no author appears in both sets, preventing information leakage and ensuring that the model is evaluated on truly unseen authors. Additionally, to mitigate overfitting, each text post is included in at most one pair, reducing the likelihood that the model memorizes specific examples rather than learning broader stylistic patterns.

3.3 Model Description

3.3.1 Proposed Models

This study employs three different deep neural networks to address the AV problem: a SN, a Feature Interaction Network (FIN), and a Pairwise Concatenation Network (PCN). Each network is designed to implement different mechanisms for comparing text pairs, emphasizing varying aspects of similarity and feature integration. Below, we provide an overview of each architecture.

Siamese Network

The SN architecture consists of two identical subnetworks f_θ, parameterized by shared weights θ, which process the two input texts $\mathbf{x}_1$ and $\mathbf{x}_2$ independently. The network produces embeddings $\mathbf{h}_1 = f_\theta(\mathbf{x}_1)$ and $\mathbf{h}_2 = f_\theta(\mathbf{x}_2)$, where $\mathbf{h}_1, \mathbf{h}_2 \in \mathbb{R}^d$, and d represents the dimensionality of the output embeddings, indicating the number of features each embedding vector contains.

The similarity between the embeddings is then computed using a distance metric D, such as cosine similarity or Euclidean distance:

$$D(\mathbf{h}_1, \mathbf{h}_2) = \|\mathbf{h}_1 - \mathbf{h}_2\|_2 \quad \text{or} \quad \cos(\mathbf{h}_1, \mathbf{h}_2) = \frac{\mathbf{h}_1 \cdot \mathbf{h}_2}{\|\mathbf{h}_1\|\|\mathbf{h}_2\|}.$$

The output of D is passed through a decision layer to predict whether the texts share the same author. The shared weights θ ensure that both inputs are projected into the same feature space, enhancing the model's ability to generalize patterns of similarity.

Feature Interaction Network

The Feature Interaction Network (FIN) explicitly models pairwise interactions between features extracted from the two texts. Let $\mathbf{h}_1, \mathbf{h}_2 \in \mathbb{R}^d$ represent the embeddings of $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively, obtained from a shared encoder f_θ. The interaction is modeled using a pairwise interaction mechanism, such as the outer product or attention mechanisms.

For example, an element-wise interaction can be expressed as:

$$\mathbf{z} = \phi([\mathbf{h}_1; \mathbf{h}_2; \mathbf{h}_1 \odot \mathbf{h}_2]),$$

where $[\cdot;\cdot]$ denotes vector concatenation, $\odot$ represents element-wise multiplication, and ϕ is a transformation (e.g., a feedforward neural network).

Alternatively, an attention-based mechanism may compute the interaction matrix $\mathbf{I} \in \mathbb{R}^{d \times d}$:

$$\mathbf{I} = \text{softmax}\left(\frac{\mathbf{h}_1\mathbf{W}_q \cdot (\mathbf{h}_2\mathbf{W}_k)^\top}{\sqrt{d}}\right),$$

where $\mathbf{W}_q$ and $\mathbf{W}_k$ are learnable projection matrices. These interactions allow the network to capture detailed feature relationships, leading to richer representations.

Pairwise Concatenation Network

In the PCN, the embeddings of the two texts $\mathbf{h}_1$ and $\mathbf{h}_2$, obtained from a shared or independent encoder, are concatenated into a single vector:

$$\mathbf{z} = [\mathbf{h}_1; \mathbf{h}_2],$$

where $\mathbf{z} \in \mathbb{R}^{2d}$.

The concatenated vector is then passed through fully connected layers:

$$\mathbf{o} = \sigma(\mathbf{W}_2\,\text{ReLU}(\mathbf{W}_1\mathbf{z} + \mathbf{b}_1) + \mathbf{b}_2),$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are weight matrices, $\mathbf{b}_1$ and $\mathbf{b}_2$ are biases, σ is the activation function (e.g., sigmoid), and $\mathbf{o}$ is the output. The model directly learns the relationship between the two texts from their joint representation.

By incorporating these mathematical formulations, the networks emphasize different aspects of feature extraction, interaction, and relationship modeling, enabling a comprehensive evaluation of their suitability for the AV task.

3.3.2 Style Features

To complement the semantic representations, we extracted a set of style features designed to reflect writing style independently of topic. We extracted the Flesch reading ease score ([70]) to measure how readable a text is. Average sentence length was used to capture syntactic complexity, while counts of nouns and verbs reflected grammatical tendencies. We also calculated the stopword ratio to account for function word usage and the average word length as a proxy for lexical sophistication. All texts were processed using SpaCy for accurate tokenization and sentence segmentation. The resulting features were normalized using a standard scaler to allow comparison across features. While we did not conduct formal feature importance analyses such as SHAP in this study, our primary aim was not to optimize a handcrafted feature set, but to evaluate whether even simple, interpretable style features can provide complementary information when combined with deep semantic embeddings. This choice of handcrafted features also reflects

a deliberate preference for interpretability and computational efficiency, which is especially important in resource-constrained or transparency-critical contexts such as forensic linguistics. To integrate stylistic and semantic information, the style features were concatenated with the RoBERTa embeddings at the input level, forming a unified representation for each text. This allowed for clear isolation of the contribution of traditional style features when combined with semantic representations.

3.3.3 RoBERTa

Adding semantic information helps the model capture meaningful differences in how authors express themselves. While style features are valuable for capturing consistent stylistic patterns, they do not fully account for semantic content. Although most sentence embedding models, including RoBERTa, are designed to focus on the semantic meaning of sentences and generally de-emphasize function words such as stopwords, semantic embeddings can still indirectly reflect stylistic consistencies—especially when combined with style features or fine-tuned for the authorship verification task. It is important to clarify that RoBERTa primarily captures what is said rather than how it is said. Therefore, the combination of semantic embeddings with explicit style features allows the model to better leverage both the content and style dimensions necessary for robust authorship verification across different topics.

RoBERTa (Robustly Optimized BERT Pretraining Approach) is an advanced transformer-based language model introduced by [43]. It builds upon BERT by optimizing its training process through dynamic masking, removal of the next sentence prediction objective, and training on larger datasets with increased batch sizes. These improvements enhance RoBERTa's ability to capture contextual and semantic relationships in text, making it particularly effective for various NLP tasks, including AV. By leveraging pre-trained representations from RoBERTa, AV models can benefit from deep contextualized embeddings, allowing for more robust comparisons of writing style and linguistic patterns.

RoBERTa can only be used on a maximum number of characters/words. By taking only that number of words from the texts, the model generalizes on any length of text, short and long. However, a bit of information might get lost from the long texts.

3.3.4 Loss Function

The Weighted Binary Cross-Entropy (WBCE) is an adaptation of the standard Binary Cross-Entropy (BCE) loss function, designed to address issues arising

from class imbalance in binary classification tasks. In many real-world scenarios, the distribution of classes is skewed, meaning that one class may be significantly more frequent than the other. This can lead to models that perform well on the majority class but poorly on the minority class. WBCE seeks to mitigate this issue by assigning different weights to the two classes. Consequently, the introduction of weights enhances model performance. WBCE allows the model to achieve better recall and precision for the underrepresented class, improving overall classification performance in imbalanced scenarios. The WBCE loss function ultimately leads to better generalization and predictive accuracy concerning our task.

The standard BCE loss function is defined as:

$$\text{BCE}(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right], \tag{3.1}$$

where:

- y_i is the true label (0 or 1).
- $\hat{y}_i$ is the predicted probability of the positive class (1).
- N is the total number of samples.

This loss function effectively measures the performance of a classification model whose output is a probability value between 0 and 1. To employ WBCE, we introduce weights w_0 and w_1 to penalize the model differently for misclassifying the two classes. The WBCE loss is then defined as:

$$\begin{aligned} \text{WBCE}(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} \Big[& w_1 \cdot y_i \log(\hat{y}_i) \\ & + w_0 \cdot (1 - y_i) \log(1 - \hat{y}_i) \Big], \end{aligned} \tag{3.2}$$

where:

- w_1 is the weight for the positive class (1).
- w_0 is the weight for the negative class (0).

3.4 Experimental Design

In this section, we outline the experimental setup for the networks evaluated in this study, including choices for hyperparameters, loss functions, and evaluation metrics. We use three neural network architectures: FIN, PCN, and SN. To ensure a fair and accurate comparison between the models, we conducted a separate grid

search for each model. This helped us find the best hyperparameter settings, such as learning rate, dropout rate, batch size and number of epochs, for each model individually. This individualized tuning accounts for the distinct optimization dynamics of each architecture, ensuring stable training and fair performance comparison. Models were also tuned separately for configurations with and without style features, as the inclusion of style-based information alters input dimensionality and learning behavior. This approach ensures that all models are evaluated under conditions that reflect their true potential, avoiding bias due to suboptimal training settings. The final architectural and training choices for all six configurations are summarized in Table 3.1.

Table 3.1: Optimal hyperparameters per model configuration, selected through grid search.

Model	Style	Learning Rate	Dropout	Batch Size	Epochs
Feature Interaction	No	$\exp(-4)$	0.01	16	15
Feature Interaction	Yes	$\exp(-3)$	0.01	32	15
Pairwise Concatenation	No	$\exp(-4)$	0.1	16	10
Pairwise Concatenation	Yes	$\exp(-4)$	0.1	32	15
Siamese Network	No	$\exp(-3)$	0.01	16	10
Siamese Network	Yes	$\exp(-3)$	0.01	16	15

3.4.1 Siamese Network

The SN is a deep learning model designed for comparing two input texts. It utilizes a shared base network that processes both inputs in parallel. The base network consists of dense layers with ReLU activations, followed by batch normalization and dropout to mitigate overfitting. The output of both inputs is a fixed-size representation that is compared using cosine similarity, calculated as the dot product between the processed outputs of the two inputs. The similarity score is then scaled, using a Lambda layer [71]. The final output layer uses a sigmoid activation function to classify whether the two inputs are from the same author.

Hyperparameters: The SN is trained using the Adam optimizer with a learning rate of 1×10^{-3} for both the model with and without style features. The model with style features is trained for 15 epochs, while the model without style features is trained for 10 epochs. Both models use a batch size of 16. To address class imbalance, class weights are calculated and applied during training. Dropout is applied with a rate of 0.01 in both models' base networks to prevent overfitting. Additionally, L2 regularization with a coefficient of 0.0001 is used to encourage weight sparsity and mitigate overfitting.

Model Architecture: The Siamese Network (SN) consists of a base network

$f(\mathbf{x})$, which maps the input text embeddings to a latent space. The base network is composed of three fully connected layers:

- A dense layer with 256 units, ReLU activation, batch normalization, dropout, and L2 regularization.
- A dense layer with 128 units, ReLU activation, batch normalization, dropout, and L2 regularization.
- A dense layer with 64 units, ReLU activation, batch normalization, dropout, and L2 regularization.

The two input embeddings $\mathbf{x}_1$ and $\mathbf{x}_2$ are processed independently through the base network, yielding embeddings $f(\mathbf{x}_1)$ and $f(\mathbf{x}_2)$. The network then computes the cosine similarity between these embeddings using the dot product:

$$\text{Cosine Similarity} = \frac{f(\mathbf{x}_1) \cdot f(\mathbf{x}_2)}{\|f(\mathbf{x}_1)\|_2 \|f(\mathbf{x}_2)\|_2}.$$

The output is a similarity score, which is transformed to a range of [0, 1] using a Lambda layer and is used for classification.

3.4.2 Feature Interaction Network

The network operates by taking two input texts, transforming them into embeddings, and performing several element-wise operations (subtraction, absolute difference, and multiplication). These interactions are concatenated and passed through a series of fully connected layers with ReLU activations. Dropout is applied to the hidden layers to prevent overfitting. The output layer uses a sigmoid activation function to predict whether the two texts come from the same author. The network is optimized using a custom weighted BCE loss to handle class imbalance effectively.

Hyperparameters: The FIN is trained with a learning rate of 1×10^{-4} (0.0001) using the Adam optimizer, which adapts the learning rate during training and is suitable for minimizing complex, non-convex loss functions. The model is trained for a total of 15 epochs, with a batch size of 16, balancing training time and memory usage. Dropout is applied at a rate of 0.01 after each dense layer to prevent overfitting. The model is trained without style features. For the version of the FIN with style features, the learning rate is set to 1×10^{-3} (0.001). The model is trained with a batch size of 32 for 15 epochs, using a dropout rate of 0.01 after each dense layer to avoid overfitting.

Model Architecture: The model consists of two inputs, each representing one text in the pair, which are processed using simple arithmetic operations to model their interaction. Specifically, we compute the element-wise difference, absolute

difference, and multiplication of the input vectors. These interaction features are then concatenated and passed through three fully connected layers with 4096, 256, and 64 units, respectively. Each layer uses the ReLU activation function, followed by dropout to regularize the model. The final output layer has a sigmoid activation to produce a binary classification prediction. No attention mechanisms are used; the model relies entirely on these arithmetic operations to capture both symmetric and asymmetric relationships between the paired inputs.

3.4.3 Pairwise Concatenation Network

The PCN operates by taking two input texts, transforming them into embeddings, and concatenating these embeddings along the feature dimension. The concatenated representation is then passed through three fully connected layers, each with ReLU activation. Dropout is applied after each layer to mitigate overfitting. The final output layer uses a sigmoid activation function for binary classification, predicting whether the two texts come from the same author.

Hyperparameters: The model is trained using the Adam optimizer with a learning rate of 1×10^{-4}, both with and without style features. The model with style features is trained for 15 epochs with a batch size of 32, while the model without style features is trained for 10 epochs with a batch size of 16. To address class imbalance, class weights are calculated and applied during training. Dropout is applied with a rate of 0.1 in both models.

Model Architecture: The network begins with two input layers, $input_a$ and $input_b$, each of shape (input_shape). These embeddings are concatenated along the last axis, forming a tensor of shape $(2 \times \text{input\_shape})$. The concatenated vector is passed through three fully connected layers with 4096, 256, and 64 units, respectively. Each of these layers uses the ReLU activation function, and dropout is applied after each layer to prevent overfitting. The final output layer uses a sigmoid activation function to perform binary classification.

The model is compiled using the Adam optimizer, and a custom WBCE loss function addresses class imbalance during training.

3.4.4 Training and Validation

All networks are trained and evaluated on the same training and test datasets. The training dataset is used to fit the models, while the test dataset is used to assess generalization. The models are evaluated at the end of each epoch on the test set, and the best performing model based on validation accuracy is selected.

3.4.5 Loss Function

To address class imbalance, the network is trained with a custom weighted BCE loss function, designed to optimize for the pairwise similarity task while addressing class imbalance. The WBCE loss is shown in Equation (3.2).

We apply a fixed weighting scheme where misclassifying the minority class is penalized more heavily than the majority class. We apply the WBCE Loss where false negatives receive a weight of 10, while false positives receive a weight of 1. This encourages the model to focus more on correctly identifying the minority class while still considering the majority class.

3.4.6 Evaluation Metrics

The model is evaluated using a variety of metrics to provide a comprehensive performance evaluation:

- **Accuracy:** Measures the overall correctness of the model's predictions.
- **Precision:** The proportion of true positives among all predicted positives.
- **Recall:** The proportion of true positives among all actual positives.
- **F1 Score:** The harmonic mean of precision and recall.
- **AUC-PR:** The area under the precision-recall curve, which provides a more informative evaluation in imbalanced classification problems.

3.5 Results

We evaluated three models for the AV task: FIN, SN, and PCN. Each model was tested with and without the inclusion of style features, which capture authorial writing patterns beyond the semantic content encoded in transformer embeddings. Below, we report and analyze the final test results for all model variants.

3.5.1 Performance Measures

We report six evaluation metrics for each model variant: loss, accuracy, precision, recall, F1 score, and area under the ROC curve (AUC). These metrics are presented in Table 3.2, allowing for a detailed comparison of performance between versions with and without style features. Each column corresponds to one of the three model architectures—FIN, SN, and PCN—evaluated with and without style information. The results provide a comprehensive view of the models' ability to distinguish between same-author and different-author text pairs, and highlight the effects of including surface-level writing features alongside contextual embeddings.

Table 3.2: Final test set performance across all models with and without style features.

Metric	Feature Interaction (No Style)	Feature Interaction (Style)	Siamese Network (No Style)	Siamese Network (Style)	Pairwise Concatenation (No Style)	Pairwise Concatenation (Style)
Loss	1.4631	**1.4409**	1.5475	1.4745	1.4521	1.4526
Accuracy	0.8263	0.8281	0.8383	**0.8472**	0.8311	0.8292
Precision	0.7183	0.7058	0.7326	0.7402	**0.7621**	0.7049
Recall	0.6681	0.6961	0.7107	0.7265	0.6121	**0.7052**
F1 Score	0.6791	0.6906	0.7085	0.7228	0.6632	**0.6942**
AUC	0.7995	0.8062	0.8114	0.8260	0.7879	**0.8015**

In addition to summary statistics, we visualize classifier performance using receiver operating characteristic (ROC) curves for all model variants in Figure 3.2. The ROC curves provide a comparative view of true positive rate versus false positive rate across different decision thresholds. As shown, all models outperform the random baseline, with the SN with style model achieving the highest overall performance (AUC = 0.89), followed by the SN and FIN models (both AUC = 0.88). Incorporating style features consistently improves or maintains performance across all architectures.

To monitor model learning behavior, we tracked accuracy, recall, precision, F1 score, and AUC across training epochs. Training and validation curves for each metric are provided in 3.D, illustrating convergence patterns and the effects of style features. These plots serve to complement the final test results in Table 3.2.

For a detailed view of the classification outcomes, the confusion matrices for all models—both with and without style features—are included in 3.B. These matrices illustrate differences in false positive and false negative rates across models and show how the inclusion of style features affects prediction balance.

3.5.2 Comparison to (PAN) Baselines

Our models were evaluated in a more challenging setting than the ones used in PAN 2020 and PAN 2022, where datasets were artificially balanced, texts were well-structured, and authors often wrote about consistent topics in grammatically correct English. In contrast, our dataset contains platform-agnostic, stylistically diverse texts with realistic class imbalance, closely resembling practical AV scenarios.

The PAN 2020 shared task [36] featured fanfiction texts and offered both a large and small dataset for training. The top-performing model ([60]) achieved an AUC of 0.969 and an F1-score of 0.936. However, this setting favors high performance

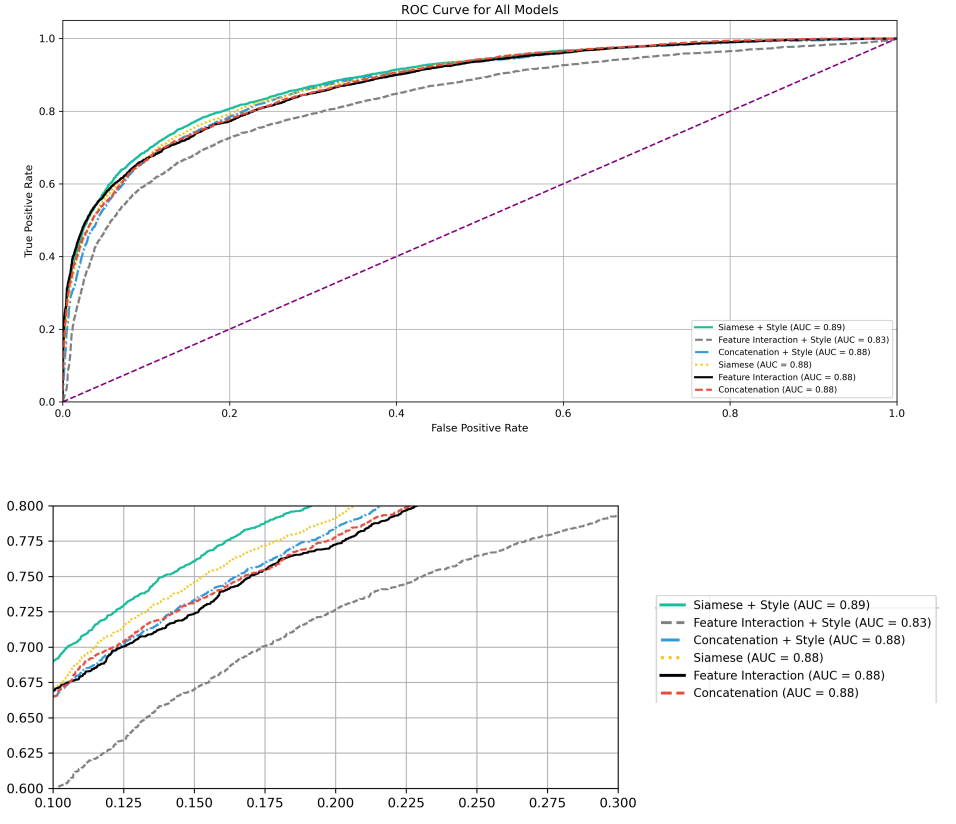


Figure 3.2: ROC curves comparing all model variants (top), with detailed views shown below.

due to homogeneous text structure, topic consistency, and ample training data. In our case, the best model—PCN with style features—achieved an AUC-PR of 0.7953 and an F1-score of 0.6782, reflecting solid performance in a much less constrained and noisier environment.

Compared to PAN 2022 [37], which introduced discourse variation across platforms but still maintained balance and demographic uniformity, our models also outperform the top results. The best F1-score at PAN 2022 was 0.669, with an AUROC of 0.546. Our models, trained on heterogeneous data, achieved a higher F1-score and precision-recall AUC, emphasizing improved real-world applicability.

3.5.3 Performance vs Text Length

To investigate whether the models exhibit any bias related to input text lengths, we computed Pearson correlation coefficients between three variables: average input length (`avg_len`), absolute length difference between the two texts (`length_diff`), and whether the prediction was correct (`correct`). The tables in 3.A present these correlation matrices for the six different models. Across all models, the correlations between the correctness of a prediction and both length-based metrics are consistently close to zero, indicating no practical dependence on either input size or imbalance in length.

3.5.4 Topic Similarity Analysis Across Models

Specifically, we investigated the relationship between topic similarity scores and the models' mean prediction probabilities. Topic similarity was computed using two approaches: (1) LDA-based similarity, where documents were represented by their topic distributions obtained from a Latent Dirichlet Allocation model [72], and similarity between document pairs was measured using cosine similarity on these distributions; and (2) RoBERTa-based similarity, where sentence embeddings from a pre-trained RoBERTa model were used, and similarity between document pairs was computed via cosine similarity on these embeddings.

Across all six model variants, LDA-based topic similarity revealed only weak associations with model outputs. The highest Spearman correlation was observed for the Siamese network without style features ($\rho \approx 0.425$, $R^2 \approx 0.164$), while the lowest correlations occurred for pairwise concatenation and feature interaction models with style features ($\rho \approx 0.256$–0.275, $R^2 \approx 0.099$–0.102). In general, including style features slightly reduced the correlation with topic similarity, suggesting that stylistic cues contribute to predictions beyond topical content.

RoBERTa-based similarity showed consistently low correlations for all models ($\rho \leq 0.058$, $R^2 \leq 0.115$), reflecting saturation near high similarity values. This indicates that semantic embeddings in this context do not meaningfully distinguish between model prediction scores.

Overall, these results indicate that topical similarity contributes only modestly to model predictions, and that higher predictions are not simply a result of overlapping topics. The weak correlation across models implies that the models, particularly when incorporating style features, rely more heavily on other textual characteristics—likely stylistic or deeper semantic patterns—rather than surface-level topical overlap. This provides confidence that the models are capturing nuanced signals relevant to authorship verification rather than relying solely on topic content.

For further clarity, detailed plots of the topic similarity distributions, along with a table summarizing the corresponding values, are provided in 3.C.

3.6 Discussion

This study set out to address the challenges of AV in real-world scenarios by evaluating multiple model architectures and exploring the value of incorporating style features alongside contextual embeddings. Unlike previous AV benchmarks, such as those demonstrated at PAN 2020 and PAN 2022, which focused on well-curated datasets with controlled topics, consistent grammar, and balanced class distributions, our approach dealt with a more complex and realistic setting. Texts originated from multiple platforms and exhibited considerable variation in discourse type, writing style, and structure. Moreover, the class distribution was intentionally imbalanced to reflect the rarity of positive (same-author) cases in real-life applications.

Our findings highlight a couple of main contributions. First, all three evaluated models—FIN, SNs, and PCN—achieved solid performance in a cross-platform AV setting. Among these, the SN model yielded the highest accuracy (0.8472), indicating strong overall predictive performance. However, the FIN model with style features achieved the best results in F1 score (0.7228), recall (0.6961), and AUC (0.8260), highlighting its strength in correctly identifying positive matches. These findings suggest that while the SN architecture excels at general classification, the FIN model, when enhanced with style cues, is particularly effective at capturing author-specific patterns crucial for accurate AV.

Second, the integration of style features consistently improved recall across all models, which is particularly relevant in operational scenarios where minimizing false negatives is important. Surface level features, such as sentence length, lexical variety, and punctuation use, provide complementary information that enhances the discriminative power of contextual embeddings, supporting the use of hybrid models that combine deep language representations with explicit stylistic indicators. Overall, models incorporating style features performed better across most metrics, achieving the highest accuracy, recall, F1 score, and AUC. This suggests that style features enhance the model's ability to capture author-specific characteristics, which are essential for AV tasks. While precision was slightly better in non-style models, recall, being more critical for identifying true matches, was significantly higher in style-based models. The improved recall and F1 score in these models make them more effective for AV, as they are better at correctly identifying matches and minimizing false negatives.

Third, while benchmark datasets like PAN 2020 and PAN 2022 report higher res-

ults in more controlled settings, our models are evaluated in a more challenging and realistic scenario. The best-performing model achieves an F1 score of 0.7228 and an AUC score of 0.8260, demonstrating improved robustness compared to PAN 2022 and a stronger trade-off between precision and recall. These results underscore the practical applicability of the proposed methods to real-world AV tasks.

Finally, this study outlines a flexible framework for building AV systems that generalize across platforms, domains, and unknown authors. By reframing the task as one of measuring stylistic similarity, rather than closed-set classification, this work contributes to the development of more interpretable and adaptable solutions for forensic and large-scale AV applications.

3.6.1 Major Findings

The results demonstrate that incorporating style features consistently improves the performance of all three model architectures across most evaluation metrics. Notably:

- The SN with style features achieved the highest accuracy (0.8472) and F1 score (0.7228) among all models, indicating that this combination is particularly effective at identifying authorship similarity when stylistic nuances are considered.
- Although the PCN model (without style features) had the highest precision (0.7621), it showed significantly lower recall (0.6121), suggesting that it was more conservative in making positive predictions. Once style features were added, recall improved to 0.7052, resulting in a more balanced performance across precision and recall.
- The FIN model with style features showed consistent improvements over its no-style counterpart in recall (from 0.6681 to 0.6961), F1 score (from 0.6791 to 0.6906), and AUC (from 0.7995 to 0.8062), although precision slightly decreased. This suggests that while the model became marginally less precise, it became more effective at identifying true positives, making it more suitable for real-world AV tasks where recall is crucial.

Overall, adding style features benefits recall and F1 score the most, indicating that style information helps models better generalize over variations in writing, particularly in subtle cases where surface-level content similarity may be insufficient. Importantly, in AV, recall is a critical metric, as it reflects the model's ability to correctly identify true same-author pairs. A higher recall means fewer false negatives, which is essential in applications where missing a true authorship match could undermine trust in the system or lead to incorrect conclusions in forensic or

security-related contexts.

The FIN generally has higher complexity than SN due to the explicit interaction layers that compute pairwise combinations between features of both inputs. This added complexity results in longer training times for FIN, larger model size due to more parameters in the interaction layers and higher memory usage during both training and inference. However, this added cost is minimal and leads to a notable gain in recall. Across all variants tested (with and without style), FIN consistently outperformed SN in recall. We consider that the increased recall from FIN's added complexity is a worthwhile trade-off, especially in high-stakes scenarios such as authorship verification or forensic analysis, where missing a true positive can have serious consequences. While FIN does increase computational cost, it remains feasible for near-real-time use on modern hardware, and further optimizations, such as model pruning or batching, can help reduce latency.

3.6.2 Limitations

The proposed model, while effective, has certain limitations that should be acknowledged. The process of extracting style features from texts is based on a predetermined set of features, which may not capture the full complexity of an author's writing style. This reliance on predefined features can limit the model's ability to fully understand subtle or complex stylistic nuances. As a result, the model may miss important aspects of authorship that are not represented by the extracted features.

RoBERTa, like other transformer-based models, processes only a limited number of tokens at a time, typically constrained by the model's input size (e.g., 512 tokens). This means that each datapoint has a fixed maximum length, and texts longer than this limit are trimmed. Consequently, important contextual or stylistic information may be lost if the input text exceeds this length, potentially affecting the accuracy of the AV task.

3.6.3 Future Research

Future research could focus on several directions to enhance the current work and address the limitations identified in this study. One potential avenue for improvement is to capture more comprehensive information from the entire text. Currently, due to RoBERTa's token length limitation, only a portion of the text (e.g., the first 512 tokens) is processed. By extending the model's ability to handle longer texts, we could better capture the full context and style features of a document, leading to more accurate AV.

We could also explore the application of more advanced feature extraction techniques for short texts, such as those using deep learning models like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), which have demonstrated effectiveness in authorship attribution tasks for informal texts like tweets [73].

Another area for future research is the expansion of the style features extracted from the texts. At present, only a minimal set of features is used to represent an author's style. By incorporating a broader range of style features, such as measures of lexical diversity, discourse structure, or punctuation usage, the model could achieve a more nuanced understanding of an author's writing patterns, which might improve performance, particularly in more challenging cases. We also recognize the value of neural style embeddings (e.g., from BERT or similar models), and incorporating these richer representations of style is a natural next step. Our current findings already show that even basic stylometric features can enhance performance, suggesting that more sophisticated style modeling could improve results further.

Moreover, while our architecture allows the networks to implicitly learn interactions between style and semantic signals, no explicit analysis was performed to isolate or visualize the contribution of individual style features in relation to semantic components. Future work may investigate such interactions using attention mechanisms or feature attribution methods to better understand the interplay between writing style and meaning, which could enhance interpretability and model robustness.

A further valuable extension would be to develop a model whose architecture or loss function dynamically adapts based on the degree of class imbalance. This could involve incorporating imbalance-aware sampling strategies, cost-sensitive learning, or designing auxiliary components that assess and adjust decision thresholds according to the observed class ratio. By making the model's behavior a function of class distribution, rather than assuming uniformity, we may improve its robustness and generalization to real-world deployment settings where true positive cases are scarce and verification is critical.

In future work, we also aim to evaluate our approach on the PAN authorship verification datasets to enable direct comparison with existing state-of-the-art methods. This would provide a more standardized benchmark and facilitate a clearer assessment of our model's generalizability.

Given the generalizability of the models developed in this study, there is potential for their application across different domains. With small adjustments, we believe that the methods could be adapted for AV tasks in other contexts, such as academic

writing, social-media posts, or literary works. Exploring cross-domain generalizability would provide valuable insights into the robustness and adaptability of the proposed methods in various settings.

3.7 Conclusion

In this study, we proposed and evaluated several models for AV and authorship attribution, including the FIN, PCN, and SN. We analyzed the incorporation of style features into the models and the impact of these features on model performance. Our experiments demonstrated the potential for improving AV by combining semantic embeddings with style features, especially when addressing class imbalance through weighted loss functions.

Although our models demonstrated strong performance, several limitations were identified, including the constraint of using predefined style features and the limited text length due to the nature of RoBERTa embeddings. These limitations suggest areas for future improvements, such as extending the input text length or incorporating additional style features to enrich the models further.

Overall, this research highlights the importance of combining linguistic features with deep learning models to improve the accuracy of AV. The approach is adaptable across domains, and with further adjustments, it can be generalized to tackle other text classification tasks. Our hybrid models mark a significant advancement toward developing robust AV systems suitable for practical deployment.

Future work could focus on refining the models' generalization capabilities, exploring more advanced feature extraction techniques, and extending the approach to longer and more complex text corpora.

Appendix

3.A Length-Related Bias in Models with Style Features

This section presents correlation matrices examining length-related biases in all models, both with and without style features. The matrices show relationships between average text length, length difference between text pairs, and prediction correctness, providing insight into how length influences model performance.

Table 3.A.1: Correlation matrix for Pairwise Concatenation (without style).

	avg. length	length diff	correct
avg. length	1.000000	0.878600	0.023731
length diff	0.878600	1.000000	-0.005317
correct	0.023731	-0.005317	1.000000

Table 3.A.2: Correlation matrix for Feature Interaction (without style).

	avg. length	length diff	correct
avg. length	1.000000	0.878600	0.008540
length diff	0.878600	1.000000	-0.016155
correct	0.008540	-0.016155	1.000000

Table 3.A.3: Correlation matrix for Siamese (without style).

	avg. length	length diff	correct
avg. length	1.000000	0.878600	0.010822
length diff	0.878600	1.000000	-0.012645
correct	0.010822	-0.012645	1.000000

Table 3.A.4: Correlation matrix for Pairwise Concatenation (with style).

	avg. length	length diff	correct
avg. length	1.000000	0.903202	0.013779
length diff	0.903202	1.000000	-0.008542
correct	0.013779	-0.008542	1.000000

Table 3.A.5: Correlation matrix for Feature Interaction (with style).

	avg. length	length diff	correct
avg. length	1.000000	0.903202	0.008439
length diff	0.903202	1.000000	-0.000517
correct	0.008439	-0.000517	1.000000

Table 3.A.6: Correlation matrix for Siamese Network (with style).

	avg. length	length diff	correct
avg. length	1.000000	0.903202	0.017153
length diff	0.903202	1.000000	0.001718
correct	0.017153	0.001718	1.000000

3.B Confusion Matrices

Tables 3.B.1 – 3.B.3 present confusion matrices for the three evaluated models—Feature Interaction, Pairwise Concatenation, and Siamese—both with and without the inclusion of style features. These matrices provide detailed insights into the classification performance and error distribution for each configuration.

Table 3.B.1: Confusion matrices for the Feature Interaction model.

Without Style

		Predicted	
		Yes	No
Real	Yes	3695	1185
	No	1973	9632

With Style

		Predicted	
		Yes	No
Real	Yes	3114	1766
	No	1473	10000

Table 3.B.2: Confusion matrices for the Pairwise Concatenation model.

Without Style

		Predicted	
		Yes	No
Real	Yes	3557	1323
	No	1651	9954

With Style

		Predicted	
		Yes	No
Real	Yes	3774	1106
	No	2127	3774

Table 3.B.3: Confusion matrices for the Siamese model.

Without Style

		Predicted	
		Yes	No
Real	Yes	3805	1075
	No	2086	9519

With Style

		Predicted	
		Yes	No
Real	Yes	3818	1062
	No	1944	9661

3.C Topic Bias Plots

Figures 3.C.1 – 3.C.6 displays topic bias plots for all three models, both with and without style features. These plots illustrate the relationship between topic similarity and the models' mean predictions, highlighting how topic influence varies across different model configurations.

Table 3.C.1: Correlation between model predictions and topic similarity across models. R^2 indicates the proportion of variance explained, and ρ is Spearman's rank correlation.

Model	Style Features	R^2	ρ (Spearman)
Pairwise Concatenation (PCN)	No	0.125	0.327
Pairwise Concatenation (PCN)	Yes	0.102	0.256
Feature Interaction (FIN)	No	0.141	0.305
Feature Interaction (FIN)	Yes	0.099	0.275
Siamese Network (SN)	No	0.164	0.425
Siamese Network (SN)	Yes	0.148	0.394
RoBERTa SN	No	0.115	0.058
RoBERTa SN	Yes	0.069	0.056
RoBERTa FIN	No	0.052	0.030
RoBERTa FIN	Yes	0.052	0.042
RoBERTa PCN	No	0.035	0.027
RoBERTa PCN	Yes	0.045	0.036

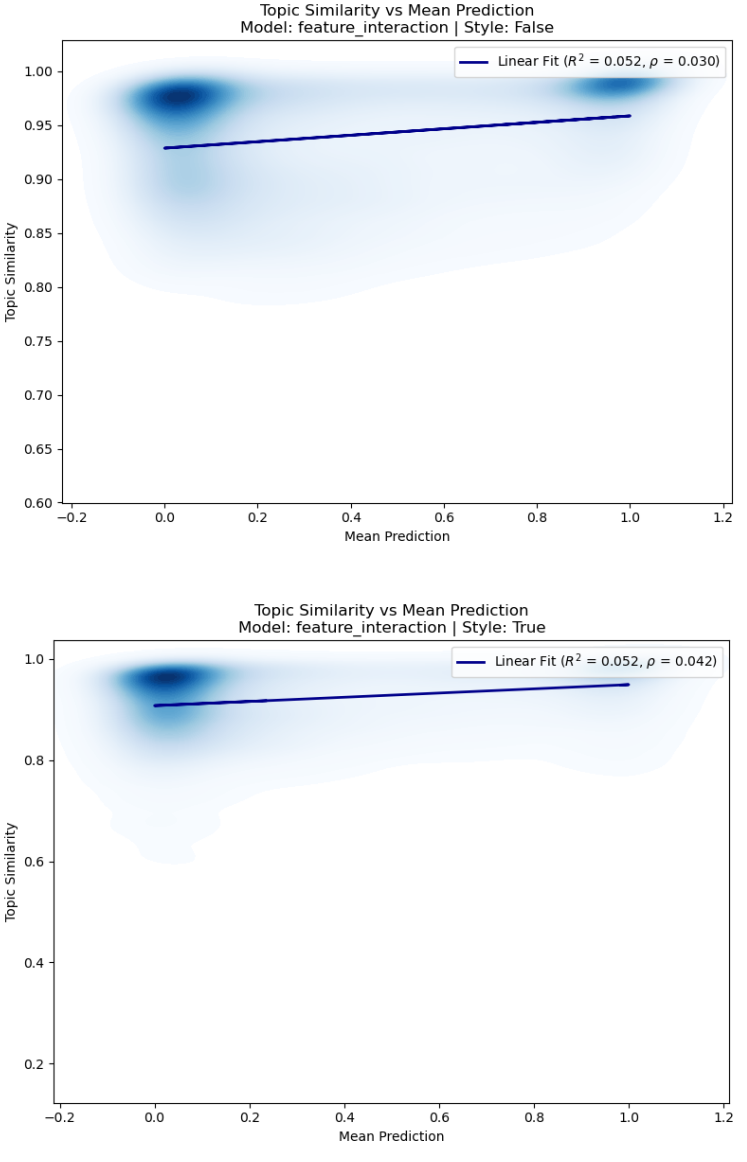


Figure 3.C.1: Feature Interaction model performance with and without style features based on RoBERTa topic similarity. The top figure shows the model without style features, where a weak positive relationship between topic similarity and mean prediction is observed. The bottom figure includes style features, where the influence of topic similarity remains weak.

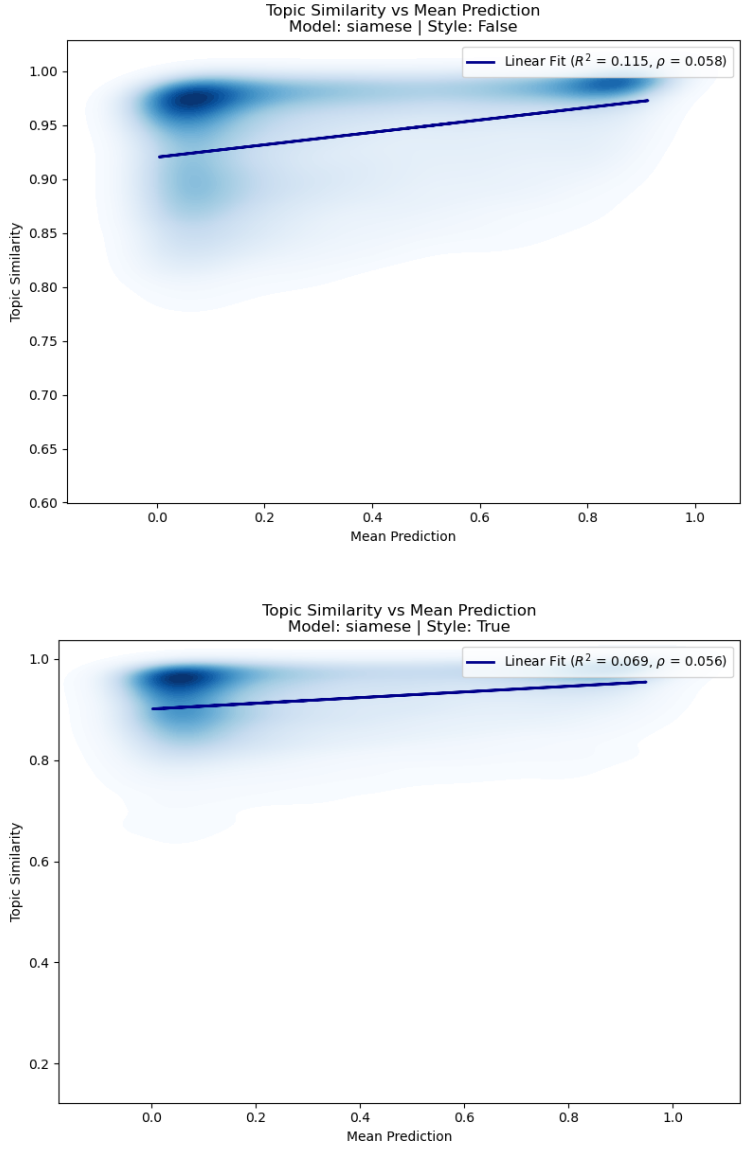


Figure 3.C.2: Siamese model performance with and without style features based on RoBERTa topic similarity. The top figure shows the model without style features, which exhibits the strongest topic influence among the evaluated models. The bottom figure includes style features, where the influence of topic similarity is reduced.

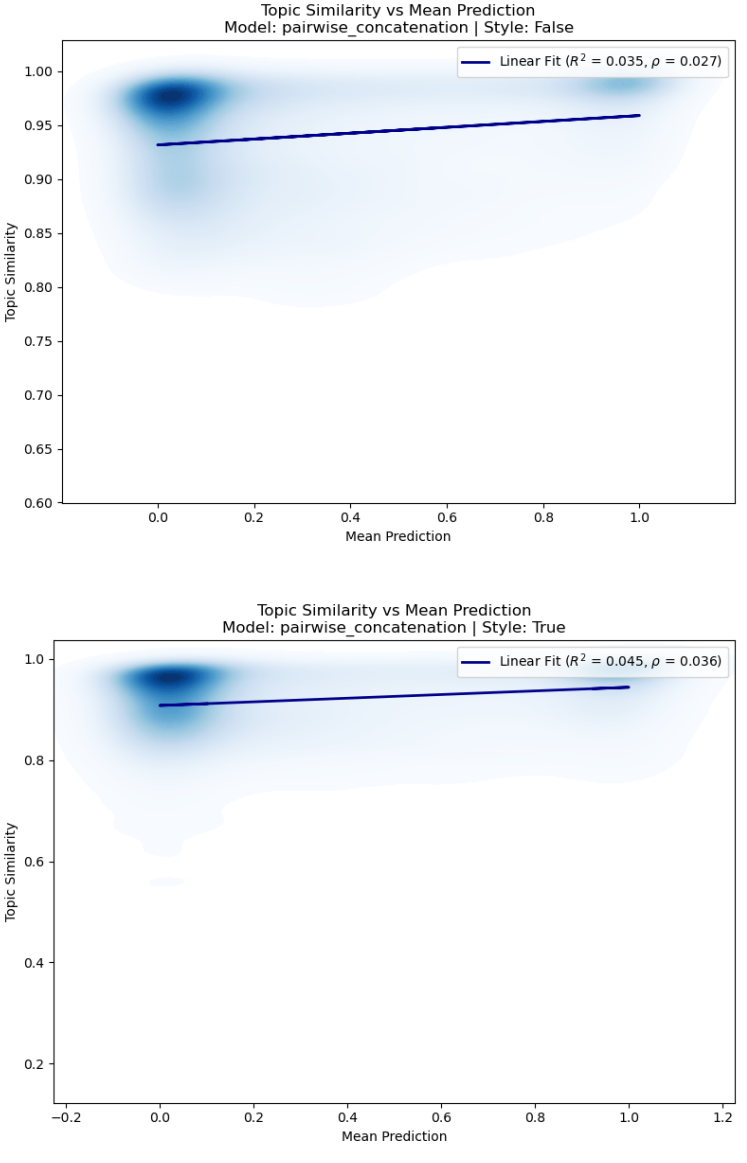


Figure 3.C.3: Pairwise Concatenation model performance with and without style features based on RoBERTa topic similarity. The top figure shows the model without style features, exhibiting the weakest topic correlation. The bottom figure includes style features, where a slight increase in topic dependence is observed.

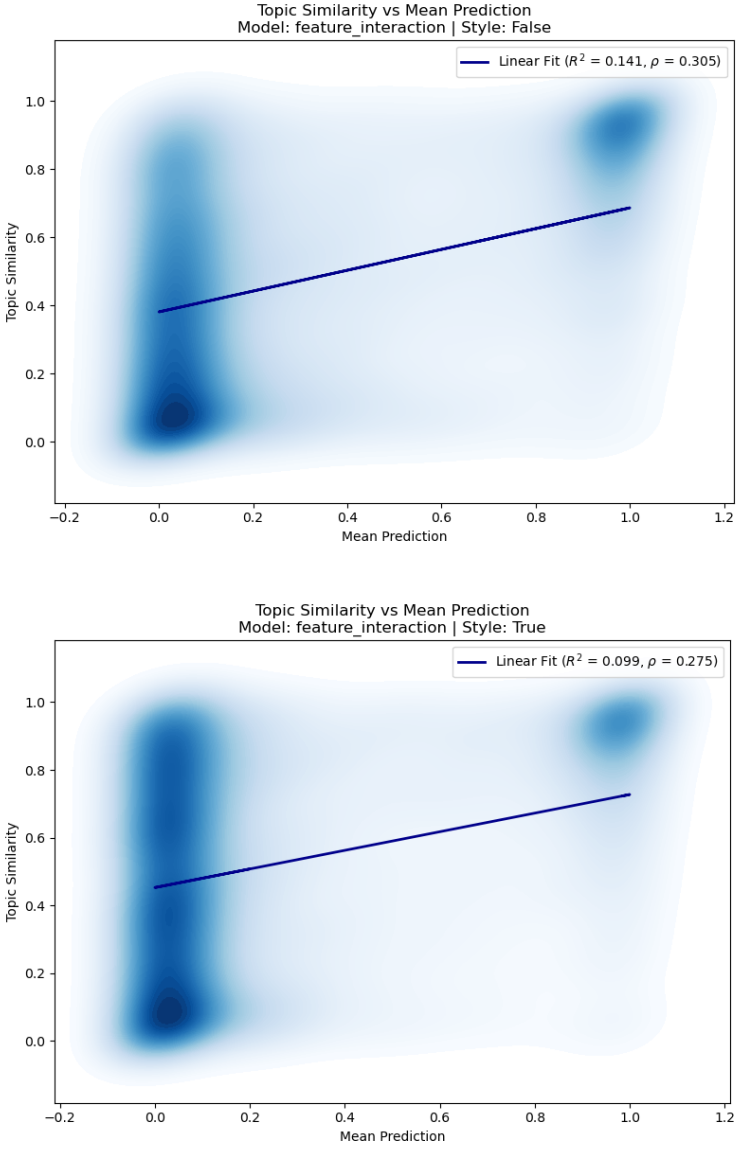


Figure 3.C.4: Feature Interaction model (LDA-based topic similarity): Without style features, the model shows a slightly more spread distribution with a weak positive trend. With style features, there is a slightly increased topic influence compared to the RoBERTa-based similarity setting.

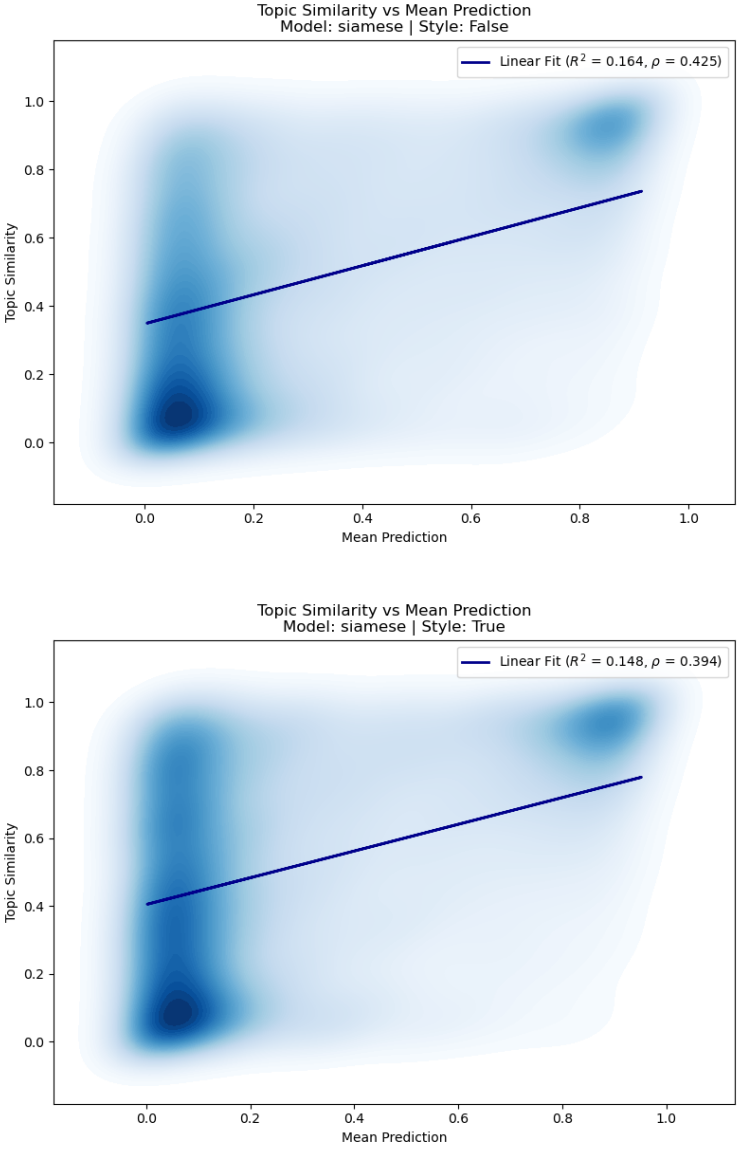


Figure 3.C.5: Siamese model (LDA-based topic similarity): Without style features, the model shows a moderate positive trend. With style features, topic influence becomes weaker than without style and is similar to the RoBERTa-based results.

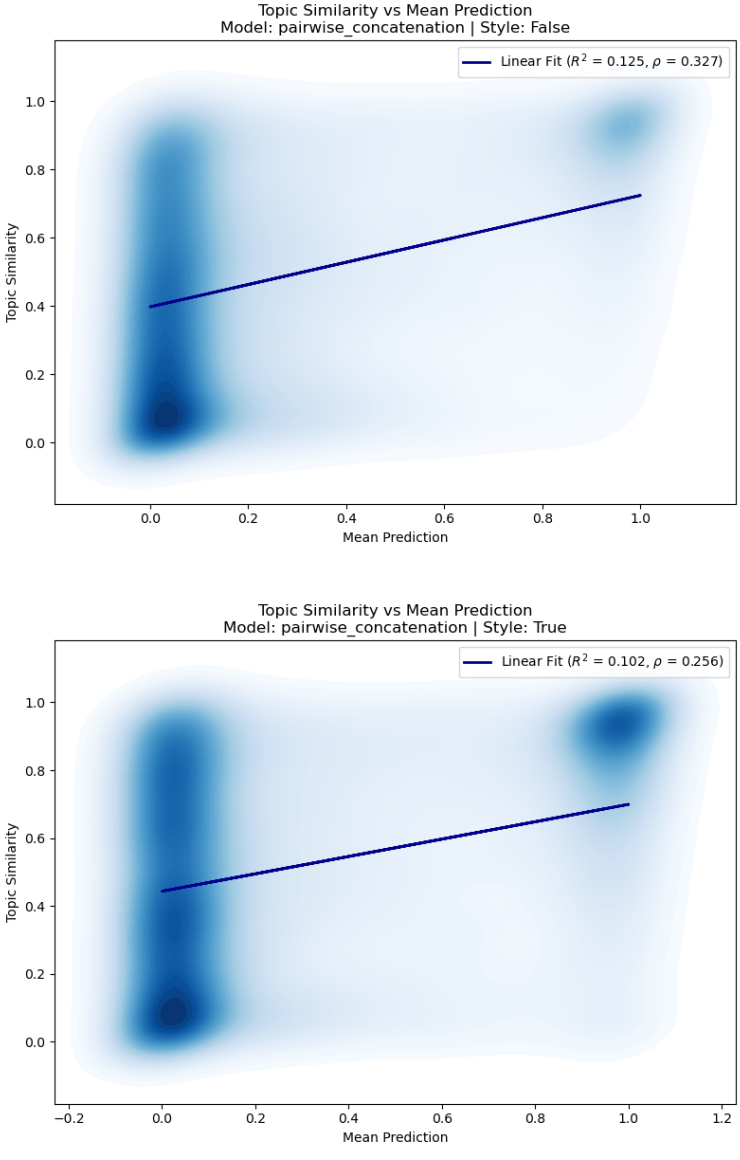


Figure 3.C.6: Pairwise Concatenation model (LDA-based topic similarity): Without style features, the model exhibits weak topic dependence. With style features, there is a slightly increased topic correlation.

3.D Training Curves

To supplement the main evaluation metrics, Figures 3.D.1–3.D.5 display the training and validation performance for all models over the course of training.

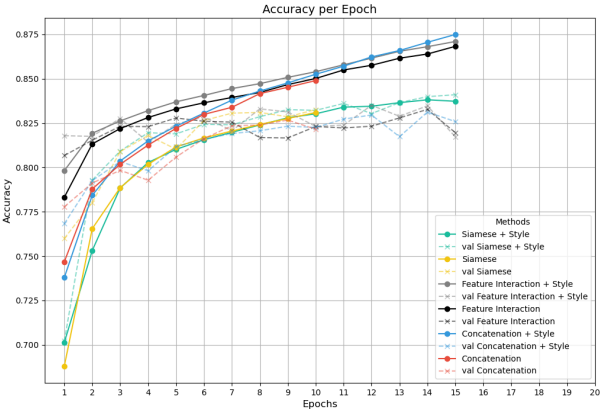


Figure 3.D.1: Accuracy.

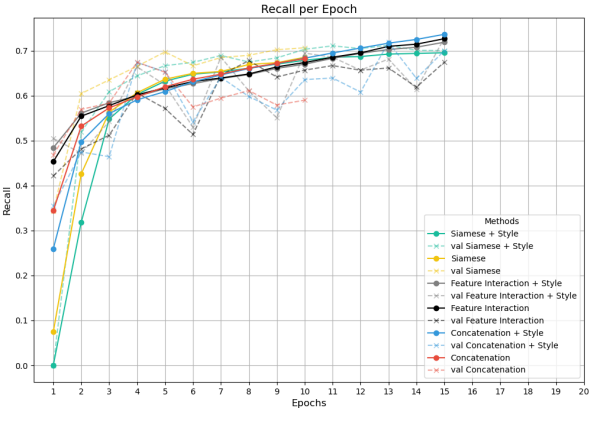


Figure 3.D.2: Recall.

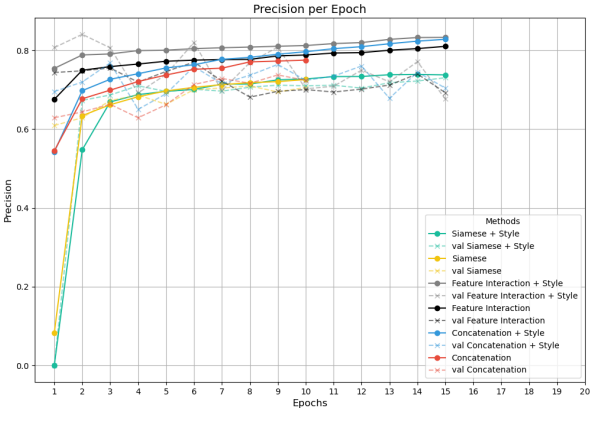


Figure 3.D.3: Precision.

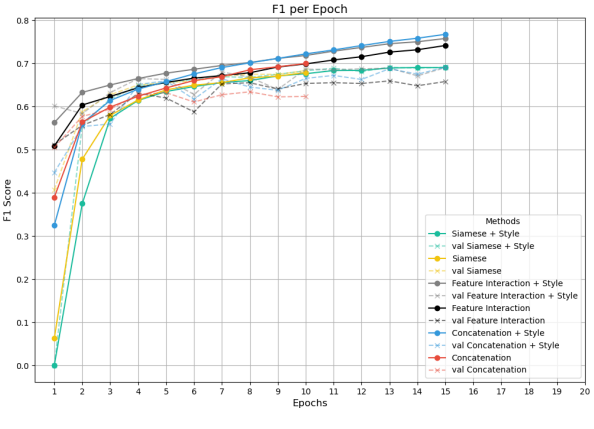


Figure 3.D.4: F1 Score.

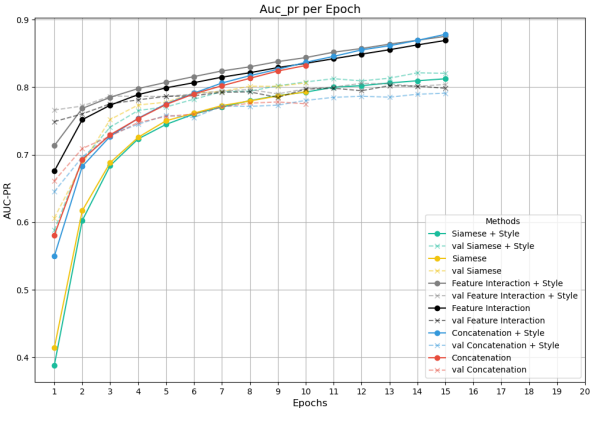


Figure 3.D.5: AUC-PR.

4

Cross-Domain Authorship Verification with Feature Interaction Networks: Evaluating No-Holdout and Holdout Protocols

Contents

Based on:
B. van Leeuwen, S. Bhulai, and R.D. van der Mei.
"Cross-Domain Authorship Verification with Feature Interaction Networks: Evaluating No-Holdout and Holdout Protocols."
Under review.

Abstract

Authorship verification (AV) across communication domains is a challenging task, as models must determine whether two texts are written by the same author despite variations in style, topic, and modality. In this work, we adopt a hybrid approach that combines contextual embeddings from RoBERTa with handcrafted stylometric features to capture both high-level semantic patterns and low-level stylistic cues. We evaluate our method using a cross-domain setup, considering both same-domain and strictly cross-domain pairs, including challenging "hard negatives" where authors write in markedly different styles across domains. To explicitly assess generalization to unseen domains, we further employ a leave-one-domain-out strategy, in which entire domains are excluded during training and used only for testing. Our experiments span multiple domain pairs including emails, text messages, essays, interviews, and speech-to-text data. Results show that the adopted approach consistently outperforms state-of-the-art research on comparable domain pairs, achieving higher overall verification scores while maintaining strong performance across individual metrics. Our findings demonstrate that combining contextual and stylistic representations, together with hard negative sampling, enables robust generalization across heterogeneous text types. Moreover, the leave-one-domain-out evaluation highlights the model's ability to handle previously unseen domains, while analysis of same-domain test pairs shows that these representations also transfer to within-domain verification without explicit training on such pairs. This work provides a comprehensive evaluation of cross-domain AV under realistic conditions and offers insights into building more robust AV systems.

4.1 Introduction

4.1.1 Motivation

AV is the task of determining whether two texts were written by the same author, independent of their content. This problem arises in a variety of real-world scenarios, including plagiarism detection, forensic linguistics, online fraud investigation, and cross-platform identity analysis. In many of these applications, investigators must determine whether texts originating from different sources or platforms can be attributed to the same individual, often without prior knowledge of the author. As digital communication continues to expand across platforms such as emails, social media, forums, and messaging applications, the ability to reliably identify writing style across heterogeneous sources has become increasingly important.

Unlike authorship classification, which assumes a closed set of known authors, AV must generalize to previously unseen authors. This distinction makes AV fundamentally more challenging: instead of learning author-specific classes, models must learn patterns that characterize writing style itself. In addition, texts used for verification may originate from different domains, such as formal essays, informal messages, or spoken transcripts. These domain differences introduce variations in vocabulary, structure, and length, making it difficult for models to distinguish stylistic signals from domain-related characteristics.

4.1.2 Related Work

Stylometry and Feature-Based Authorship Analysis

Stylometry, the quantitative study of writing style, forms the foundation for many authorship-analysis tasks, including authorship attribution, verification, and profiling [74, 75]. These tasks aim to capture the unique stylistic fingerprints of authors to identify authorship, detect textual similarities, or infer author characteristics. Traditional approaches represent style using lexical, syntactic, semantic, structural, and domain-specific features. Early work relied on controlled corpora with large training sets and a limited number of candidate authors, achieving good performance under these constrained conditions [74, 76, 77, 78].

With the proliferation of online texts, authorship analysis has shifted toward more realistic settings involving noisy, heterogeneous, and unbalanced datasets [79]. Machine learning and natural language processing techniques—including probabilistic models, distance-based methods, and compression approaches—have been increasingly applied to capture stylistic patterns in such complex scenarios [74,

77]. However, performance is strongly affected by dataset size, text length, and the number of candidate authors, making reliable analysis challenging in many practical applications.

Feature extraction remains critical for capturing an author's style, yet there is no consensus on an optimal feature set [74, 77]. Low-level indicators such as word or character frequencies and syntactic statistics can effectively distinguish authors but fail to fully capture subtler stylistic nuances [80]. Moreover, correlations between stylistic features and contextual factors such as genre or domain are not well understood, motivating research into richer representations that generalize across authors, domains, and textual contexts [74, 80].

Neural and Representation-Based Approaches

The advent of pretrained language models has transformed authorship analysis by enabling the extraction of contextualized embeddings that capture semantic and syntactic nuances [81, 82]. Large-scale evaluation frameworks such as the PAN shared tasks [3] have further standardized experiments [83, 84, 85]. Results from these evaluations show that while neural models can achieve good performance, traditional stylometric methods based on n-grams remain competitive, particularly when training data is limited [86, 87].

Several studies use pretrained models for AV by training auxiliary classification tasks and then comparing text embeddings using similarity metrics [54, 81, 88]. These approaches can improve robustness in cross-domain scenarios, but they often depend on sufficient training data and carefully curated normalization corpora [82]. Pairwise learning frameworks, such as contrastive learning, extend these methods by directly optimizing similarity relationships between text pairs and incorporating hard-negative sampling to distinguish between stylistically similar authors [87, 89].

Despite these advancements, AV remains challenging under realistic conditions. Cross-domain experiments from PAN show substantial drops in performance when texts originate from different discourse types, such as emails, essays, or text messages [86]. In some scenarios, simple n-gram baselines outperform sophisticated neural models, particularly when training data is sparse [86, 89]. Similar trends are observed in cross-domain attribution, where performance decreases as domain differences increase or candidate sets grow [90].

4.1.3 Research Gap

Taken together, these observations suggest that current methods, while powerful, struggle to capture stable stylistic signals across domains and discourse types.

Traditional stylometric features may overlook subtle patterns, whereas neural embeddings often require abundant domain-specific data to generalize.

Pairwise learning and hard-negative sampling have shown promise in related areas such as authorship attribution in closed-set settings [87] and other areas like computer vision [89], yet their integration AV, specifically in cross-domain scenarios, remains underexplored. This gap is partly due to the nature of commonly used benchmarks, such as PAN, where text pairs are predefined and the sampling procedure is not disclosed.

In particular, there is a need for methods that can effectively handle: (i) heterogeneous texts with varying length, style, and communicative purpose, (ii) limited training samples, and (iii) hard-to-distinguish authors whose writing styles are similar. Addressing these challenges motivates the present work.

4.1.4 Proposed Approach

Recent advances in natural language processing enable the extraction of rich contextual embeddings from pre-trained language models such as RoBERTa [91], capturing high-level syntactic and semantic patterns in text. However, embeddings alone may not fully reflect individual writing style. Traditional stylometric features complement these representations by encoding lexical and structural characteristics, and their combination provides a more comprehensive view of authorship [92, 93].

In contrast to the usual PAN settings, our setting allows explicit control over pair construction, enabling the incorporation of hard negatives. This provides a novel opportunity to study their impact on cross-domain AV. To better distinguish between similar authors, we incorporate a hard negative sampling strategy, encouraging the model to focus on subtle stylistic differences [54, 94]. These features are integrated within a feature interaction framework that models pairwise text similarity [93].

To address cross-domain generalization, we adopt a leave-one-domain-out (LODO) setting, where models are evaluated on domains not seen during training. This setup reflects realistic scenarios in which writing styles and text types vary, requiring the model to rely on domain-independent stylistic signals.

Our contributions are as follows:

1. We study AV under a rigorous cross-domain and LODO evaluation protocol, providing a more realistic assessment of model generalization.

2. We introduce a hard negative sampling strategy tailored for cross-domain AV, which improves discrimination between stylistically similar authors.
3. We provide an extensive empirical analysis of feature interaction-based AV models under challenging cross-domain conditions.

4.1.5 Paper Structure

The remainder of this paper is organized as follows. In Section 4.2, we describe the data and present the analysis that underpins our study. Section 4.3 details the methodology used, including the model design and feature extraction processes. The experimental setup, including training procedures, evaluation metrics, and implementation details, is presented in Section 4.4. Results of our experiments are reported in Section 4.5, followed by a discussion of the findings, their implications, limitations, and directions for future work in Section 4.6. Finally, Section 4.7 concludes the paper.

4.2 Data Analysis

4.2.1 Dataset Construction

For the purpose of this research, we use a dataset provided by PAN [95]. They created a novel dataset from a subset of the Aston 100 Idiolects Corpus in English. The reported key statistics from their preprocessed dataset can be found in Table 4.1. While the data is typically structured as pairs, for our project we received the raw data and construct the pairs ourselves. This is imperative for our second experiment. The raw dataset consists of 25,290 texts written by 112 distinct authors across multiple communication domains. On average, each author contributes 225.8 texts (SD = 25.62), with a median of 224 texts per author. The minimum and maximum number of texts per author are 148 and 298, respectively. This relatively narrow distribution ensures that no single author dominates the dataset while still providing sufficient material to construct robust author profiles.

We included Table 4.2 with the key statistics for our dataset splits and pairing. Compared to the PAN 2022 cross-discourse type dataset [95], our dataset exhibits a similar balance between positive and negative pairs, with roughly 50% of pairs being authored by the same individual. However, the distribution across discourse types differs notably. While the PAN dataset emphasizes email–text message and essay–email pairs, our dataset contains a higher proportion of short-form communications such as text messages, and a larger number of rare combinations grouped under “other” pairs. Average text lengths also differ: essays in our dataset are slightly shorter, while text messages are comparable in length. These differ-

ences underscore the challenges posed by heterogeneous domain distributions and variable text lengths, highlighting the importance of our hard negative sampling strategy.

Table 4.1: Key statistics of the dataset for the 2022 cross-discourse type AV task.

Subset	Training	Test
Author match (text pairs)		
positive (same author)	6,132 (50.0%)	5,239 (50.0%)
negative (different author)	6,132 (50.0%)	5,239 (50.0%)
Discourse type pairings (text pairs)		
Email–Text message	7,484 (61.0%)	6,092 (58.1%)
Essay–Email	1,618 (13.2%)	1,454 (13.9%)
Essay–Text message	1,182 (9.6%)	1,128 (10.8%)
Business memo–Email	1,014 (8.3%)	900 (8.6%)
Business memo–Text message	780 (6.4%)	718 (6.9%)
Essay–Business memo	186 (1.5%)	186 (1.8%)
Discourse type text length (avg. characters)		
Essay	11,098	10,117
Email	2,385	2,323
Business memo	1,255	1,042
Text message	611	601

The distribution of texts across subcorpora is highly imbalanced, as shown in Table 4.3. Short-form communication types dominate the dataset: text messages constitute nearly half of all texts, followed by emails and interviews. Long-form or less frequent modalities such as essays, handwritten texts, and speech-to-text transcripts are comparatively scarce.

Text length varies substantially across the corpus and differs by subcorpus. Short-form communications, such as text messages, dominate the dataset, whereas longer forms like essays and speech-to-text transcripts are less frequent. Figure 4.1 illustrates these patterns across subcorpora, highlighting both the skewness and the variation in text length, which can impact model training and evaluation.

In addition to textual content, the dataset contains metadata fields such as *id*, *thread id*, and *sequence*. These fields allow reconstruction of conversational structure and author-level grouping. Such metadata enables the construction of aggregated author profiles and supports cross-platform verification experiments.

Overall, the dataset is well-suited for cross-domain AV. However, several characteristics must be considered carefully: (i) the strong imbalance across subcorpora,

Table 4.2: Key statistics of the dataset splits for non-holdout.

Subset	Training	Test
Author match (text pairs)		
positive (same author)	6,700 (50.0%)	3400 (50.0%)
negative (different author)	6,700 (50.0%)	3400 (50.0%)
Discourse type pairings (text pairs)		
email–text message	5,730 (42.8%)	2,797 (41.1%)
interview–text message	2,931 (21.9%)	1,402 (20.6%)
email–interview	2,156 (16.1%)	1,088 (16.0%)
se query–text message	743 (5.5%)	455 (6.7%)
email–se query	656 (4.9%)	428 (6.3%)
interview–se query	368 (2.7%)	201 (3.0%)
essay–text message	123 (0.9%)	69 (1.0%)
email–essay	101 (0.8%)	53 (0.8%)
speech to text–text message	108 (0.8%)	47 (0.7%)
email–speech to text	77 (0.6%)	22 (0.3%)
handwritten–text message	64 (0.5%)	39 (0.6%)
other pairs	886 (6.6%)	773 (11.4%)
Discourse type text length (avg. characters)		
essay	10,986	10,361
speech to text	1,979	2,405
handwritten	1,741	1,953
interview	464	428
email	285	286
text message	36	37
se query	28	30

(ii) the predominance of short texts, and (iii) the structured thread information. These properties make the task both realistic and challenging.

4.2.2 Pair Generation

To model AV, the corpus is transformed into a pairwise dataset. Each instance consists of two texts and a binary label indicating whether both texts were written by the same author.

Table 4.3: Distribution of texts per subcorpus.

Subcorpus	Count
text message	12,398
email	7,219
interview	3,470
se query	1,760
essay	186
speech to text	129
handwritten	128
se query memo	112

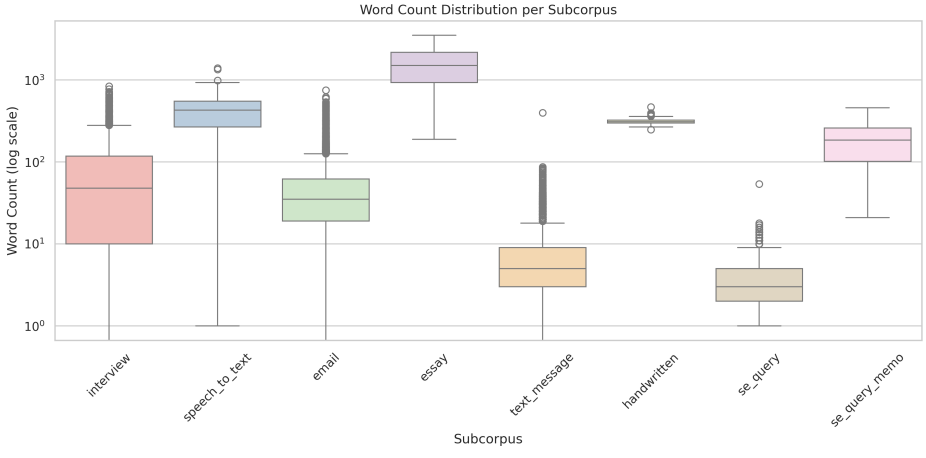


Figure 4.1: Word-count distributions for each subcorpus. Each bar shows the range and frequency of text lengths within the domain.

Positive Pair Construction

Positive pairs are created by sampling two texts written by the same author. To encourage cross-domain generalization, pairs are only formed when the texts originate from different subcorpora. This constraint prevents the model from exploiting domain-specific artifacts and encourages learning stylistic patterns that transfer across communication modalities.

Hard Negative Pair Construction

Negative pairs are formed by pairing texts written by different authors. Rather than sampling negatives randomly, we employ a *hard negative sampling* strategy to provide the model with more informative examples.

For each positive pair, we consider candidate texts from different domains and compute cosine similarity between their embeddings and the anchor text. Instead of selecting the most similar examples, we sample negatives from a mid-similarity range, avoiding both trivially dissimilar and overly similar pairs. This produces semi-hard negatives that are challenging yet diverse.

In addition, a portion of negatives is sampled randomly to maintain variability. All negative pairs are constructed across domains, ensuring that the model cannot rely on domain-specific cues. This combination of semi-hard and random cross-domain negatives encourages the model to learn subtle stylistic distinctions rather than superficial differences.

Each positive pair is paired with one hard negative pair, resulting in a balanced dataset with equal numbers of positive and negative examples. By using these carefully chosen hard negatives, the model develops a more discriminative representation of authorial style, improving its ability to generalize across unseen authors and domains.

The dataset contains a large pool of potential cross-domain combinations. On average, each author can generate tens of thousands of valid positive pairs, ensuring sufficient diversity when constructing training and evaluation splits.

4.3 Model Description

4.3.1 Feature Extraction

Building on our previous research in AV [93], where we demonstrated that a FIN achieves strong performance when combining contextual embeddings with carefully selected stylometric features, we adopt a similar dual-feature representation in this study.

Each text is represented using two complementary feature types: contextual embeddings and stylometric features.

Contextual Embeddings

We extract 768-dimensional contextual embeddings using RoBERTa-base. For each text, the final hidden layer is mean-pooled across tokens, producing a single fixed-length vector representation.

These embeddings capture high-level semantic and syntactic information, enabling the model to recognize stylistic similarity beyond surface-level lexical overlap.

Stylometric Features

To incorporate traditional authorship cues, we compute a set of numeric style features capturing lexical diversity, character statistics, punctuation usage, and readability-related metrics.

Unlike contextual embeddings, these features explicitly model lower-level structural and stylistic patterns. They are precomputed once and stored to avoid repeated computational overhead.

Feature Merging

For each text, the 768-dimensional embedding vector is concatenated with the stylometric feature vector into a unified representation. The resulting pairwise feature matrix contains 1,555 columns, including metadata and features for both texts.

No normalization is applied during feature extraction. All scaling is performed exclusively on the training data within each split to prevent information leakage into validation or test sets.

4.3.2 Preliminary Similarity Analysis

As a sanity check, we computed cosine similarity between embedding representations of paired texts. Positive pairs exhibit a higher mean similarity (0.0197) compared to negative pairs (0.0066), indicating that same-author texts are, on average, more similar in embedding space.

However, the separation remains limited. A simple logistic regression classifier trained on similarity scores achieves an AUC of 0.561, only modestly above chance level (0.5). This suggests that linear similarity alone is insufficient and motivates the use of more expressive neural architectures for cross-domain AV.

4.4 Experimental Design

4.4.1 Experimental Data Splits

We evaluate our approach in two experimental scenarios. The first uses a *no-holdout* setup, in which training, validation, and test splits are strictly author-disjoint but drawn from the same set of domains. The second follows a *holdout* protocol, leaving one domain completely unseen during training to assess cross-domain generalization. These experiments provide insights into different aspects of model generalization: author-level and domain-level.

No-Holdout Setting

The first experiment adopts a protocol similar to the PAN shared tasks. We use solely cross-domain pairs and all communication domains are allowed to appear in training, validation, and testing.

Authors are partitioned into three disjoint sets: training, validation, and testing. Texts written by a given author appear exclusively in one partition, ensuring that test authors are never seen during training.

Pairs are generated independently within each author partition, maintaining both positive (same-author) and negative (different-author) examples. This protocol isolates the challenge of verifying authorship for previously unseen authors, while leveraging information across all available domains. It serves as a baseline that reflects the conventional PAN-style experimental design.

Domain Holdout

The second experiment introduces a stricter LODO protocol to evaluate domain generalization. In each run, one subcorpus is designated as the held-out domain and is completely excluded from the training and validation sets.

Training and validation pairs are constructed exclusively from the remaining domains. Test pairs are then generated such that one text always originates from the held-out domain, while the other text comes from a non-held-out domain. Both positive and negative pairs are included.

This setup assesses the model's ability to generalize to a communication domain that was never observed during training, capturing the added challenge of a more realistic cross-domain AV.

Together, these two protocols provide complementary perspectives: the no-holdout experiment evaluates generalization to unseen authors within familiar domains, while the domain holdout experiment evaluates robustness to unseen domains in addition to author verification.

4.4.2 Feature Preprocessing

Prior to model training, all feature vectors are standardized using statistics computed exclusively from the training data. For each feature dimension, the mean and standard deviation are estimated on the training set and subsequently applied to transform the validation and test sets. This procedure ensures consistent feature scaling across splits while preventing information leakage from evaluation data.

All preprocessing operations are performed offline before model training to minimize runtime overhead. Consequently, the neural network receives as input two normalized feature vectors corresponding to the texts in each pair.

As a preliminary sanity check, a simple logistic regression classifier was trained on the difference between paired feature vectors. Although this baseline achieved performance slightly above chance, its limited discriminative ability indicates that linear similarity alone is insufficient to capture the complex stylistic relationships required for cross-domain AV.

4.4.3 Neural Architecture

AV is formulated as a binary classification task over pairs of feature vectors. Let x_a and x_b denote the feature representations of two texts.

To explicitly model relationships between the two representations, the network constructs additional interaction features:

- Element-wise absolute difference: $|x_a - x_b|$.
- Element-wise product: $x_a \odot x_b$.

These interaction features are concatenated with the original vectors to produce a joint representation:

$$[x_a, x_b, |x_a - x_b|, x_a \odot x_b].$$

The resulting vector is passed through a feed-forward neural network consisting of three fully connected layers:

- Dense layer with 512 units and ReLU activation
- Dropout layer with rate 0.3
- Dense layer with 256 units and ReLU activation
- Dropout layer with rate 0.3
- Dense layer with 128 units and ReLU activation

The final layer is a single sigmoid unit that outputs the probability that the two texts were written by the same author.

Figure 4.1 provides a visual overview of the complete AV pipeline, illustrating feature extraction, interaction, and classification.

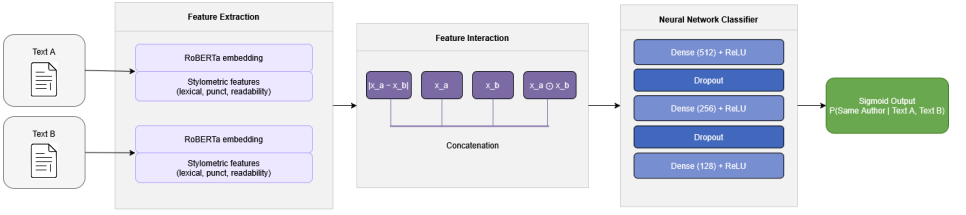


Figure 4.1: Flowchart of the proposed AV method, showing feature extraction, feature interaction, concatenation, neural network classifier, and final prediction.

4

4.4.4 Training Procedure

The model is implemented using TensorFlow/Keras and optimized with the Adam optimizer [30]. Training is performed using mini-batch gradient descent with a batch size of 32.

The initial learning rate is set to 5×10^{-5}. During training, a *ReduceLROnPlateau* scheduler [96] monitors the validation loss and reduces the learning rate by a factor of 0.5 when no improvement is observed for two consecutive epochs. The learning rate is bounded below by 10^{-6}.

Training proceeds for a maximum of 20 epochs. To prevent overfitting, early stopping is applied with a patience of four epochs based on validation loss. When triggered, the model restores the parameters corresponding to the best validation performance.

To account for potential imbalance between positive and negative pairs after sampling, class weights are computed from the training labels and incorporated into the loss function during optimization.

4.4.5 Effect of Hard Negative Sampling

To determine an appropriate balance between random and hard negatives, we evaluate model performance across varying hard negative ratios.

Figure 4.2 illustrates the impact of increasing the proportion of hard negatives on multiple evaluation metrics. As the ratio increases, recall improves, indicating that the model becomes more sensitive to detecting same-author pairs. However, this comes at the cost of reduced precision and overall calibration, leading to an increase in false positives.

Performance across combined metrics, such as F1 and the overall score, peaks at a moderate ratio of approximately 0.4. Beyond this point, performance degrades,

suggesting that excessive reliance on hard negatives reduces the diversity of training signals.

Based on these observations, a hard negative ratio of 0.4 is used in all subsequent experiments.

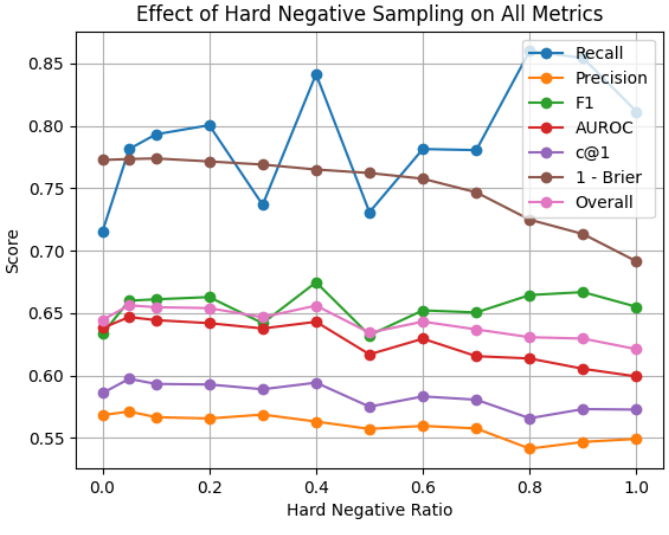


Figure 4.2: Effect of hard negative sampling ratio on evaluation metrics. Moderate ratios yield the best Overall and F1 scores, while higher ratios increase false positives and reduce calibration.

4.4.6 Evaluation

Model performance is evaluated using several standard metrics for binary classification. To facilitate comparison with prior work in AV, results are reported using the PAN evaluation framework, which includes:

- AUROC
- $c@1$ [97]
- $F_{0.5u}$
- Brier score
- Overall score: the average of the scores above.

Performance is reported both overall and per domain pair, enabling analysis of how verification accuracy varies across different cross-domain scenarios.

4.5 Results

4.5.1 No Holdout

Table 4.1 summarizes overall performance across all domain pairs. The cross-domain-only model outperforms the PAN 2022 reference on most reported metrics, including AUROC, c@1, $F_{0.5u}$, Brier, and the Overall score, while the PAN system achieves a slightly higher F1 score.

Table 4.1: Overall AV performance comparing the current cross-domain-only model with the PAN 2022 participant achieving the highest overall score across all domain pairs.

Dataset	AUROC	c@1	F1	$F_{0.5u}$	Brier	Overall
Overall (current model, cross only)	0.640	0.587	0.654	0.596	0.775	0.651
Overall (PAN 2022 best)	0.598	0.571	0.669	0.542	0.749	0.600

Table 4.2 presents results for selected domain pairs. For each pair, the cross-domain-only model generally achieves competitive or higher scores than the best PAN 2022 participant across several metrics. In particular, the Overall score is higher for all three domain pairs shown, indicating improved overall verification performance despite variation in individual metrics.

Table 4.2: AV performance for selected domain pairs, comparing the current cross-domain-only model with the PAN 2022 participant achieving the highest overall score for each pair.

Domain pair	System	AUROC	c@1	F1	$F_{0.5u}$	Brier	Overall
email–text message	Ours (cross only)	0.584	0.565	0.650	0.585	0.763	0.629
	PAN 2022 best	0.613	0.583	0.672	0.581	0.628	0.614
essay–email	Ours (cross only)	0.688	0.604	0.618	0.531	0.770	0.642
	PAN 2022 best	0.531	0.481	0.659	0.531	0.750	0.590
essay–text message	Ours (cross only)	0.556	0.594	0.725	0.649	0.760	0.657
	PAN 2022 best	0.568	0.553	0.673	0.556	0.595	0.599

Table 4.3 presents detailed results for all evaluated cross-domain pairs. Some pairs are omitted because either the corresponding domain was not available in our dataset (e.g., *business memo*) or the number of valid pairs was too small to compute reliable metrics. The bottom three rows report PAN 2022 reference results for comparison, allowing the reader to contextualize our model's performance relative to previously reported benchmarks.

Figure 4.2 visualizes the model's Overall scores across all domain pairs in a symmetric heatmap, where each cell corresponds to a domain pair regardless of order. Higher scores indicate better AV performance. The heatmap highlights substantial variability across domain combinations. While some pairs, particularly those

Table 4.3: Cross-domain AV results for all evaluated domain pairs using only cross-domain pairs. The bottom three rows show PAN 2022 results for domain pairs not present in our dataset (involving business memo), using the best overall scores from the PAN 2022 task to allow comparison with our model's performance on other pairs.

Domain Pair	AUROC	c@1	F1	$F_{0.5u}$	Brier	Overall
se query–text message	0.529	0.835	0.910	0.865	0.854	0.799
handwritten–text message	0.719	0.744	0.853	0.824	0.842	0.796
se query memo–text message	0.481	0.833	0.909	0.899	0.855	0.796
email–speech to text	0.810	0.773	0.667	0.641	0.842	0.746
essay–interview	0.726	0.654	0.690	0.625	0.815	0.702
email–se query memo	0.468	0.643	0.783	0.709	0.763	0.673
essay–text message	0.556	0.594	0.725	0.649	0.760	0.657
speech to text–text message	0.633	0.574	0.667	0.629	0.774	0.656
email–essay	0.688	0.604	0.618	0.531	0.770	0.642
email–text message	0.584	0.565	0.650	0.585	0.763	0.629
email–handwritten	0.516	0.556	0.714	0.610	0.733	0.626
email–interview	0.705	0.633	0.480	0.473	0.808	0.620
email–se query	0.528	0.542	0.673	0.595	0.745	0.617
interview–text message	0.557	0.536	0.599	0.542	0.761	0.599
PAN 2022 results for reference						
business memo–email	0.606	0.586	0.612	0.589	0.633	0.605
business memo–text message	0.525	0.376	0.673	0.455	0.748	0.555
essay–business memo	0.496	0.500	0.635	0.547	0.739	0.584

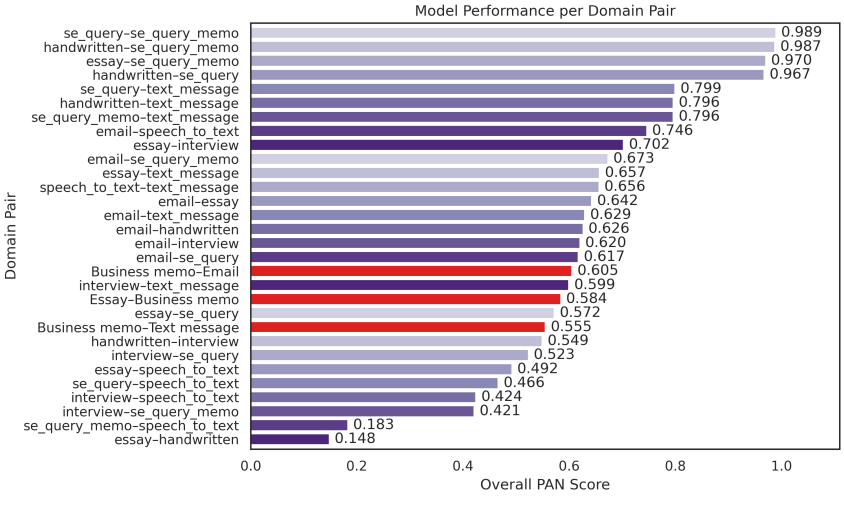


Figure 4.1: Overall scores for all cross-domain pairs in our dataset. Red bars indicate PAN 2022 business memo pairs, which are included for reference but not part of our corpus. Only strictly cross-domain pairs are shown.

involving short-form or conversational domains such as *text message*, achieve relatively strong performance, other combinations yield more moderate results.

Notably, certain domain pairs exhibit exceptionally high scores, such as combinations involving *se_query* and *se_query_memo* or *handwritten* text, indicating that stylistic signals can be highly discriminative in specific cross-domain settings. In contrast, domains such as *interview* and some *speech-to-text* pairings tend to show lower performance, reflecting increased difficulty in capturing consistent stylistic patterns.

Similarly, Figure 4.3 presents AUROC scores for all domain pairs. The observed patterns largely align with the Overall scores, confirming that some domain combinations allow for more reliable discrimination between same- and different-author pairs. However, AUROC values also show considerable variation across domain pairs, indicating that discriminative performance depends strongly on the specific characteristics of the domains involved rather than belonging to a single category of text types.

Together, these heatmaps provide a comprehensive overview of cross-domain model performance, allowing visual comparison across all combinations of domains in the dataset.

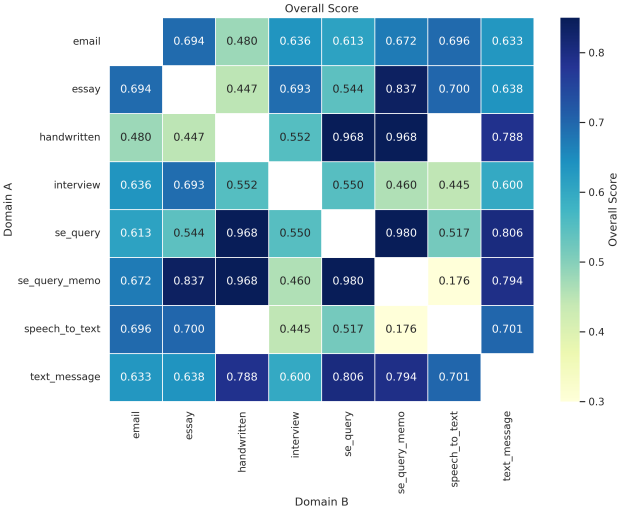


Figure 4.2: Heatmap of Overall scores for all domain pairs. Each cell represents the Overall score of the cross-domain AV model for the corresponding pair of domains. The matrix is symmetric, as the order of domains in the pair does not affect the score.

4.5.2 Holdout

Table 4.4 reports results for the author-disjoint cross-domain evaluation, where each held-out domain is not represented in the training set. Results are shown

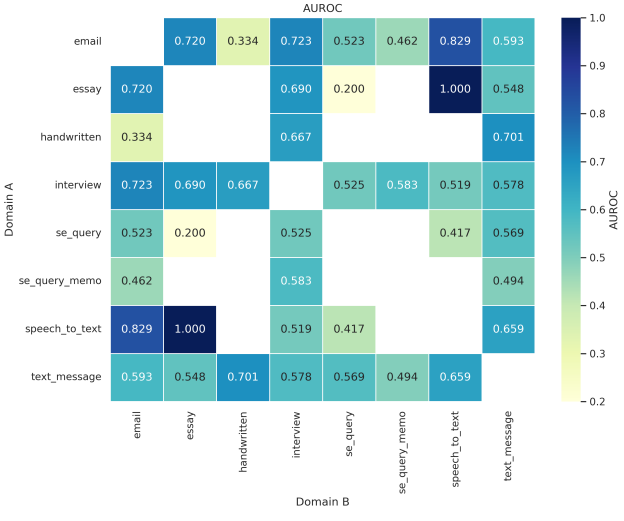


Figure 4.3: Heatmap of AUROC scores for all domain pairs. Each cell represents the AUROC achieved by the model for the corresponding domain pair. The matrix is symmetric, reflecting that domain order does not affect the result.

for two test settings: cross-domain pairs only and cross + same-domain pairs. Metrics include Accuracy, Precision, Recall, F1, Balanced Accuracy, AUROC, Brier, c@1, $F_{0.5u}$, and Overall score.

Performance is reported separately for each domain. Domains such as *Interview*, *Email*, *Text message*, and *SE Query* include results for both evaluation settings, while some domains only contain cross-domain test pairs due to the absence of valid same-domain combinations.

The results demonstrate that the model is able to perform AV on both cross-domain pairs and cross + same-domain pairs, even though same-domain pairs were not included during training. This indicates that the learned representations generalize to both evaluation settings without requiring explicit training on same-domain comparisons.

Table 4.5 reports the percentage of positive samples in each split. While the training and validation sets remain relatively balanced across domains, the test distributions vary more substantially, particularly for domains such as *SE query* and *SE query-memo*. These differences in class balance arise from the way the test sets are sampled for each domain, with some domains containing proportionally more positive pairs than others, which directly affects metrics such as Precision, Recall, and F1.

Table 4.4: AV results for held-out domains, two test settings: cross-domain pairs only ("cross") and cross + same-domain pairs ("cross + same"). Some domains had no same domain pairs available.

Domain	Setup	Acc	Prec	Rec	AUROC	c@1	F1	$F_{0.5u}$	Brier	Overall
Interview	cross	0.543	0.430	0.600	0.550	0.543	0.501	0.456	0.691	0.548
	cross + same	0.552	0.456	0.617	0.567	0.552	0.524	0.481	0.696	0.564
Speech-to-text	cross	0.625	0.524	0.846	0.686	0.625	0.647	0.567	0.771	0.659
Email	cross	0.498	0.454	0.928	0.597	0.498	0.610	0.505	0.642	0.570
	cross + same	0.532	0.496	0.937	0.618	0.532	0.649	0.548	0.665	0.602
Essay	cross	0.611	0.507	0.720	0.667	0.611	0.595	0.539	0.757	0.634
Text message	cross	0.546	0.531	0.683	0.575	0.546	0.597	0.556	0.723	0.599
	cross + same	0.587	0.612	0.724	0.607	0.587	0.663	0.631	0.744	0.647
Handwritten	cross	0.444	0.632	0.240	0.531	0.444	0.348	0.476	0.648	0.489
SE query	cross	0.574	0.563	0.957	0.603	0.574	0.709	0.613	0.711	0.642
	cross + same	0.585	0.575	0.959	0.615	0.585	0.719	0.625	0.718	0.652
SE query memo	cross	0.488	0.643	0.346	0.584	0.488	0.450	0.549	0.713	0.557

Table 4.5: Proportion of positive (same-author) pairs in each domain split. Training and validation distributions are identical for both settings; test distributions are shown for cross-domain only and cross + same-domain pairs.

Domain	Train (%)	Val (%)	Test (cross) (%)	Test (cross + same) (%)
Interview	55.3	56.8	41.8	37.8
Speech-to-text	50.1	50.0	42.0	40.6
Email	55.4	54.1	47.0	46.1
Essay	50.0	50.0	50.6	39.7
Text message	41.9	43.0	53.5	56.1
Handwritten	49.8	50.0	64.9	61.7
SE query	47.2	46.8	64.8	55.4
SE query memo	49.9	49.7	74.4	60.5

Overall, the results highlight the difficulty of AV under strict cross-domain conditions, where the model must generalize to stylistic patterns that were not observed during training.

4.5.3 Error analysis

The error analysis reveals that the model predominantly produces false positives, with 2,281 false positives compared to 533 false negatives. This indicates a bias toward predicting the positive class, suggesting that the model tends to classify text pairs as being written by the same author even when this is incorrect.

Training with hard negative sampling further contributes to this effect. While this strategy improves the model's ability to discriminate between similar authors, it also shifts the decision boundary toward more permissive similarity estimates, increasing the likelihood of false positive predictions in ambiguous cases.

An analysis of errors across cosine similarity bins shows relatively stable error

rates across all similarity ranges, indicating that embedding similarity alone is not strongly predictive of model correctness in this setting. This suggests that the model does not rely solely on geometric similarity in the embedding space when making predictions.

At the author level, error rates are relatively consistent across the most affected authors, with no single author dominating the failure cases. This indicates that errors are largely systematic rather than driven by specific outlier authors.

Overall, the results suggest that the main source of error is an overestimation of similarity between texts from different authors, leading to an increased number of false positives. This behavior is consistent with training under hard negative sampling and highlights a trade-off between discriminative power and calibration in cross-domain AV.

4.6 Discussion

4.6.1 Major Findings

Non-Holdout Experiments

In the non-holdout setting, all discourse types are included in the training data. Evaluation is performed exclusively on cross-domain pairs, aligning directly with the PAN 2022 cross-discourse task and providing the fairest basis for comparison. Accordingly, our discussion focuses on results obtained from strictly cross-domain pairs, as reported in Tables 4.1, 4.2, and 4.3.

The model consistently outperforms the PAN 2022 best-performing system across multiple evaluation metrics. For overall pairs (Table 4.1), the cross-domain configuration achieves an Overall score of 0.654, compared to 0.600 for PAN 2022. These findings indicate that the model effectively captures stylistic signals that generalize across discourse types.

Examining individual domain pairs (Table 4.2), the cross-domain model shows particularly strong performance on the most frequent and linguistically rich pairs. For instance, *essay–email* achieves an Overall score of 0.694, *essay–text message* reaches 0.638, and *email–text message* achieves 0.633. These scores exceed the corresponding PAN 2022 baselines (Overall scores of 0.590, 0.599, and 0.614, respectively), indicating that the model effectively leverages stylistic patterns shared across domains. Improvements are especially pronounced for *essay–email*, where the Overall score increases by 0.104, highlighting the model's ability to generalize between structured long-form and shorter, less formal texts.

It is worth noting that the PAN 2022 dataset includes *business memo* pairings, which are absent in our corpus. While we cannot directly compare on this specific domain, our dataset contains a broader range of other cross-domain pairings, particularly short-form communications such as *text messages* and *SE queries*, which are highly relevant for evaluating cross-domain generalization. This diversity demonstrates that the model's performance gains are not limited to the exact domains present in PAN 2022, but extend to a wider variety of real-world text types.

Taken together, these results show that in a non-holdout setting, the model consistently surpasses PAN 2022 baselines on comparable cross-domain pairs. The improvements are most notable for domain pairs that combine longer, structured texts with shorter or informal communications, suggesting effective learning of cross-domain stylistic cues. Overall, the model demonstrates strong generalization to unseen domain pairings within the training set, underscoring its potential applicability to heterogeneous text corpora beyond the original PAN 2022 data.

Holdout Experiments

In the holdout setting, the training data is drawn exclusively from non-holdout domains, while the test set contains pairs involving held-out domains. Results are reported for two evaluation settings: cross-domain pairs only and cross + same-domain pairs (when available). This design creates a stringent evaluation scenario, where the model must generalize stylistic signals from known domains to entirely unseen ones, representing a realistic cross-domain verification challenge.

Focusing on strictly cross-domain pairs (Table 4.4), the model demonstrates consistent performance across several domains. Structured, long-form texts such as *Essay* and *Speech-to-text* achieve Overall scores of 0.634 and 0.659, respectively. Short-form communication domains, including *Text message* and *SE query*, also yield competitive results with Overall scores of 0.599 and 0.642, indicating that stylistic signals can be captured across both formal and informal text types.

Some domains show different levels of difficulty under cross-domain evaluation. *Handwritten* obtains an Overall score of 0.489, while *SE query memo* reaches 0.557 in the cross-domain setting, reflecting variation in domain characteristics such as writing style and structure.

For several domains, results are also available when same-domain pairs are included in the test set. In these cases, such as *Interview*, *Email*, *Text message*, and *SE query*, the model is evaluated under both cross-domain and cross + same-domain conditions, allowing comparison of performance across both settings within the same domain. This also shows that the model is able to handle same-

domain pairs for domains that were not observed during training, indicating that the learned representations also generalize to same-domain comparisons within unseen domains.

Overall, these results demonstrate strong verification performance in a challenging evaluation scenario, where entire domains are withheld from training. This simulates the practical scenario of encountering completely new text types in real-world applications, such as new communication platforms, emerging text genres, or previously unseen authors. These results suggest that the model can generalize across heterogeneous text corpora, highlighting its applicability to real-world AV tasks.

4.6.2 Limitations

Several limitations should be noted when interpreting these results.

First, some domain pairs contain relatively few samples. This leads to unstable evaluation metrics and occasionally undefined values such as NaN AUROC scores. Small sample sizes may also artificially inflate certain metrics, particularly F1 and $F_{0.5u}$, when predictions happen to align with the limited ground truth labels.

Second, the domain distribution in the dataset is highly imbalanced. For example, *text messages* and *emails* contain significantly more samples than domains such as *speech-to-text*, *handwritten*, or *SE query-memo*. As a result, the model may learn domain-specific patterns for the more frequent domains, which could bias evaluation results.

Third, the error analysis indicates that the model exhibits a bias toward predicting the positive class, resulting in a higher number of false positives compared to false negatives. This behavior is consistent with training using hard negative sampling, which encourages the model to focus on subtle inter-author differences but may also lead to more permissive similarity estimates. As a result, the model may overestimate similarity in ambiguous cross-domain cases, which should be considered when interpreting the reported performance.

The PAN category *business memo* was not present in our data, so we compared our model against all other available domain pairs instead. While this provides a reasonable proxy — and most of our domain pairs outperform the business memo pairs — the stylistic characteristics of these domains may not perfectly align with the original PAN categories.

Finally, the holdout experiment presents additional challenges specific to cross-domain generalization. In this setup, the training set contains only pairs from

non-holdout domains, while the test set includes pairs where one text comes from a holdout domain. Some holdout domains are relatively scarce or highly heterogeneous, which can result in greater metric variability and increased difficulty in capturing reliable stylistic patterns. These factors represent inherent limitations of evaluating AV in a realistic, unseen-domain setting.

4.6.3 Future Work

Several directions for future work could further improve cross-domain AV.

One promising direction is the development of domain-invariant representations. Techniques such as adversarial domain adaptation or contrastive learning could encourage the model to focus on author-specific stylistic signals rather than domain-specific features, potentially improving generalization to unseen domains.

Another avenue for improvement is the incorporation of explicit stylistic features. While modern transformer-based embeddings capture many linguistic patterns, combining them with interpretable stylometric features may improve robustness across domains.

A particularly interesting extension involves systematic holdout experiments to quantify the impact of individual domains in the training set. For example, one could train the model on only a subset of the available domains and evaluate it on a specific holdout domain. This would provide insight into how much each training domain contributes to cross-domain performance and whether the model can effectively generalize to new domains based on limited domain diversity. Such experiments could help optimize training data selection and reveal which domain combinations are most informative for AV.

Future research could also investigate more balanced evaluation protocols, ensuring that each domain pair contains sufficient samples to produce stable performance estimates. Expanding the dataset with additional texts from underrepresented domains may significantly improve the reliability of cross-domain evaluation.

Another direction is refining the test set to better reflect real-world conditions. Instead of balancing positive and negative pairs, the test set could contain a majority of negative pairs, mimicking the natural scarcity of same-author examples. Adjusting pair generation in this way may provide a more realistic evaluation of cross-domain AV performance.

Finally, further work could explore profile-based AV approaches, where models compare new texts against aggregated author profiles rather than individual text pairs. Such methods may better capture consistent stylistic characteristics across multiple domains and writing contexts.

4.7 Conclusion

This study presents a comprehensive approach to cross-domain AV, evaluated across multiple heterogeneous text domains. In non-holdout experiments, where all discourse types are represented in the training data, the proposed model consistently outperforms the best PAN 2022 participant on both overall and per-domain metrics, including AUROC, c@1, F1, $F_{0.5u}$, Brier, and Overall score. Strongest performance is observed for domain pairs combining longer, structured texts with shorter or informal communications, such as *essay–email*, while shorter or less structured domains, including *handwritten* or *SE query-memo*, remain more challenging, reflecting inherent differences in text style and structure.

Holdout experiments further demonstrate the model's robustness in more realistic scenarios, where training data contains only non-holdout domains while the test set contains pairs in which one text originates from a held-out domain.

Even under these stringent conditions, the model achieves strong performance on strictly cross-domain pairs, confirming its ability to capture domain-invariant stylistic signals that generalize to unseen writing contexts. Interestingly, Including same-domain pairs in the test set provides an additional perspective, showing that the model is also able to handle same-domain comparisons for domains not observed during training. Differences between evaluation settings can be attributed in part to variations in class balance and pair distributions, highlighting the influence of sampling on reported metrics. These results emphasize the importance of holdout evaluations for assessing cross-domain generalization beyond benchmarks such as PAN 2022.

Detailed per-domain analyses reveal variability in performance, emphasizing that domain characteristics—such as text length, structure, and style—significantly influence verification outcomes. While some PAN categories, such as business memos, were not present in our dataset, the model demonstrates consistent performance across a broad range of domain pairings. This indicates that the model effectively captures stylistic similarities across diverse text types, beyond the specific domains represented in PAN 2022.

Overall, the findings indicate that the proposed approach provides a robust and generalizable solution for cross-domain AV, effectively modeling stylistic similarities even for previously unseen domains. Future work could include expanding to larger and more diverse datasets, integrating additional stylometric or semantic features, and further exploring domain-specific contributions to generalization, particularly in low-resource or highly heterogeneous settings.

Part III

Kinship Recognition

5

Explainable Kinship: A Broader View on the Importance of Facial Features in Kinship Recognition

Contents

Based on
B. van Leeuwen, A. Gansekoele, J. Pries, E. van de Bijl, and J. Klein, "Explainable Kinship: A Broader View on the Importance of Facial Features in Kinship Recognition," *International Journal On Advances in Life Sciences*, volume 14, numbers 3 and 4, 2022.

Abstract

Kinship Recognition, the ability to distinguish between close genetic kin and non-kin, could be of great help in society and safety matters. Previous studies on *human* kinship recognition found interesting insights when looking for the most relevant features. Results showed that analyzing only the top half of a face gives equal or even better performance compared to analyzing the whole face. In this paper, we aim to find the key features for *automated* kinship recognition based on the theory of *human* kinship recognition; this set of features was researched using features from pre-trained metrics from the StyleGAN2 model. Three different experiments were performed focusing on different aspects of facial features. We found that the most important facial features from the selection of 40 features are mostly focused on the facial hair traits. Furthermore, age-related features were found to play a significant role. This set of features does not entirely comply with the set of features emphasized in *human* kinship recognition. Previous research has shown *human* kinship recognition performance does not decrease when removing the bottom half of the image of the face. In contrast, our results show that for *automated* kinship recognition, removing either the bottom or the top half of a face results in a decrease in the performance of our classifiers. Moreover, only using a selection of facial features corresponding with the important features in human kinship recognition did not prove to be sufficient for the task of Kinship Recognition.

5.1 Introduction

This work is an extension on our previous research in [98] on the importance of facial features in Kinship Recognition.

5.1.1 Kinship Recognition

One of the fields in artificial intelligence that is currently of great interest is computer vision. Computer vision is defined as the study domain that revolves around techniques developed to automate seeing and understanding the contents of digital images such as photographs and videos by computers [99]. This new field started to emerge around the 1960s [100]. However, projects such as getting a computer to describe what it saw via a linked camera, proved much more complex than first thought [101]. Computer vision began to rise after a couple of decades, as the internet advanced and, therefore, access to data improved. At that time (the 80s and 90s), it was facial recognition that grew to be more promising. Subsequently, with the boost of the internet and following later social media, we came as far as Facebook using face recognition every day [102]. One of the subjects that keeps computer vision attractive today is face recognition. From unlocking your phone using your face to automated passport checking at an airport, face recognition is used more often than we realize.

The subject of kinship recognition (KR) is fairly new within face recognition. Kinship recognition is the ability to distinguish between close genetic kin and non-kin. The distinction involves people who are directly related and people who are not. One example of the usage of KR is of families who are spread throughout multiple refugee camps. One of these cases involved a father and his daughter being in one camp, while his wife and other children were in another camp. It took them over a year to get reunited by the Red Cross Restoring Family Links [103]. Even with an organization on these reunion cases, it still takes them a considerable amount of time to solve the problem. If a KR system were able to pick them out as a possible match for a kinship relation, it could be of great help. Upon request, a picture could be taken of a refugee and put in a database. This database could then check for kinship relations with other refugees in the same database. If a missing person is registered in a camp, a kinship relation could be detected instantly this way. They would be reunited much faster and more efficiently. Issues such as communication and limited manpower could also be reduced with the discussed automation.

Another example of the usage of KR is focused on safety measures. Imagine a situation where a terrorist is not in the system. No information can be found on them, only an image of their face is accessible. KR systems could try to match

possible family members that are in the system of known suspects. This could lead to finding the terrorist sooner or to finding their accomplices. Like this, there are more situations where automated KR would be really helpful. With the reality of these problems, KR can bring families together and provide more safety.

The main contribution of this paper is to make a first step towards understanding automated KR and the importance of facial features in it. In the field of KR, there is a lot of room for improvement, especially on the importance of facial features. In our research, we tackled whether kinship is recognizable by using a set of extracted facial features with the use of machine learning. Specifically, we focus on what specific set of features is important for automated KR and if this set of features complies with the set of features important in human KR. First presented in this paper is a literature discussion on human as well as automated KR. We then discuss the data in Section 5.2. In Section 5.3, an overview of the used models is presented. Next, we discuss the results of different experiments in Section 5.4. Lastly, a discussion and conclusion of the presented experiments are given in Sections 5.5 and 5.6.

5.1.2 Related work

Kinship Versus Look-alike

In various researches, which we will discuss later, it has been shown that automated KR is possible to a certain degree. The main question we are left with is how we would separate the classification of two family members from two people that look alike [104], [105]. On one side, two people could be unrelated but their faces as a whole could look alike. If their facial features would be extracted and compared, the features would probably not have high similarity [106]. They have lower inter-class variations. Inter-personal variations refer to the differences in race or genetics. This includes variations such as eye color and the shape of a nose, features that are not possible to be (easily) changed. On the other side, intra-personal variations refer to variations in features that are easily changed, such as hair, facial accessories, cosmetics, pose and illumination [105]. Their face looks similar due to these intra-personal variations. With this information, we expect each feature separately to not show significant similarities. For example, having a close look at the shape of their nose, it is considerably different from the shape of the other person's nose. The intra-class similarity is higher and the inter-class variation is lower for the two individuals of this example [105].

On the other side, there are two people that are related, a daughter and father for example. They do not necessarily look alike since they differ in age and sex. They are in general not seen as lookalikes. However, when looking at their facial

features, most of the time you would see that there is a high similarity for some features, whereas other features would not be similar at all [107]. An example of this is a father and daughter who have a nose and mouth that are very similar. However, their faces as a whole do not look alike, because the rest of their facial features are not similar at all. This would be due to the heredity in kinship, as not all traits inherited from a parent to a child are reflected in the child's appearance.

Human Kinship Recognition

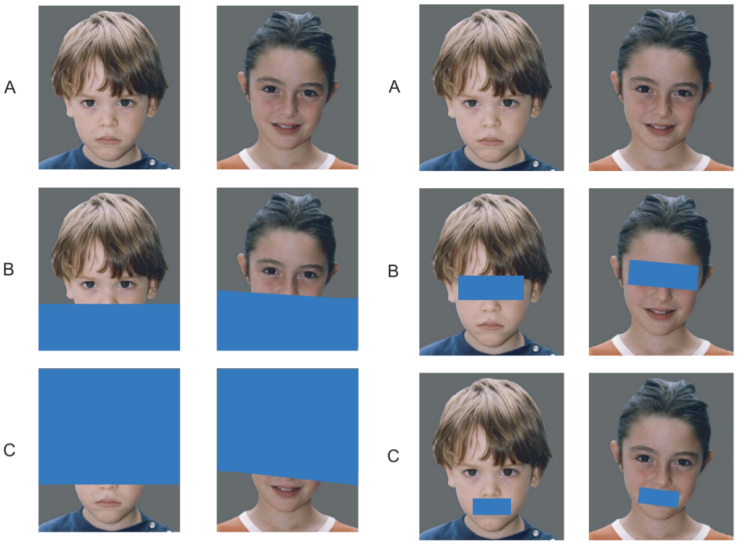
Studies on human KR contribute to our search for the set of important features in automated KR. Several studies [4], [108], [109] have been conducted on human KR, which showed that kinship is indeed recognizable by humans. Robinson et al. [110] used the Families-In-the-Wild (FIW) data set for their human performance measurement. This data set contains images of people's faces that are extracted from family pictures. In total, the data set consists of $656,954$ pairs of images that show a kinship relation. Robinson et al. state that humans scored an overall average of 56.6% accuracy. In this experiment, two pictures were shown and a binary classification was performed between related by kinship and unrelated. Other research on KR [108], [109], [111], [112] shows similar results. The results from Lu et al. [110] are shown in Table 5.1. They performed two different experiments to test human KR. The test group is split up into group A and B. Group A was only shown a cropped face region, whereas group B was shown the whole original color images. Group A intends to test kinship verification purely based on face, while group B intends to test kinship verification based on multiple cues including face, hair, skin color, and background [110]. Their results show that the average accuracy of human KR is higher when face, hair color, and background are taken into account compared to when the focus is purely on the face.

Table 5.1: Results of the two experiments on human kinship recognition focused on four kinship types by [110]. The numbers represent the accuracy and the different columns represent the kinship relations father and son (F-S), father and daughter (F-D), mother and son (M-S), and mother and daughter (M-D).

Method	F-S	F-D	M-S	M-D	Mean
HumanA	61.00	58.00	66.00	70.00	63.75
HumanB	67.00	65.00	75.00	77.00	71.00

We take a look at the Feature Importance (FI) in some of these studies on human kinship detection. The reason behind this specific set of features for human KR might be of help in automated KR. One of the studies is by Martello and Maloney [108], [109], who raised the question which parts of a face are most important for

human KR. In [108], a study is conducted in which humans were tested on their KR skills based on three separate conditions: (1) the right hemi-face masked, (2) the left hemi-face masked, and (3) the face fully visible. Most interestingly, the results showed that there is no significant difference in results for recognizing kinship by humans when the left or right part of the face is covered. On the contrary, a similar study [109] showed that the covering of the top or bottom part of a face does give a significant difference. The effect on kin recognition performance of masks that covered the upper half or the lower half of the face (experiment 1) and the eye region or the mouth region (experiment 2) were measured. An example of the covering up of facial parts for experiments 1 and 2 can be seen in Figures 5.1a and 5.1b below.



(a) Experiment 1: Masking the bottom and top half of the face.

(b) Experiment 2: Masking the eye area and lip area of the face.

Figure 5.1: Illustration of the masking of faces in [109].

In these experiments, it was found that masking the eye region led to a 20% reduction in performance whereas masking the mouth region did not yield a significant difference in performance. This leads us to consider the theory that the performance in KR is dependent on only the upper half of a person's face. Curious is to see how this theory could be used in automated KR. Another discovery is that the eye region contains only slightly more information about kinship than the upper half of the face outside of the eye region. Moreover, the theory is discussed that splitting up face images in different patches can improve the ability of humans to

recognize kinship. This would be caused by the mouth area (i.e., the bottom half of the face) containing lesser kinship features as it is subjected to considerable changes during development [113]. The lower face does not reach its final form until early adulthood [114]. Consequently, this area has fewer stable cues to relatedness. Another view discussed [113] is that environmental effects have little influence on the detection of kinship using facial similarities. This indicates that genetically irrelevant facial information is ignored when human KR is performed.

Overall, the theory that we researched is based on the change in performance when using a specific set of facial features compared to facial features from the whole face. The theory implies that there is a mere necessity of these facial features for KR. These are the features that are located in the upper half of the face, which could lead to only requiring specific parts of faces to identify kinship relations. This could lead to more accessible data since only parts of faces are also sufficient for extracting the important features. Moreover, it could decrease the computational cost of KR models.

Automated Kinship Recognition

For *automated* KR, several approaches have been proposed. Most approaches are not only focused on machine learning models, but also on feature selection. Feature-based methods aim to preserve facial, genetically determined characteristics in the feature descriptors used for the model. These methods identify local facial features such as inconsistencies in an individual's eyes, mouth, nose and skin from the individual's image. Feature-based methods can decrease computational costs and improve model performance. Most of the proposed models and algorithms were only trained on small data sets.

These are data sets like KinFaceW [110], [115] where only four types of kinship relations were given on a handmade data set of around 150 images [116]. Another data set in the field of KR is TSKinFace (Tri-Subjects Kinship Face Database). This data set has been used in some studies [117], [118], but also proved to be too small. These data sets demonstrated to be insufficient for the task at hand. Most of the proposed classifiers are lower-level models and algorithms which use handcrafted feature extraction (features using information presented in the image itself), Support Vector Machines or K-Nearest Neighbor classifier.

Since 2016, a more extensive data set has been constructed in [4]: Families-in-the-Wild (FIW). This data set has been produced to verify kinship and classify relations [119]. The creators of this data set specify promising results in detecting kinship. Robinson et al. [110] state the best results were obtained when using the SphereFace model with an average accuracy of 69.18% and standard deviation

of 3.68. All models performed well compared to previous work, although much improvement could still be made.

After publishing the FIW data set, more research in the field of KR models was done. Many models in KR include the use of FaceNet or other small feature selections for their models' input [120]. FaceNet is a neural network that extracts features of an image. The model provides a mapping from a picture of a face to the Euclidean space. The distances in this space correlate to the amplitude of face resemblance [121]. It produces an output vector to be used as input for a classification model. FaceNet creates embeddings by learning the mapping from images. A disadvantage of using FaceNet is that especially when looking at FI, information gets lost due to lack of feature interpretation [122]. FaceNet could help improve KR models, although we are interested in the similarities between faces by using facial features instead of the faces as a whole. Hence, we use a different approach than FaceNet.

Fang et al. [123] proposed different feature extractions. They performed classification on a pair of images based on the difference between feature vectors of the pair. These pairs are a potential parent and child pair. The top selected features showed to be right eye RGB color, skin gray value, left eye RGB color, nose-to-mouth vertical distance, eye-to-nose horizontal distance and left eye gray value. The results show a high importance for eye related features. 10 out of the 14 top features include the eye area. This study does include specific facial features like eye color, still it only included 22 low-level features. It is indeed shown that most of the selected features are in the upper face area, which complies with the hypothesis.

The top selected features are right eye RGB color, skin gray value, left eye RGB color, nose-to-mouth vertical distance, eye-to-nose horizontal distance and left eye gray value. The results show a high importance for eye related features. 10 out of the 14 top features include the eye area. While this study does include specific facial features like eye color, it only included 22 low-level features. It is indeed shown that most of the selected features are in the upper face area, which complies with the insight.

Most studies on the subject focus on either the overall similarities between faces, or on pre-determined facial feature sets. These studies treat KR tasks similar to the task of a standard facial recognition. Guo et al. [124] argue that kinship classification should be treated differently, since trait similarities are measured across age and gender. Additionally, kinship has a combination of traits and familial traits, which are special for each family pair.

Models proposed by researchers in this field are based on an input of just the

images with little to no alterations. Although, some research focus on specific facial features by using for example a weighted graph embedding-based metric learning framework [125] or by using sparsity to model the genetic visible features of a face [126].

Another group of researchers thought of combining the StyleGAN2 algorithm with KR [127]. In the task at hand, there is a restriction that family members should be recognized on the basis of physical facial features. However, several mentioned attempts neglect this constraint and do not employ any facial landmark before using a classification model. For this reason, Nguyen et al. [127] experimented with KR models using StyleGAN2 as an encoder to incorporate a facial landmark map. This method resulted in an average accuracy of 0.548 for recognizing kinship. Against expectations, no improvement was shown in the results from using StyleGAN2 in this manner, which is presumed to be due to the lack of a proper classification and thus it is argued to need more investigation. An algorithm proposed by Guo et al. [124] use familial traits extraction and kinship measurement based on a stochastic combination of the familial traits. The authors use a similarity score based on a Bayes decision for each pair of facial parts. However, facial features used by the algorithm are limited to the eyes, nose and mouth and, in line with the observations by Guo et al. [124], more parts of the face could be explored. Existing data sets use faces from the same family picture, so models learn about the background similarity. This causes the models to get a higher performance, but when tested on real life pictures, not taken from a family picture, the performance could be lower. When using pre-determined features, this does not present a problem.

5.2 Data

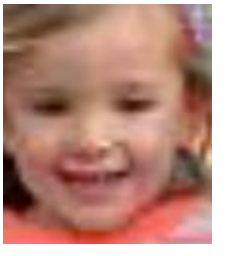
We used the Families in The Wild (FIW) data set. FIW is made up of 11,932 natural family photos of 1,000 families. Other data sets contain less images. KinFaceI consists of 1000 pairs of pictures (so even less unique pictures) [128] and TSKinFace consists of 787 pictures [129]. This makes the FIW data set by far the biggest data set being used for KR.

The data contains images of people's faces that are extracted from family pictures, hence the images vary in quality. All images of the persons are of the same size (108x124 pixels). Some pictures are zoomed in on more than others, which causes this quality difference. In some images, the face and its facial features are clearly shown, but other images are very blurry.

The data is split up into training and test data using hold-out cross-validation. The data is split up in a 70/30 split, respectively. The training set consists of

information on families, persons and relations between persons including images of the persons. The data is distributed as follows. An average of about 12 images per family, each with at least 3 and as many as 38 members. Each family is assigned a unique id, each person is assigned an id and each image collected is assigned a unique id. The data set includes good-quality images of a person's face, but also blurry images of faces, as shown in Figures 5.1a and 5.1b, respectively.

(a) Image from data set.



(b) Blurry image from data set.

Figure 5.1: Example data from the Families-in-the-Wild data set.

A file containing all matches in the training data set is available. However, this does not include data on combinations of persons that do not have a familial relationship. So, these pairs have been constructed by taking random pairs of images from the set of training images of the FIW data set. This is excluding existing related pairs and each pair is unique. This resulted in 205,285 related and 205,285 unrelated pairs of images. The kinship relations are labeled as related. Of all related data points, 21% of the data points are zeroth generation (siblings), 75% are first generation (parents and children) and 4% are second generation (grandparents and grandchildren).

Binary classification is used for predicting relatedness, so the data set is balanced accordingly. Since the focus is on the importance of each of the facial features in recognizing kinship, we have a split in the data between related and unrelated. The distinction between the types of kinship relations is not made. For now, the types are not taken into the classification process, considering the aim is to have a general interpretation of the important facial features. However, the distribution between the types of pairs can help to understand the possible patterns found in feature importance.

StyleGAN2 Metric: Linear Separability

This research is focused on FI in KR. To be able to understand the FI of a model, the features extracted from a model should be interpretable. To collect a bunch of features and to avoid having to do manual annotation, we decide to use a feature

description method from the StyleGAN2 model. With this, it can be easily deducted which of the features of a face are seen as most important by a model for detecting kinship. The pictures in the data are of size 108x124, while the StyleGAN2 description method expects pictures of size 256x256 as input. Interpolation of the pictures in the data is used to overcome this problem. The StyleGAN2 model contains a certain metric called linear separability. StyleGAN2's linear separability metric can be used to steer a generated picture in a certain direction by specifying 40 facial features which are shown in Table 5.2. For example, the models can be used to make the generated face have blond hair and high cheekbones. What we are most interested in for this research are the pre-trained models used in StyleGAN2 which produce probabilities of the 40 features to be true for an image of a person.

Table 5.1: Facial features of linear separability metric.

- 5-o-clock-shadow,
- arched eyebrows,
- attractive,
- bags under eyes,
- bald,
- bangs,
- big lips,
- big nose,
- black hair,
- blond hair,
- blurry,
- brown hair,
- bushy eyebrows,
- chubby,
- double chin,
- eyeglasses,
- goatee,
- gray hair,
- heavy make up,
- high cheekbones,
- male,
- mouth slightly open,
- mustache,
- narrow eyes,
- no beard,
- oval face,
- pale skin,
- pointy nose,
- receding hairline,
- rosy cheeks,
- sideburns,
- smiling,
- straight hair,
- wavy hair,
- wearing earrings,
- wearing hat,
- wearing lipstick,
- wearing necklace,
- wearing necktie,
- young.

The metric was trained using the CelebA Data set (CelebFaces Attributes Data set). This is a face attributes data set with 202,599 celebrity images, each with five landmark locations and 40 attribute annotations. StyleGAN2's linear separability metric is meant to be used for the StyleGAN2 model and its corresponding data. We are interested in using the metric on the data from FIW. The information gathered from the linear separability metric (the facial features) is used as a starting point for the kinship classification models. Transfer learning does not only save time, but it also has the possibility of making a learning process more efficient [130].

Consequently, some adjustments to the data were necessary to apply the metric. This resulted in an output of 40 features for all images in the data set, which then

could be used to train the chosen automated KR models. As data points for the models, we chose a list of length 40 and a list of length 80, composed of the metric values for the features per two pictures. Two input types were experimented with: (1) a list of 80 features, consisting of 40 features per image, and (2) a list of 40 features, taking the absolute difference of the feature values between the images per feature.

5.3 Model Description

We implemented and tested several models to see how well the models work on our data and to find a recurring pattern in FI. For all models, the FI is investigated. The results of this are then used to understand whether the theory of human KR will hold for automated KR as well. Various machine-learning models were selected for this task. For each model, the accomplished accuracy is obtained by K-fold cross-validation. The number of folds is set to 10 and the data is shuffled before splitting into batches.

Machine Learning Methods

Using StyleGAN2's linear separability metric on our data results in an output of 40 features for all images in the data set, which then are used to train the models. As data points for the models, we chose a list of either length 40 or 80, composed of the metric values for the features per two pictures. The five models we decided to experiment with are decision tree, random forest, Gaussian naive Bayes, linear support vector machine and logistic regression.

Decision Tree

First, we have the decision tree algorithm with a maximum depth set to 10, where we obtain the FI by using the Gini importance. The Gini importance is calculated as shown in Equation (5.1) with ni_j the importance of node j, w_j the weighted number of samples reaching node j, C_j the impurity value of node j and $left(j)$ and $right(j)$ the child nodes from left and right split respectively on node j.

$$ni_j = w_j C_j = w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)}. \tag{5.1}$$

The Gini importance value ni_j for feature i is then used for the feature importance of feature i with Equation (5.2) where fi_i is the importance of feature i and ni_j the importance of node j. These values were then normalized to a value between 0 and 1 by dividing by the sum of all feature importance values [131].

$$fi_i = \frac{\sum_{\text{j: node j splits on feature i}} ni_j}{\sum_{\text{k}\in \text{ all nodes}} ni_k}. \tag{5.2}$$

Decision trees are easily interpretable. Because of the use of decision-making logic, the information on the model's features is easily extracted in a comprehensible form [132]. Decision trees have a built-in feature selection, which is beneficial for our research [133]. However, overfitting is common when using decision trees. This is due to the trees being too complex.

Random Forest

Second, we have the random forest consisting of 100 trees, where the FI is obtained by using the impurity importance. The feature importance is computed as an average over all trees. The splitting rules of a random forest scale down the impurity presented by a split. When a split shows a considerable decrease in impurity, the split is seen as important. This theory results in the impurity importance calculation for a variable in the random forest as shown in Equation (5.3) where $RFfi_i$ is the importance of feature i in the random forest, $normfi_{ij}$ the feature importance for feature i in tree j normalized, T_{all} the set of all trees, and T the number of trees in the random forest [131], [134], [135].

$$RFfi_i = \frac{\sum_{j \in T_{all}} normfi_{ij}}{T}. \tag{5.3}$$

Where decision trees are susceptible to overfitting, specifically when a tree is notably deep, the random forest algorithm reduces this possibility of overfitting. This is due to random forest algorithms constructing multiple decision trees where more combinations of conditions are represented [136].

Gaussian Naive Bayes

Then, we have the Gaussian Naive Bayes, which obtains FI by using the permutation importance. The permutation importance is calculated by taking the difference between the prediction error of the baseline metric and the prediction error of the permuted feature metric as shown in Equation (5.4). The permutation importance is obtained as follows. First, the model m is scored s on data D. Then permutation variable importance of feature j is calculated. For each feature, the feature column is randomly shuffled to create an adjusted version of the data $\hat{D}_{k,j}$. The model is then again scored on the data, although now with the feature f replaced by the adjusted version. This results in the score $s_{k,j}$, after which the importance i_j for feature f_j can be calculated with Equation (5.4) [137].

$$i_j = s - \frac{1}{K}\Sigma_{k=1}^{K} s_{k,j}. \tag{5.4}$$

This algorithm generally works very fast and can easily predict the class of a test data set. It is not sensitive to irrelevant features [138]. The naive Bayes algorithm

does perform the best overall when there is independence between the features, while some of our features are dependent. It assumes that all the features are independent [139]. However, even without independence between the features, the naive Bayes algorithm generally performs well.

Linear Support Vector Machine

Next is the linear Support Vector Machine (SVM), where the weights of the model are used to determine FI. These weights are used as vector coordinates. The vector coordinates are orthogonal to the hyperplane represented by the weights. The directions of the vectors represent the class prediction. The difference in the size of the weights is used to determine the feature importance [140]. SVMs have a low risk of overfitting [141], outliers have less influence in the algorithm and the SVM algorithm is relatively memory efficient [142]. Nonetheless, understanding the final SVM model and interpreting the feature importance is difficult. Additionally, SVMs are usually not very suitable for large samples of data, although LSCVs handle this better [143].

Logistic Regression

Lastly, we have logistic regression, where the FI is determined by using the coefficients of the decision function. To get a feeling for the 'influence' of a given parameter in a linear classification model, the magnitude of the coefficient for each feature times the standard deviation of the corresponding parameter in the data is considered. The positive coefficients correspond to outcome 1 (related) and the negative coefficients correspond to outcome 0 (unrelated). This means that a higher positive value of the corresponding feature pushes the classification more towards the negative class [144]. Logistic regression is easy to implement and interpret and very efficient to train. Therefore, it does not require high computation power [145]. The algorithm does make the assumption of linearity between the log odds and the independent variables [146].

5.4 Results

Three different approaches have been researched, the original StyleGAN2 description method, the pre-selected features method and the bottom and top masked method. The results of these approaches are discussed and an overview of the results is provided.

5.4.1 Original StyleGAN2 Descriptor Experiment

The initial approach is taking the results of the StyleGAN2 model and using them as input for the different algorithms. Over all images, we calculated the probabilities of the image complying with the given 40 features. Extracting 40 features per picture resulted in 80 different values since we were working with two images per data point. The FI was determined per model. For the 80 feature input, we took the sum of each feature per picture. An overview of all the results from the StyleGAN2 descriptor experiment can be found in Table 5.4 and Table 5.5.

Decision Tree

The accuracy of the decision tree with 40 features as input has a mean of 0.61 with a standard deviation of 0.003. The 80 features input gives a mean accuracy of 0.66 with a standard deviation of 0.005. The model is more leaning towards giving a positive (related) classification. The feature importance for the decision tree model is shown in Figure 5.1. For the decision tree model with input of 40 features, *arched eyebrows*, *no beard* and *heavy makeup* are the most important features. For the input of 80 features, the top most important features include *young*, *no beard* and *wearing necklace*.

Random Forest

The accuracy of the random forest with 40 features as input has a mean of 0.74 with a standard deviation of 0.003. The 80 features input gives a mean accuracy of 0.80 with a standard deviation of 0.004. The model does not have a clear preference for either a positive or negative classification. With the model giving 51.39% and 50.63% positive classifications for 40 and 80 features respectively, the even distribution of the data in half-positive and half-negative data points is represented well with a slight deviation towards positive classifications. The feature importance for the random forest model is shown in Figure 5.2. For the random forest model with input of 40 features, *arched eyebrows*, *mustache* and *heavy make up* are the most important features. For the input of 80 features, the top most important features include *young*, *no beard* and *mustache*.

Gaussian Naive Bayes

The accuracy of the Gaussian naive Bayes with 40 features as input has a mean of 0.60 with a standard deviation of 0.004. The 80 features input gives a mean accuracy of 0.59 with a standard deviation of 0.005. The model has a preference for positive classification. With the model giving 59.56% and 64.21% positive classifications for 40 and 80 features respectively, most errors are false positives.

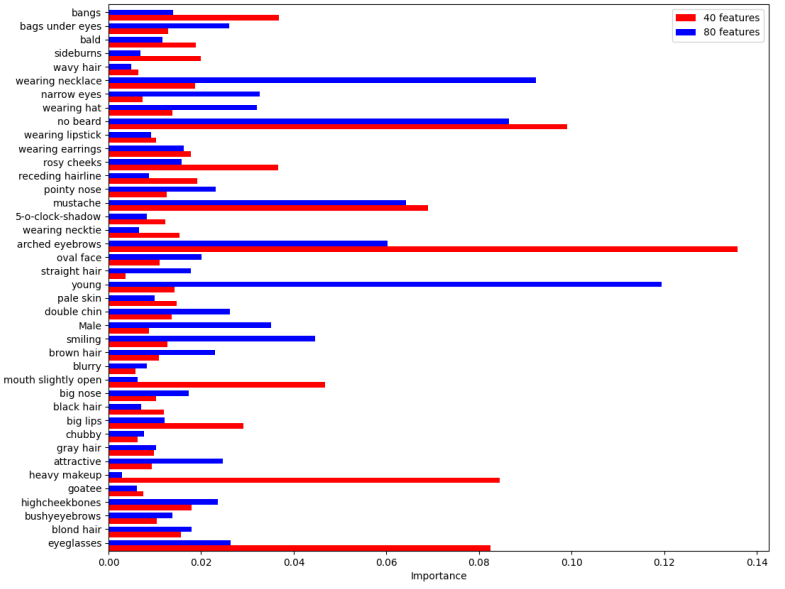


Figure 5.1: Barplots of feature importance for the decision tree model.

The feature importance for the Gaussian naive Bayes model is shown in Figure 5.3. For the Gaussian naive Bayes model with input of 40 features, *eyeglasses*, *mustache* and *arched eyebrows* are the most important features. For the input of 80 features, the top most important features include *eyeglasses*, *rosy cheeks* and *no beard*.

Linear Support Vector Machine

The accuracy of the linear SVM with 40 features as input has a mean of 0.59 with a standard deviation of 0.004. The 80 features input gives a mean accuracy of 0.63 with a standard deviation of 0.005. The model does not have a clear preference for a positive or negative classification. With the model giving 47.08% and 52.92% positive classifications for 40 and 80 features respectively, we see a slight effect of the different input values. The 40 values input gives the model a bit more lenience towards negative classification and the 80 values input gives the model slightly more lenience towards positive classification. The feature importance for the LSVM model is shown in Figure 5.4. For the LSVM model with input of 40 features, *arched eyebrows*, *no beard* and *heavy make up* are the most important

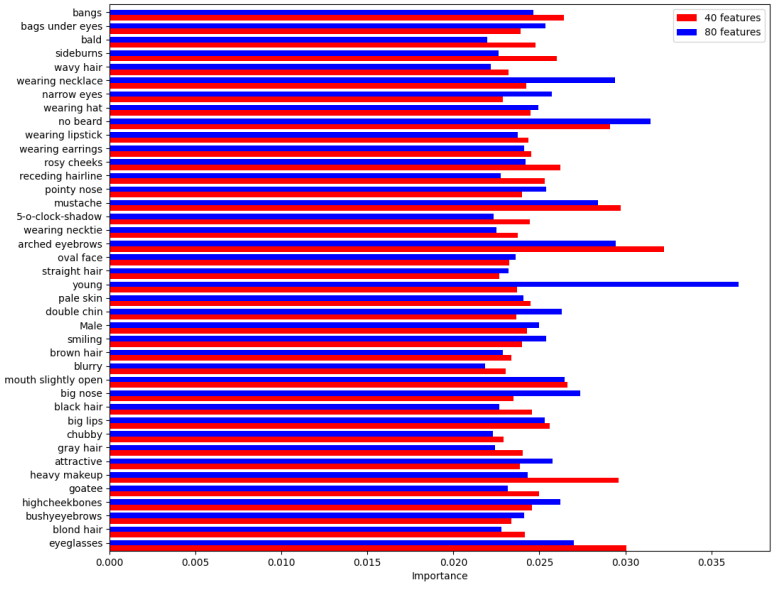


Figure 5.2: Barplots of feature importance for the random forest model.

features. *no beard* and *arched eyebrows* are also among the most important features for the input of 80 features. Here the top most important features include *arched eyebrows*, *narrow eyes* and *no beard*.

Logistic Regression

The accuracy of the logistic regression with 40 features as input has mean 0.60 with a standard deviation of 0.003. The 80 features input gives a mean accuracy of 0.63 with a standard deviation of 0.005. The model does not have a clear preference for a positive or negative classification. The model gives 50.40% and 51.61% positive classifications for 40 and 80 features respectively, which shows the balance of the data with a slight deviation towards positive classification. The feature importance for the Logistic Regression model is shown in Figure 5.5. For the logistic regression model with input of 40 features, *arched eyebrows*, *no beard* and *eyeglasses* are the most important features. These are also among the important features for the input of 80 features. Here the top most important features include *no beard*, *arched eyebrows* and *pale skin*.

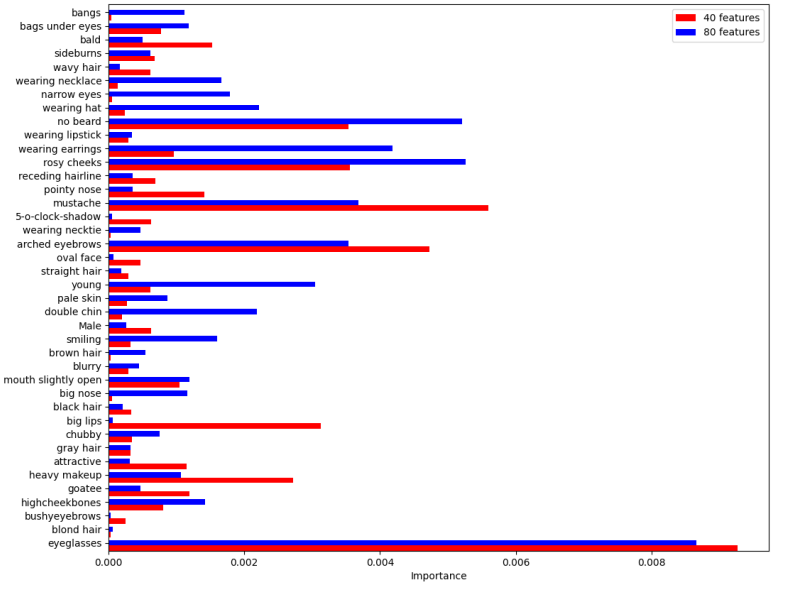


Figure 5.3: Barplots of feature importance for the naive Bayes model.

5.4.2 Selected StyleGAN2 Descriptor Experiment

The second approach is based a certain selection of features. Two different selections were experimented on. The first is focused on the human KR theory. A pre-selection of features is applied to the selected models. The selection of features is focused on the top half of a face. This selection is shown below in Table 5.1.

Table 5.1: Selected set of top half facial features of linear separability metric.

- Wavy hair,
- 5-o-clock-shadow,
- arched eyebrows,
- bags under eyes,
- bald,
- bangs,
- black hair,
- blond hair,
- brown hair,
- bushy eyebrows,
- eyeglasses,
- gray hair,
- high cheekbones,
- narrow eyes,
- receding hairline,
- sideburns,
- straight hair,
- wearing earrings,
- wearing hat.

This approach, despite the supporting theory, did not give better results than using all the features. The expectation was stability in the accuracy scores by only

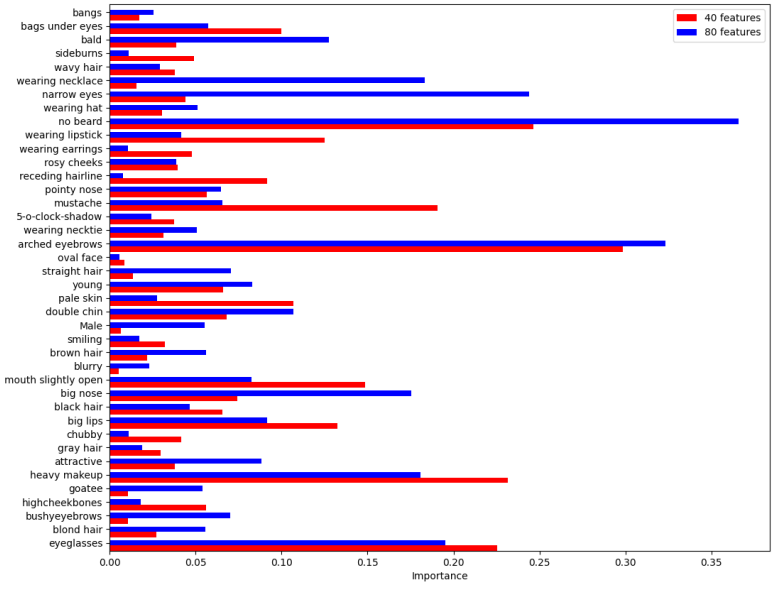


Figure 5.4: Barplots of feature importance for the LSVM model.

focusing on the allegedly most important features. However, most models were less successful and only some of the models continued to perform roughly the same. The accuracy scores for the models using the 19 selected features are given in Table 5.3. For this experiment, the differences were taken between the features, so the results should be compared to the results of the method where the input was 40 features as shown in Table 5.3.

The second approach is based on the selection of features found to be most important according to the original StyleGAN2 approach. The top most important features for this approach can be found in Table 5.2. When using only these features as input, the results are as shown in Table 5.3.

Table 5.2: Set of most important found facial features of linear separability metric.

1. Arched eyebrows,
2. eyeglasses,
3. heavy makeup,
4. mustache,
5. narrow eyes,
6. no beard,
7. young.

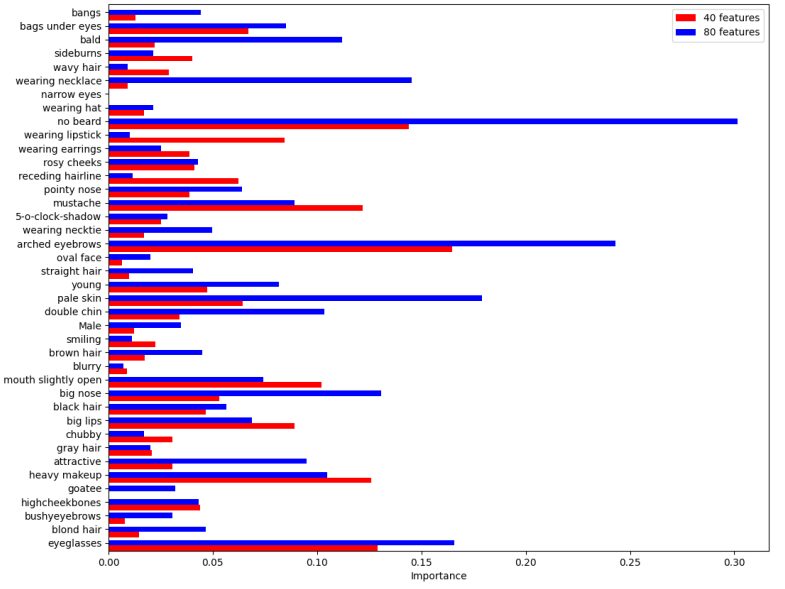


Figure 5.5: Barplots of feature importance for the logistic regression model.

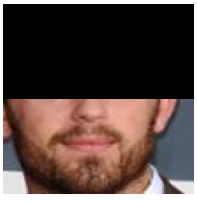
Table 5.3: Overview of results, presented as accuracy, for the two sets of selected features compared to the results of all features.

	Selection Top	**Selection Important**	**All 40**
Decision Tree	0.61	0.61	0.59
Gaussian Naive Bayes	0.57	0.59	0.61
Support Vector Machine	0.57	0.57	0.60
Logistic Regression	0.57	0.57	0.60
Random Forest	0.71	0.62	0.74

5.4.3 Masked StyleGAN2 Descriptor Experiment

To support the theory we found, all of StyleGAN2's linear separability features were taken of not the original image, but over an image with the bottom part of the face masked black like shown in Figure 5.6. The same was done with the top part of the face masked black, comparable to the experiments performed by Martello et al. [108], [109]. All the models are exactly the same as for the original StyleGAN2 description method. Only the input changed.

(a)



(b)

Figure 5.6: Example data from the Families in the Wild data set with bottom masked (a) and top masked (b).

Bottom Half Masked

This experiment was done with all models previously used in the original StyleGAN2 descriptor experiment. The accuracy and FI were obtained for the decision tree, random forest, Gaussian naive Bayes, LSVM and logistic regression models. An overview of the accuracy and important features for all the models from the bottom masked StyleGAN2 descriptor experiment can be found in Table 5.4 and Table 5.5. Again, the results show that the 80 value input gives an overall better performance than the 40 value input and the best-performing model is the random forest for both inputs. Some of the most important features for the bottom masked approach are related to the nose (*pointy nose* and *big nose*) and the hair (*grey hair*, *blond hair* and *waivy hair*).

Top Half Masked

This experiment was done with all models previously used in the original StyleGAN2 descriptor experiment. The accuracy and FI were obtained for the decision tree, random forest, Gaussian naive Bayes, SVM and logistic regression models. An overview of the accuracy and important features for all the models from the bottom masked StyleGAN2 descriptor experiment can be found in Table 5.4 and Table 5.5. Again, the results show the 80 value input gives overall better performance than the 40 value input and the best performing model is the random forest for both inputs.

5.5 Discussion

Multiple models have been tested on FI. Some approaches were based on the human KR experiments from [108], [109]. These experiments showed a certain area of the face to contain the important facial traits needed for KR. We researched the set of features that is most important for automated KR. Pre-trained metrics

Table 5.4: Accuracy for the 40 and 80 value input per experiment: complete, bottom masked and top masked.

Model	Input	Complete	Bottom	Top
Decision Tree	40	0.61 ± 0.003	0.57 ± 0.004	0.57 ± 0.003
	80	0.66 ± 0.005	0.64 ± 0.004	0.65 ± 0.003
Random Forest	40	0.74 ± 0.003	0.62 ± 0.003	0.63 ± 0.002
	80	$\mathbf{0.83 \pm 0.004}$	0.81 ± 0.001	0.82 ± 0.001
Gaussian Naive Bayes	40	0.60 ± 0.004	0.53 ± 0.003	0.55 ± 0.002
	80	0.59 ± 0.005	0.55 ± 0.003	0.57 ± 0.002
Support Vector Machine	40	0.59 ± 0.004	0.55 ± 0.002	0.57 ± 0.002
	80	0.63 ± 0.005	0.60 ± 0.002	0.61 ± 0.002
Logistic Regression	40	0.60 ± 0.003	0.55 ± 0.003	0.57 ± 0.002
	80	0.63 ± 0.005	0.59 ± 0.003	0.61 ± 0.002

Table 5.5: Most important features per experiment.

	Complete	Bottom Masked	Top Masked
Decision Tree	young, no beard, arched eyebrows, eyeglasses	attractive, blond hair, pointy nose, grey hair	young, no beard, arched eyebrows, eyeglasses
Gaussian Naive Bayes	eyeglasses, no beard, young, arched eyebrows	wavy hair, blond hair, pale skin, heavy makeup	eyeglasses, no beard, young, arched eyebrows
Support Vector Machine	young, no beard, pointy nose, arched eyebrows	grey hair, pale skin, wavy hair, big nose	young, no beard, pointy nose, arched eyebrows
Logistic Regression	blurry, no beard, wearing necklace, pointy nose	wavy hair, young, grey hair, big nose	blurry, no beard, wearing necklace, pointy nose
Random Forest	young, no beard, mustache, arched eyebrows	pointy nose, grey hair, smiling, attractive	young, no beard, mustache, arched eyebrows

from the StyleGAN2 model that are meant to be used for synthesizing artificial examples of faces were used. The pre-trained models give 40 values for specific facial features. These 40 values can also be taken from pictures using the pre-trained models. These values were used as input for our machine learning models:

decision tree, random forest, Gaussian naive Bayes, support vector machine and logistic regression. These models were trained and evaluated to show which of the features were seen as most important by the models. More experiments were conducted with the top and bottom parts of a face masked black to also test the theory of human KR.

Major Findings

Interesting results were found when comparing the different models using the original StyleGAN2 description method. Four out of five models had a higher accuracy score when all features for both pictures were kept separate. The models are able to learn about combinations of different features between the two pictures, which has a positive influence on the accuracy score of the models.

The best-performing model seems to be the random forest. Since this model has a very high accuracy compared to the other models, we are specifically interested in its corresponding FI scores. Accordingly, we mainly focus on the results of the random forest model. This model gives high importance values to the features *young*, *no beard*, *mustache* and *arched eyebrows*. It is also noticeable that in two of the five models, the feature *young* is found to be very important and in the other three models, the FI increases when using 80 features instead of 40 features as input. On top of that, in all models, the features *arched eyebrows* and *no beard* are in the top four of the most important features for the model. There is a clear pattern in the importance of facial hair. Beards, mustaches and arched eyebrows are found to be important features for most of the models. Another pattern is the age difference. This gives us reason to believe that the combination of facial hair and the age of a person is strongly correlated to the classification. While the correlation scores do not show a correlation between the two features, the combination of the features does matter when comparing two pictures. A reason for these features to be found important is that most of the kinship relations (75%) in the data set are zero-generation and first-generation relations. Young people are not able to grow facial hair, if they have the genes, it comes with age. This would explain why both facial hair and age are found to be more important.

The set of features that were found to be the most important in our research does not comply with the selection of features proposed by Fang et al. [123]. The set of features used in their research is different, although it is clear that the eye area was found to be the most important by them. Contrasting, the set of important features we found is not particularly focused on the eye area.

For the masked experiment, all five models had a higher accuracy score when all features for both pictures were kept separate. When looking at the bottom

masked method results, a clear decrease in the performance is found compared to the original StyleGAN2 description method. Remarkable is that the feature *young* and the features on facial hair are not found in the top features of almost all models. The original StyleGAN2 approach showed these features to be very important. This leads to the believe that the bottom part of a face is essential for extracting the feature *age*. This would also explain why the feature *grey hair* is found to be important in three out of five models. Grey hair is usually a sign of a higher age. When looking at the top masked method results, a decrease in the accuracy is found, although this decrease is not as excessive as with the bottom masked method. Above is mentioned that the feature young is likely to be extracted from mostly the bottom of a face. However, this is not shown in the results of the top masked method. It is curious that the feature *young* is still not found to be one of the most important. Like the original approach, the top masked method shows the feature *arched eyebrows* to be important. Although a pattern is difficult to find in the top masked method results.

For the bottom masked approaches the difference with the original approach is clear. Where humans showed equal or even better performance when masking the top half of a picture, the algorithms showed the opposite effect.

The set of features does not comply with the set that we expected it to comply with. The gathered results do not give any information that would validate the hypothesis that the most important features would be in the upper half of the face, specifically the eye region. On the contrary, the results are more lenient towards age and facial hair traits to be of great importance. As for the approach with the pre-selected StyleGAN2 linear separability features, the results showed us no improvement when focusing on solely the upper half of the face. When considering the results of the pre-selected StyleGAN2 and the masked StyleGAN2 descriptor experiments, rejecting the hypothesis is even more reasonable.

Limitations

The data set might not be very compatible with the StyleGAN2 metrics, which is an uncertainty. However, as of now, there are no other data sets that contain enough images which are of adequate quality. So we have to accept this limitation for now. An issue was also encountered when using the linear separability metric for a different purpose than StyleGAN2. The results for the top masked method showed one very noteworthy important feature, namely the *arched eyebrows* feature. This feature should be focused on the top part of a face. However, it is found to be important when the top part of a face is masked. More features which show unusual behavior are *smiling* and *pointy nose*, since these are found to be important when masking the bottom half of the face. This is one of the problems

that is encountered when combining StyleGAN2 metrics with other models. The models that are trained for the linear separability metric behave differently than intuitively expected. Using the metric in tasks for which it is not initially intended can cause limitations to the models.

Unexpected Findings

A surprising matter is the difference in performance between the top masked and bottom masked StyleGAN2 description method. Masking the bottom half of the face decreased the performance. As masking the top half of the face decreased the performance as well, it still performed better than the bottom masked method. This is against expectations and raises the question of whether the bottom part of a face contains more information than the top part of a face does for KR.

5.6 Conclusion

We researched the set of features that is most important for automated KR. For this, multiple models have been tested on FI. The results showed that the most important facial features from the selection of 40 features are mostly focused on the facial hair traits and age-related features.

One of the issues we ran into is on transfer learning. The question rises whether StyleGAN2 is compatible enough for transfer learning when combined with our data set. It could be more effective to write a new metric that focuses on more solid facial features. Despite that, the StyleGAN2 metrics are the most elaborate method for finding pre-determined facial features. Other models do not include as many facial features or need manual annotation. It would be contributory to find a way to annotate all parts of the face for many more features to train the models on.

In conclusion, this paper is an important first step towards understanding automated KR, but there are many challenges to be faced before it can be used in real-world applications. As it is now, a large set of clear pictures of complete faces are needed for a model to perform decently. Learning more about the most important parts of our face for automated KR is the next step to take to improve the field of KR.

6

Evaluating Pre-trained Models for Facial Kinship Recognition: A Comparative Study of Video and Image-based Approaches

Contents

Based on:
B. van Leeuwen, Y. Hartman, and M. Hoogendoorn.
"Evaluating Pre-trained Models for Facial Kinship Recognition: A Comparative Study of Video and Image-based Approaches."
Under review.

Abstract

This paper investigates the role of pre-trained models in facial kinship recognition (FKR) under limited supervision. Two fine-tuning strategies were examined: KinFusionNet, a Siamese-style classification model, and Family label Aware Contrastive Learning (FLACL), a supervised contrastive learning framework. Both were evaluated across video-based and image-based inputs using the KAN-AV dataset, with experiments designed to assess the impact of data modality, backbone pretraining, and learning strategy on classification performance.

Our results show that image-based approaches consistently outperformed video-based models, extending prior work by quantitatively comparing both modalities on a large-scale dataset. The best performance was achieved by the image-based FLACL network, reaching an overall accuracy of 70.1%, followed closely by KinFusionNet with frame filtering. Interestingly, this finding challenges the assumption that video data inherently provides more useful kinship signals, suggesting instead that the quality and diversity of pretraining may outweigh the potential benefits of temporal information. Video-based models provided only limited improvements from frame filtering and showed inconsistent accuracy across relationship types. The comparison between KinFusionNet and FLACL further highlights how different fine-tuning strategies impact the transferability of pre-trained features.

Analysis suggests that these differences are partly due to pretraining: the ResNet backbone, trained on large-scale identity datasets, transferred more effectively to kinship classification than the MARLIN backbone, which was optimized for video reconstruction. These findings underscore both the potential of contrastive learning for refining kinship representations and the challenges that remain in leveraging temporal information for this task.

6.1 Introduction

Determining kinship between individuals has traditionally relied on DNA testing, which is highly accurate but also costly and often impractical. As an alternative, automatic Facial Kinship Recognition (FKR) focuses on determining familial relationships directly from images or videos. This automated FKR is typically framed as a binary classification problem or a multiclass identification of specific relationship types.

FKR has gained increasing attention due to its potential applications in various contexts. It ranges from everyday use cases like organizing family photo collections to humanitarian efforts, such as family reunification, to criminal investigations. Despite the recent increase in research, developing reliable systems remains challenging. These challenges stem from two main sources: (1) practical limitations in data collection, and (2) fundamental complexities in the nature of kinship recognition itself. The task itself presents difficulties with large intra-class variations (e.g., differences in age, gender, or expression among relatives) and small inter-class variations (unrelated individuals who appear similar). On the data side, real-world facial images and videos often have inconsistent lighting, different camera angles, varying distances, and partial face obstructions.

Recent advances in transfer learning and self-supervised learning have shown remarkable success in visual representation learning [147], often outperforming supervised approaches on benchmark datasets. Yet these methods remain largely unexplored in FKR, particularly in video-based settings with limited data. In this work, we investigate whether pre-trained models can enhance the performance of video-based FKR when fine-tuned under such constraints.

To address this, we explore two strategies. The first strategy, *KinFusionNet*, is a Siamese-style model designed for image-based inputs, where a pair of facial frames (either static images or selected video frames) is processed through a shared encoder. The resulting feature embeddings are fused and passed through a small MLP to predict whether the individuals are related. This lightweight architecture enables direct evaluation of the encoder's ability to capture kinship-relevant features.

The second strategy, FLACL, builds on contrastive learning principles inspired by SimCLR [148]. Rather than training a classifier, FLACL learns to organize the feature space by pulling embeddings of related individuals closer together while pushing unrelated ones apart, using family labels as supervision. Kinship is then determined based on the similarity between embeddings, using a learned threshold.

6.1.1 Related Work

Kinship Recognition

Before the development of automated systems, visual kinship recognition was widely studied across disciplines including genealogy, neuroscience, psychology, and behavioral analysis [149], [150], [151], [152], [153]. Research focused on understanding how humans recognize family relationships through facial features. Martello et al. [153] conducted a comprehensive study to identify which facial regions are most crucial for kinship recognition. They found that masking the mouth region surprisingly led to the highest accuracy. However, these findings are challenged by research on the importance of the mouth region in identifying parent–child relationships [154]. Additionally, it was found that facial similarity patterns vary significantly between genders [149].

Lopez et al. [155] provided further insight with results highlighting that age differences significantly impact recognition accuracy, and that both humans and machines perform best at recognizing same-gender sibling relationships compared to cross-gender or parent–child relationships.

Early automated kinship recognition systems relied exclusively on handcrafted features—attributes such as color, facial geometry, and landmark distances—fed into traditional classifiers like k-Nearest Neighbors (KNN) and Support Vector Machines (SVM) [156], [157], [158]. Later, researchers incorporated texture descriptors and image-processing techniques for performance improvement [159], [160], [161], [162], [163].

Metric learning emerged as a key innovation, learning similarity functions rather than direct labels [162], [164], [165].

A key innovation was the rise of the so-called metric learning models [162], [164], [165]. Rather than directly predicting kinship labels, these models learned a similarity function to distinguish kin pairs from non-kin.

Subsequently, the first video-based FKR approach was proposed using handcrafted spatio-temporal features, demonstrating that short video sequences could improve recognition accuracy [166].

The shift toward deep learning began with Zhang et al. [167], who introduced the first CNN-based model for image-based kinship verification, achieving a 10% improvement over prior methods. Subsequent research on images explored modern architectures, including Siamese CNNs [168], Graph Neural Networks [169], [170], attention-based networks [171], and GANs [172], [173], [174]. Hybrid models combining handcrafted and deep features have also shown promise [175],

[176], [177], [178], [179].

Table 6.1: Overview of model performance on KinFaceW-II dataset.

Year	Model	Approach	Accuracy	Citation
2014	NRML	Metric + shallow	76.5%	[160]
2014	DMML	Metric + shallow	78.25%	[162]
2015	CNN-points	Deep	88.4%	[167]
2019	Attention Net	Deep	92.0%	[171]
2020	Graph Network	Deep	90.6%	[169]
2021	Relational Network	Deep + Metric	88.8%	[180]
2021	AdvKin	Deep + Metric	88.8%	[174]
2022	TXQEDA+WCCN	Deep + Metric	90.3%	[178]

Video-based methods also benefited from deep learning. Boutellaa et al. [181] used the VGG-Face model [182] to improve performance to 90%, far above the 67% accuracy of earlier models [183]. More recently, Kohli et al. introduced a supervised mixed-norm autoencoder architecture for state-of-the-art performance [184].

Multimodal approaches combining audio and visual data have emerged, supported by new datasets such as FIW-MM, KAN-AV, and TALKIN-Family [5], [6], [7].

However, most prior work has either focused on single images or on leveraging full video sequences, often without systematically comparing both modalities. Our work contributes by providing a quantitative cross-modality evaluation on a large-scale dataset, showing how image-based and video-based models differ in practice. We also introduce a frame selection strategy that allows video setups to benefit from temporal input while retaining the robustness of image-based modeling.

Visual Representation Learning

Visual representation learning extracts compact, informative features from images or videos, which are challenging in FKR due to subtle familial similarities [185]. Early approaches relied on handcrafted descriptors (HOG, LBP) [159], [160], while modern systems use deep learning.

Transfer learning addresses limited labeled data by pre-training on large datasets and fine-tuning on smaller, domain-specific datasets [147], [186]. Transfer learning has been found to consistently outperforms training from scratch, with domain similarity being critical [187]. Pre-trained models like VGGFace and CASIA-

WebFace are widely used [188], [189], [190]. Challenges include data scarcity, label shift, and spurious correlations. For example, models trained on FIW may learn background features instead of facial similarities [191].

In this context, our study adds novelty by directly comparing different pretraining paradigms. We evaluate both identity-focused pretraining (ResNet) and video reconstruction pretraining (MARLIN), revealing important differences in how transferable these representations are to kinship recognition.

Self-supervised learning (SSL) reduces reliance on labeled data via pretext tasks and augmentations [147], [192]. Contrastive methods (SimCLR [148], MoCo [193]) and generative methods (masked autoencoders like MARLIN [194]) are widely used.

Contrastive learning aligns augmented views, while masked modeling reconstructs hidden facial regions. SSL has yet to be fully applied to FKR but related methods, such as Siamese networks and autoencoders, exist [168], [195]. Challenges include weak supervision, limited pretext tasks, high computation, and noisy pseudo-labels [147], but SSL holds promise for video-based FKR with limited labeled data.

Our experiments extend this direction by applying supervised contrastive learning to FKR and demonstrating that it can refine representations learned from large-scale pretraining, particularly in image-based setups where temporal cues are absent.

6.2 Data

6.2.1 KAN-AV

For our task, we use the audio-visual KAN-AV dataset [5], which consists of over 26,000 samples from 970 public figures (actors, musicians, politicians, athletes, etc.) collected from unconstrained YouTube videos. The dataset provides annotations for identity, age, and kinship relations among 645 individuals, covering 532 first-degree pairwise relations across seven types (siblings, and parent–child combinations). The distribution of types is shown in Figure 6.1. Most subjects fall between ages 20 and 80, with a wide range of natural variations in lighting, camera angles, and occlusions (see Figure 6.2). Some examples of the data are shown in Figure 6.3.

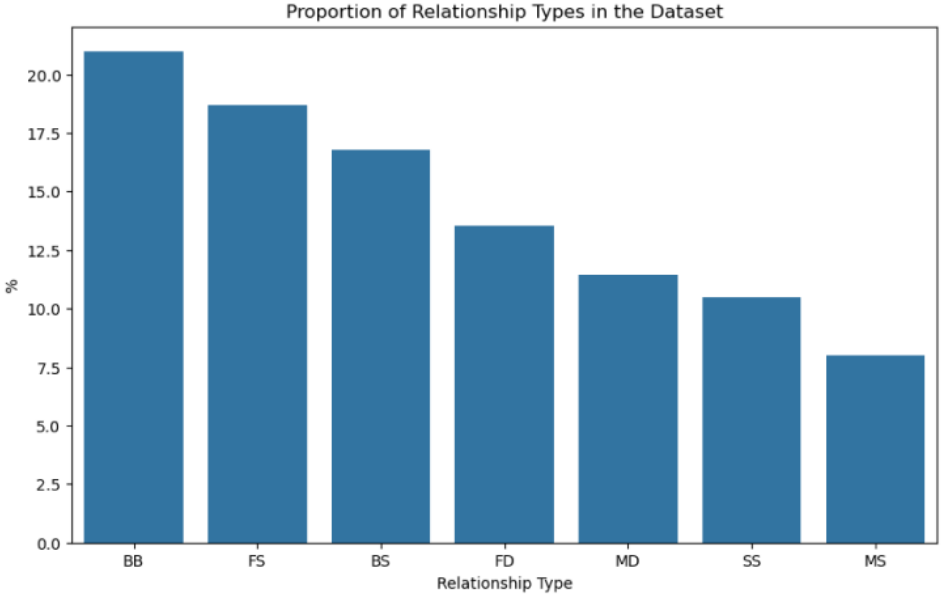


Figure 6.1: Illustration of the distribution of the relationship types in the dataset. The relationship types are: Brother-Brother (B-B), Brother-Sister (B-S), Sister-Sister (S-S), Father-Son (F-S), Father-Daughter (F-D), Mother-Son (M-S), and Mother-Daughter (M-D).

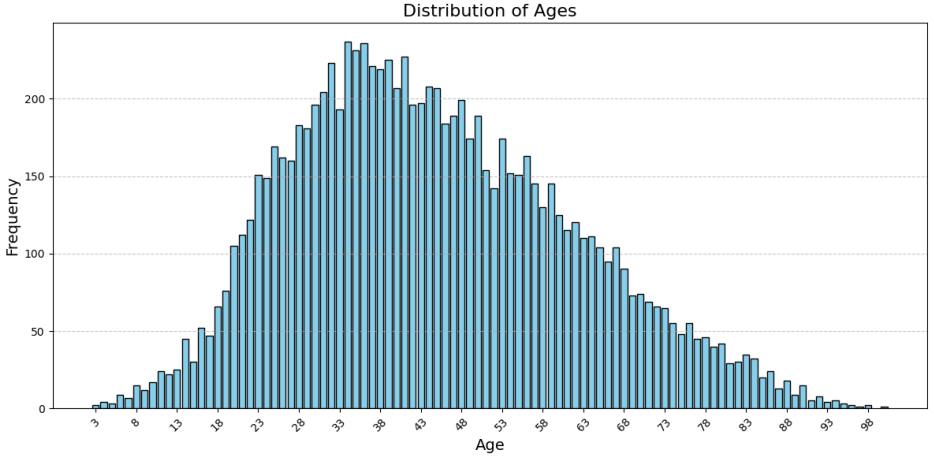


Figure 6.2: Illustration of the distribution of ages in the dataset.

Figure 6.3: A subset of the KAN-AV dataset demonstrating the natural variation in lighting conditions, camera angles, facial occlusions, and subject distance typical of unconstrained environments.

6.2.2 Preprocessing

The preprocessing pipeline follows the standard procedure in kinship recognition, which typically involves detecting the face, cropping to the region of interest, aligning landmarks, and resizing to match the model input [167], [168], [171]. In our case, each individual's face was detected and cropped using bounding boxes provided by the dataset, based on a YOLOv3-based face detector [196]. The bounding box provides the initial region of interest that isolates the face from background noise, a critical step in the kinship recognition pipeline. Figure 6.4 illustrates an example of this detection stage.

Figure 6.4: The detected face and bounding box from the face detection model by Redmon et al. [196].

To maintain informative cues such as hair and ears, margins are added before cropping. Each face is then aligned such that the eyes are horizontally leveled, reducing rotational variation while preserving structural features relevant for kinship recognition. The final extracted region is shown in Figure 6.5

Figure 6.5: The extracted area, including the added margin.

6.2.3 Data splitting

To build a balanced and representative dataset for classification, we followed a structured pairing strategy. The first step involved constructing a family tree to clearly map the relationships between individuals. This was crucial, as a single person can occupy multiple roles within a family, for example, being both a father and a son. As a result, we were able to extract family IDs, which can be utilized during training.

Out of the many possible strategies for creating kinship sample pairs, we followed the approach used by Kefalas et al. [5]. This choice allows for a more direct and fair comparison of results. In this method, the algorithm iterates over all annotated relationships in the dataset. For each relationship type (e.g., father–son, siblings), it randomly selects two video samples that belong to that relationship and links them as a positive pair. Once a sample has been used in a pair, it is excluded from further pairing, ensuring that each video appears in only one pair. This procedure results in non-overlapping pairs and simplifies the construction of clean training, validation, and test splits. While this strategy helps reduce data leakage, it does not eliminate it completely: some individuals may still appear in both training and test sets, though at different ages or in different videos.

For negative pairs, where individuals have no known familial connection, we randomly sampled from unrelated individuals within each split. We ensured that the number of negative samples matched the number of positive ones, resulting in a completely balanced dataset across all subsets.

6.3 Model Description

6.3.1 Problem Description & Formulation

FKR refers to the automated task of determining whether two or more individuals in facial images or videos share a biological relationship. In this research, we consider the classification problem of kinship verification. Following Wu et al. [197], the binary verification task can be formulated as:

$$z = f(g(x_i), g(x_j)) \quad z \in \{0, 1\}, \quad i \neq j,$$

where x_i and x_j denote two face images (or video samples) of individuals i and j, $g(\cdot)$ represents a feature extraction function (e.g., an encoder) that maps an input into an embedding space, and $f(\cdot, \cdot)$ is a classifier that takes two embeddings and predicts whether a kinship relation exists. The output z is a binary variable, indicating whether the individuals have a kinship relation or not. The constraint $i \neq j$ ensures that the two inputs correspond to different individuals, so the task evaluates true kinship verification rather than trivially comparing a person with themselves.

6.3.2 Training Paradigm

Transfer learning has proven highly effective across vision tasks, with models trained on large-scale source domains consistently outperforming those trained from scratch [187]. More than dataset size, how similar the source and target domains are largely determines performance, while multi-source pre-training rarely outperforms strong single-source models. Transfer learning is particularly advantageous for small target datasets, as it facilitates accurate model selection under limited data. For FKR, these insights highlight the importance of selecting backbones pre-trained on facial data that is both representative of the task and sufficiently diverse to enable generalization.

Pre-training Dataset

The effectiveness of transfer learning depends heavily on the dataset on which the backbone was pre-trained, including its scale, diversity, and alignment with the target task. In FKR, where subtle visual cues are essential, both dataset quality and diversity of facial attributes significantly influence transfer effectiveness. In facial recognition, widely used datasets include CASIA-WebFace [189], LFW [190], VGGFace2 [188], and MS1MV2 [198]. These vary in size, identity coverage, and visual conditions as shown in Table 6.1. For our video-based model, we adopt

MARLIN, which is pre-trained on the YouTube Faces (YTF) database [199]. It is a relatively small dataset comprising short video clips of celebrities collected from YouTube. Still, it provides dynamic real-world facial variations suitable for temporal modeling. For the image-based model, we use a ResNet backbone pre-trained on MS1MV1, a large and diverse dataset with variation in pose, age, and illumination, making it well-suited for kinship verification.

Table 6.1: Comparison of common face datasets for training facial recognition models.

Dataset	Number of Identities	Number of Images	Image Source
CASIA-WebFace	10,575	494,414	Google Images
LFW	5,749	13,233	Web Photos
VGGFace2	9,131	3.31M	Google Image Search
MS1MV2	85,000+	5.8M	Web scraping
YouTube Faces (YTF)	1,595	3,425 videos	YouTube

Pretext Task

The training paradigms for the two backbones in this study differ substantially, reflecting the nature of their input data. For video inputs, the MARLIN backbone is trained using a self-supervised generative approach. Large portions of each frame are deliberately masked, and the model learns to reconstruct the missing regions from the visible context. This masked modeling strategy encourages the encoder to capture high-level facial representations that account not only for spatial features but also for temporal dynamics across frames. Importantly, it does so without relying on identity labels, making it particularly suited for modeling facial variability in unconstrained, real-world video.

In contrast, the image-based backbone (ResNet50) follows a supervised paradigm, where the model is trained to classify faces by identity. This approach pushes the network to focus on highly discriminative facial cues that distinguish one individual from another, making it effective for large-scale face recognition tasks. However, the reliance on labeled data introduces potential biases tied to the composition of the training set, which may limit generalization when transferred to tasks like kinship recognition.

Together, these two paradigms illustrate complementary strengths. Self-supervised masked modeling provides robustness to noisy or unlabeled data while capturing broader structural patterns, whereas supervised identity classification yields strong discriminative power. In our study, we retain these paradigms as provided in the original backbone designs, applying the masked-modeling backbone to video-based kinship verification and the identity-pretrained backbone to image-based verification. While this means the two input modalities are paired

with different pretraining strategies, we consider this a feature rather than a limitation: it reflects how these backbones were originally optimized and allows us to examine how their respective strengths transfer to kinship recognition in realistic settings.

Knowledge Transfer

In this research, we adopt an inductive transfer learning approach. The source tasks, face recognition (ResNet50) and face reconstruction (MARLIN), differ from our target task of classifying kinship relationships between individuals. While both tasks operate within the visual domain and rely on facial features, their objectives are not the same. Identity recognition focuses on distinguishing individuals, whereas kinship recognition involves detecting subtle traits shared between related individuals, such as facial proportions, bone structure, or eye spacing.

Within the inductive transfer learning framework, several transfer strategies can be applied depending on what is transferred from the source to the target task [200]. These include instance-based transfer, feature representation transfer, parameter transfer, and relational transfer. In this work, we focus on *feature representation transfer*, where a pre-trained model is used to extract general-purpose features from input data. This paradigm is particularly well-suited for facial kinship recognition, where datasets are relatively small and labels are limited, making it difficult to train models from scratch. By leveraging the rich visual representations learned from large-scale face or video datasets, the model benefits from features that already encode structural and identity-related cues, which can then be specialized to the kinship task with limited fine-tuning.

Fine-tuning

Finally, we explore two fine-tuning strategies. In the first, the backbone remains frozen and only task-specific layers are trained on top. This method allows us to assess the strength of the pre-trained features on their own, without altering the internal representations learned during pretraining. This strategy provides a direct measure of how well the pre-trained features transfer to the kinship task, while also lowering the risk of overfitting.

In the second, we gradually unfreeze the final layers of the encoder and train them using a contrastive learning objective. This setup encourages the model to adjust its representations in a way that better reflects familial similarity, rather than identity. By allowing parts of the backbone to adapt, the model can fine-tune its focus from identity-specific traits to more general patterns shared among related

individuals.

6.3.3 Backbones

MARLIN

For video-based input, we adopt MARLIN (Masked Autoencoder for facial video Representation LearnINg), a self-supervised model extending VideoMAE [201]. MARLIN introduces Facial Region-Guided Tube Masking *Fasking*, which prioritizes semantically important areas (eyes, nose, mouth) identified by a face parser. During pretraining, about 90% of spatiotemporal patches are masked, with the same region removed across all frames to enforce consistent reconstruction.

Unmasked patches are encoded into latent features, which a lightweight decoder uses to reconstruct the original video. This forces the encoder to capture stable facial traits across time—cues known to be critical for kinship perception [149], [150], [153]. For our work, the decoder is discarded and the pre-trained encoder is used purely as a feature extractor (Figure 6.1).

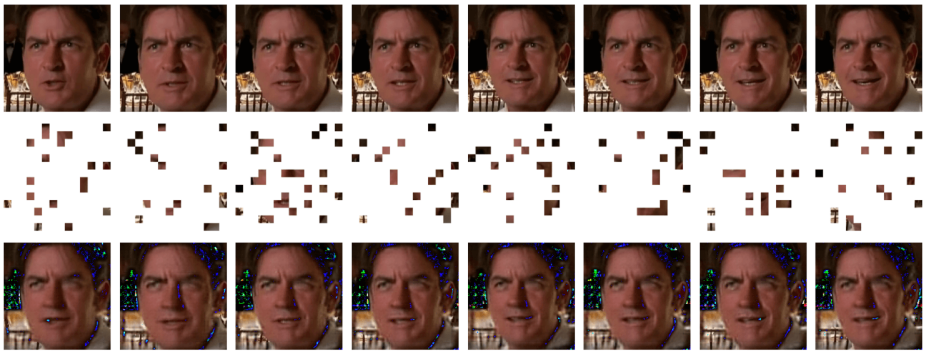


Figure 6.1: Visualization of the MARLIN model's video reconstruction process. The first row shows the ground truth. The second row depicts the input to the model, where 90% of the data is masked, leaving only 10% visible. The final row presents the MARLIN model's reconstruction of the frames.

ResNet

For image-based input, we employ ResNet [202], a widely used architecture introducing residual connections. Instead of directly learning a mapping $\mathcal{H}(x)$, each block learns a residual function

$$\mathcal{F}(x) = \mathcal{H}(x) - x \quad \Rightarrow \quad \mathcal{H}(x) = \mathcal{F}(x) + x,$$

where the shortcut adds the input back to the output (Figure 6.2). This design eases optimization, stabilizes training of deep networks, and mitigates vanishing gradients.

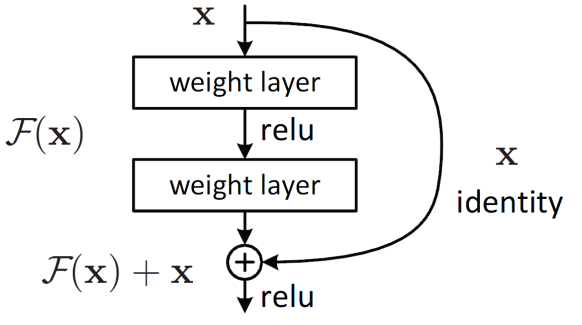


Figure 6.2: Diagram of the residual connection [203].

We use a ResNet backbone pre-trained with AdaFace [204], which improves face recognition by weighting training samples based on image quality. Quality is estimated from the embedding norm $\|f(x)\|$: high-quality samples (larger norms) are trained with a stronger adaptive margin, while low-quality ones receive weaker updates. This reduces the influence of noisy or blurred data and yields more robust features at inference, since the model emphasizes reliable visual cues without overfitting to poor inputs.

6.3.4 Model Design

To study kinship verification, we investigate two complementary model designs: a Siamese-style network we refer to as KinFusionNet, and a contrastive learning approach inspired by SimCLR. Both models operate on pairs of facial samples, yet they differ in how they exploit feature representations and supervision. As described in Section 6.3.3, the input features are extracted from pre-trained backbone encoders.

The exact form of these inputs depends on the experimental setup (Section 6.4). In experiments 1 and 2, each datapoint consists of a set of frames sampled from a video, either unfiltered or filtered for pose and quality. In Experiment 3, by contrast, a single best-quality frame is selected from each video, shifting the task from video-based to image-based kinship recognition. This distinction also determines which backbone is applied: MARLIN for multi-frame video inputs and ResNet-50 for single-frame image inputs.

In the following, we detail how each model builds on these backbone representations to predict kinship relations.

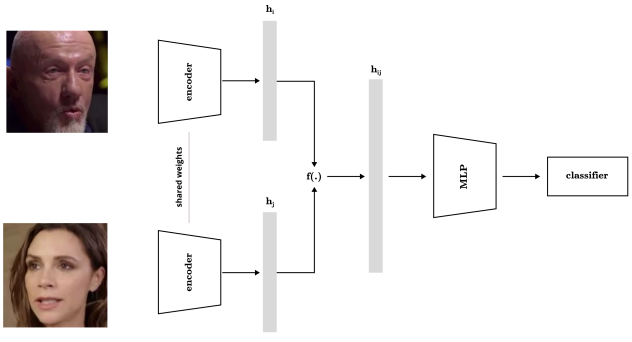


Figure 6.3: Architecture of the Siamese-style KinFusionNet.

KinFusionNet

The first model is a Siamese-style network designed to evaluate how well facial representations from a backbone encoder can determine kinship. Each input pair is processed independently through a shared encoder to obtain embeddings (h_i and h_j), which are then fused into a joint representation $h_{(ij)}$. Inspired by Li et al.[205], we combine multiple fusion operations—addition, subtraction, dot product, and concatenation—into a single composite vector, capturing both shared and contrasting features.

The fused vector is passed to a lightweight MLP. This consists of one or more fully connected layers, each followed by non-linear activations (ReLU), which enable the network to capture complex interactions between the fused features. The final layer applies a sigmoid activation to output a probability between 0 and 1, representing the model's confidence that the pair of individuals are related. By keeping the backbone frozen, we isolate the contribution of these pre-trained representations and evaluate how well they transfer to the kinship verification task. The network is optimized using Binary Cross-Entropy (BCE) loss, which directly penalizes incorrect kin/non-kin predictions while maintaining a simple and interpretable setup.

Family Label Aware Contrastive Learning (FLACL)

The second model refines the encoder using a contrastive learning objective inspired by SimCLR [148]. SimCLR learns feature representations by generating two augmented views of each image as a positive pair, while treating all other images as negatives. Each view passes through a shared encoder to produce embeddings (h_i, h_j), which are projected into a latent space (z_i, z_j) via a small MLP optimized for contrastive loss [206]. After training, the projection head is dis-

carded, and the encoder is used as a feature extractor.

In our adaptation, positive pairs are formed from biologically related individuals using kinship labels, converting the framework into supervised contrastive learning [207]. Kinship is then determined by a similarity metric (e.g., cosine similarity) rather than a classifier. To avoid false negatives from unrelated but semantically similar family members, each batch contains samples from a single family. A custom batch sampler [208] ensures diversity across epochs, promoting generalization to unseen kinship configurations. Figure 6.4 and Figure 6.4 and 6.5 illustrate the architecture and NT-Xent training process.

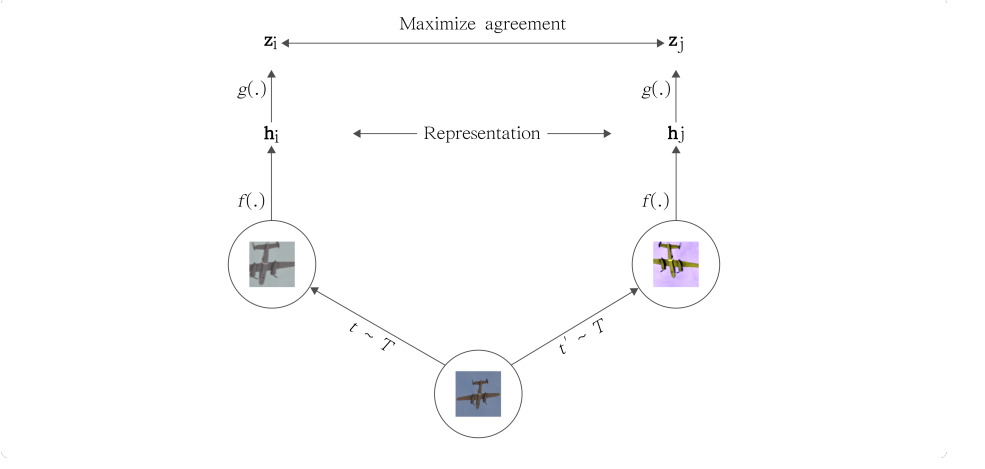


Figure 6.4: A diagram that illustrates the architecture of the original SimCLR Network with data augmentation transformation T. $f(\cdot)$ represents the encoder, whereas $g(\cdot)$ the projection head.

Normalized Temperature-scaled Cross Entropy Loss

To train contrastive learning models such as SimCLR, a specialized loss function is required to bring positive pairs closer in the embedding space while pushing negative pairs apart. We use the Normalized Temperature-scaled Cross Entropy loss (NT-Xent) [209].

The NT-Xent loss maximizes similarity between a positive pair while minimizing similarity between an anchor and all other negative samples. Cosine similarity measures closeness of embeddings, and a softmax normalizes these similarities into a probability distribution. The model is penalized when negatives are more similar than positives, and rewarded when positives stand out, guiding the encoder to learn discriminative features.

Mathematically, for a positive pair (z_i, z_j) in a batch of size $2N$, NT-Xent is defined as:

$$l_{i,j} = -\log \frac{\exp\left(\text{sim}(z_i, z_j)/\tau\right)}{\sum_{k=1}^{2N} \mathbf{1}_{[k \neq i]} \exp\left(\text{sim}(z_i, z_k)/\tau\right)}, \quad \text{sim}(z_i, z_j) = \frac{z_i \cdot z_j}{\|z_i\| \|z_j\|}.$$

Here, τ is the temperature parameter controlling the softness of the distribution, ensuring a smooth similarity scaling across pairs. NT-Xent effectively structures the embedding space, making it well-suited for downstream kinship verification tasks.

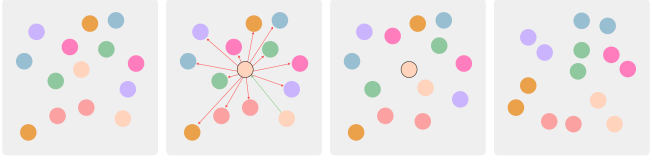


Figure 6.5: Illustration of the contrastive training process using NT-Xent loss. Positive pairs are pulled together, while negatives are pushed apart, resulting in a structured feature space.

Adaptation for Label Aware Learning

The main difference between our approach and the original SimCLR framework is in positive pair construction. SimCLR creates positives via different augmentations of the same image in a self-supervised setting, assuming no access to labels. In contrast, we use known kinship labels, forming positive pairs from images or video frames of biologically related individuals. This effectively converts the framework into supervised contrastive learning [207].

A limitation of SimCLR is false negatives: all other examples are treated as negatives, so embeddings of related individuals can be pushed apart. This is particularly problematic in kinship tasks with multiple family members per batch. SimCLR mitigates this with extremely large batch sizes, which increases computational cost. We address this by ensuring each batch contains samples from only one family, so all other individuals can safely be treated as negatives. This preserves the reliability of the contrastive loss without requiring large batches. A custom batch sampler [208] dynamically constructs batches at each epoch, randomly selecting individuals from different families and avoiding repeated groupings, promoting diversity and better generalization to unseen kinship pairs.

6.4 Experimental Design

In this section, we outline the procedures used to prepare and tune our models, followed by a description of how the data is adjusted across the three experimental conditions.

6.4.1 Model Preparation

Each model requires its own configuration due to differences in backbone architectures, covering both hyperparameters and architectural components. To ensure robust evaluation and reduce overfitting, we employ ten-fold cross-validation. Data splits are partitioned to prevent overlap of video samples or identities between training, validation, and test sets, ensuring clean separation (Section 6.2.3).

For KinFusionNet, each input pair is passed through a shared encoder to obtain feature vectors, which are fused using element-wise absolute difference $|x - y|$, average $\frac{1}{2}(x+y)$, and multiplicative interaction $x \cdot y$. This fusion strategy is motivated by the findings of Li et al. [205], who demonstrated that combining multiple fusion operations yields better performance than any single strategy. Various MLP architectures were then tested, varying depth and width, including the possibility of no hidden layers. Training uses Adam with a learning rate of 4×10^{-3} and batch size 32. Table 6.1 summarizes the 10 architectures evaluated. The final choice of MLP architecture was determined based on preliminary experiments assessing accuracy on validation sets.

Table 6.1: Experiments to test effect of different MLP setups.

Name	Dimension
deep-0-hidden-layers	[] (no hidden layers)
deep-1-hidden-layer	[2 × input-dim]
deep-2-hidden-layers	[2 × input-dim, input-dim]
deep-3-hidden-layers	[2 × input-dim, input-dim, 256]
wide-1-hidden-layer	[3 × input-dim]
wide-2-hidden-layers	[3 × input-dim, 3 × input-dim]
wide-3-hidden-layers	[3 × input-dim, 3 × input-dim, 3 × input-dim]
extra-wide-1-hidden-layer	[5 × input-dim]
extra-wide-2-hidden-layers	[5 × input-dim, 3 × input-dim]
extra-wide-3-hidden-layers	[5 × input-dim, 3 × input-dim, 1 × input-dim]

After selecting the optimal MLP architecture, we tune hyperparameters including epochs, learning rate, weight decay, batch size, optimizer, and momentum (Table 6.2). Accuracy is used as the primary criterion for selection, while precision, recall, and F1 score are reported for completeness and interpretability. These metrics provide additional insight into class balance and model behavior but do not influence hyperparameter selection.

Table 6.2: Search space for hyperparameter tuning KinFusionNet.

Parameter	Options
batch size	[32, 64, 128]
epochs	[10, 20, 30]
learning rate	[0.0001, 0.0005, 0.001, 0.005]
weight decay	[0.0, 0.0001, 0.001]
momentum	[0.95, 0.90, 0.85]
optimizer	[SGD, Adam]

For FLACL, no architectural changes are needed. Hyperparameter tuning includes batch size, epochs, learning rate, weight decay, momentum, learning rate decay schedule, and the NT-Xent temperature parameter τ (Table 6.3). Performance is evaluated using the ROC curve and the AUC, which summarizes the model's discriminative ability across all possible thresholds.

Table 6.3: Search space for hyperparameter tuning FLACL.

Parameter	Options
batch size	[32, 64]
epochs	[30, 50, 80]
learning rate	[0.0001, 0.0005, 0.001, 0.005]
weight decay	[0.0, 0.0001, 0.001]
momentum	[0.95, 0.90, 0.85]
optimizer	[SGD, Adam]
steps learning rate decay	[10, 20, 30]
gamma learning rate decay	[0.1, 0.2, 0.5]
τ	[0.07, 0.7, 1.7]

6.4.2 Experiment 1: Full Data Usage

In the first experiment, we use the complete dataset without applying any additional frame filtering, aside from the standard preprocessing steps described in Section 6.2.2. This approach ensures that the model is exposed to the full range

of natural variation present in the video data, including frames with suboptimal conditions such as motion blur or off-angle views.

6.4.3 Experiment 2: Frame Quality Filtering

The second experiment evaluates performance when only higher-quality frames are used. Frames are filtered based on facial orientation and image quality.

- *Pose:* pitch within $\pm 15^\circ$ and yaw within $\pm 30^\circ$; roll corrected during preprocessing.
- *Image quality:* BRISQUE score ≤ 40.

During a video sequence, individuals naturally move their heads, occasionally resulting in extreme angles where the face is only partially visible or entirely obscured. To address this, we estimate the head orientation for each frame using the model proposed by Hempel et al. [210]. This model estimates three key pose metrics: pitch, yaw, and roll. Based on manual inspection, we define the following thresholds for acceptable orientation: yaw within $\pm 30^\circ$ and pitch within $\pm 15^\circ$. Frames exceeding these bounds are discarded. The roll is already corrected during preprocessing by aligning the eyes horizontally, as described in Section 6.2.2.

In addition to pose filtering, we apply a no-reference image quality assessment using the BRISQUE score [211]. This metric assigns a score between 1 and 100 to each image, where lower values indicate better visual quality. We discard any frame with a BRISQUE score above 40, as these are typically blurry or distorted and may degrade model performance.

This experiment demonstrates the effect of frame quality on video-based kinship recognition while still leveraging multiple frames per individual.

6.4.4 Experiment 3: Best Frame Selection

In the final experiment, we shift from video-based to image-based kinship recognition by selecting the single best-quality frame from each video. The chosen frame is the one where the individual's head orientation is closest to a neutral position, i.e., the pitch and yaw angles are nearest to 0 degrees. This ensures that each sample represents a clear, frontal view of the subject, which is shown to be optimal for facial recognition and image-based kinship verification tasks.

By reducing each video to a single high-quality frame, this experiment isolates the effect of image-level features without temporal redundancy. In this sense, it serves as a canonical image-based recognition scenario, enabling a direct comparison with video-based approaches (Experiments 1 and 2). This also highlights the

contribution of the backbone architecture (ResNet-50) in extracting discriminative facial features from a single frame, as opposed to temporal aggregation with MARLIN in video-based setups.

Because the input format changes, we switch the backbone from MARLIN to ResNet-50. Consequently, the hyperparameter tuning process is repeated: all key parameters, including learning rate, batch size, weight decay, and fine-tuning strategy, are re-evaluated to ensure optimal performance for this image-based configuration. The results from this experiment complement those from the video-based experiments, helping to clarify how temporal information and frame quality impact kinship verification, and aligning closely with the main message of this paper regarding the importance of backbone choice and input quality.

6.5 Results

6.5.1 Overal Performance

Table 6.8 summarizes the test performance of all models and configurations, including per-relationship and non-related (NR) accuracies. Figure 6.2 visualizes the ROC curves and AUC scores for both image- and video-based models.

Table 6.1: Summary of test results for all models and configurations, including accuracy by relationship category and non-related (NR) pairs.

Model	Filter	Overall Acc.	FS Acc.	BB Acc.	FD Acc.	MD Acc.	BS Acc.	SS Acc.	MS Acc.	NR Acc.
Baseline KinFusionNet (Video)	no	63.6% ± 3.91	63.0%	61.3%	57.9%	67.0%	60.3%	58.8%	57.8%	64.5%
Baseline KinFusionNet (Video)	yes	64.0% ± 4.19	62.9%	60.6%	59.7%	68.4%	59.2%	58.7%	62.8%	66.2%
Baseline KinFusionNet (Image)	no	69.5% ± 4.82	66.8%	68.9%	63.1%	72.6%	68.3%	66.7%	64.2%	70.8%
FLACL (Video)	no	58.5% ± 4.50	59.2%	57.9%	55.2%	61.7%	59.3%	56.8%	52.7%	59.3%
FLACL (Image)	no	**70.1%** ± 5.24	66.9%	70.3%	64.2%	73.5%	69.1%	68.7%	65.8%	71.4%

From Table 6.8, the image-based FLACL model achieves the highest overall accuracy (70.1%), closely followed by KinFusionNet with best-frame selection (69.5%). Video-based models perform considerably lower, with FLACL achieving only 58.5% accuracy. These results indicate that selecting a single high-quality frame and using a strong image backbone (ResNet-50) is the most effective baseline configuration.

Compared to prior work [5], several image-based models match or exceed previously reported performance, particularly on FS, BB, and MD relationships.

6.5.2 KinFusionNet Architecture and Hyperparameter Tuning

Table 6.4 reports the results of the MLP architecture tuning phase for KinFusionNet. A single wide hidden layer performed best for both video and image

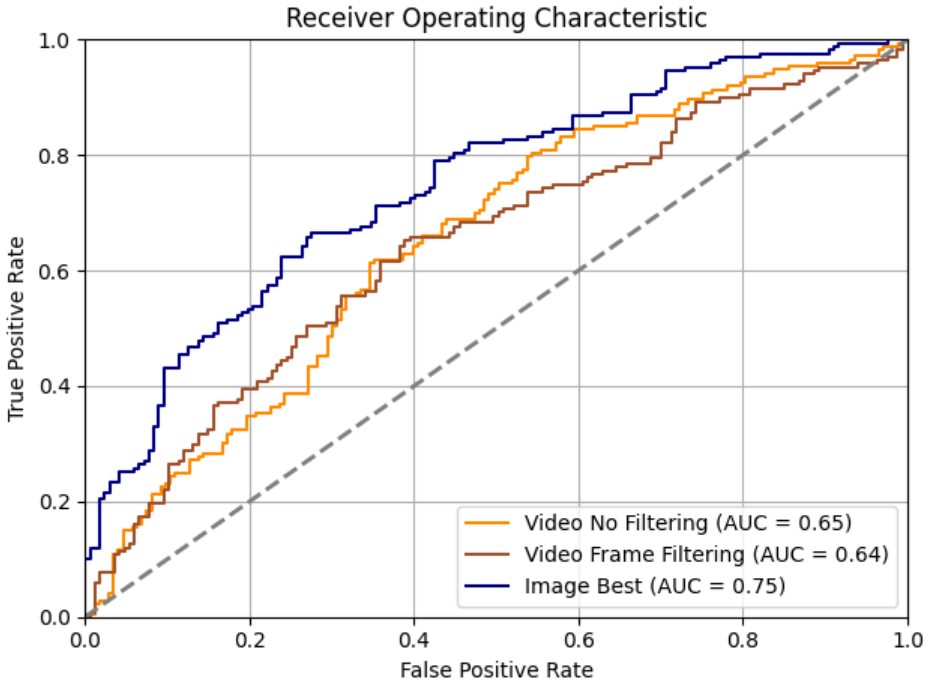


Figure 6.1: ROC curves for all baseline and FLACL models across the three experiments, including AUC scores.

backbones, suggesting that higher capacity improves interpretation of fused embeddings.

Table 6.2: Average MLP architecture tuning results for KinFusionNet on validation folds.

Backbone	Input Dim	Hidden Layers Dim	Accuracy	Precision	Recall	F1
Video	3072	5120	0.6162	0.6141	0.6339	0.6217
Image	1536	2560	0.6414	0.6519	0.6101	0.6295

Hyperparameter tuning (Table 6.5) focused on epochs, batch size, learning rate, and weight decay. Improvements were limited compared to architecture selection, indicating that architecture choice had a greater effect than hyperparameters.

We first tuned the MLP architecture separately for the MARLIN (video) and ResNet (image) backbones. As shown in Table 6.4, a single wide hidden layer performed best for both, suggesting that higher capacity improves interpretation of fused embeddings.

Table 6.3: Hyperparameter tuning results for KinFusionNet models on validation folds.

Backbone	Epochs	Batch Size	Learning Rate	Weight Decay	Accuracy	F1
Video	20	128	0.0001	0.0001	0.6021	0.6003
Image	30	32	0.00001	0.01	0.6604	0.6598

Table 6.4: Average of the MLP architecture tuning phase for the KinFusionNet models on the validation folds.

Backbone	Input Dim	Hidden Layers Dim	Accuracy	Precision	Recall	F1
Video	3072	5120	0.6162	0.6141	0.6339	0.6217
Image	1536	2560	0.6414	0.6519	0.6101	0.6295

Hyperparameter tuning covered epochs, batch size, learning rate, and weight decay. Results are summarized in Table 6.5. Improvements compared to the architecture tuning phase were limited.

Table 6.5: Averages of the hyperparameter tuning for the KinFusionNet models on the validation folds.

Backbone	Epochs	Batch Size	Learning Rate	Weight Decay	Accuracy	F1
Video	20	128	0.0001	0.0001	0.6021	0.6003
Image	30	32	0.00001	0.01	0.6604	0.6598

Test Results

Final test runs used the optimal MLP and tuned parameters. Instead of a fixed 0.5 threshold, the decision boundary was chosen from the ROC curve to maximize accuracy. Figure 6.2 shows the ROC curves and AUC scores for all experiments.

As seen in Figure 6.2, frame filtering shows no noticeable performance benefit. This is also reflected in the accuracy curves across the ten-fold cross-validation in Figure 6.3, which remain consistent regardless of filtering strategy. Among the baseline setups, selecting the best frame from each video gave the highest test accuracy.

These results show that selecting the best frame from each video yields the strongest performance among the baseline setups. Several factors may explain this. First, the ResNet model benefits from pre-training on a much larger and more diverse dataset than the MARLIN video model. This gives it a substantial advantage in generalizing to new data. Second, building a representation from

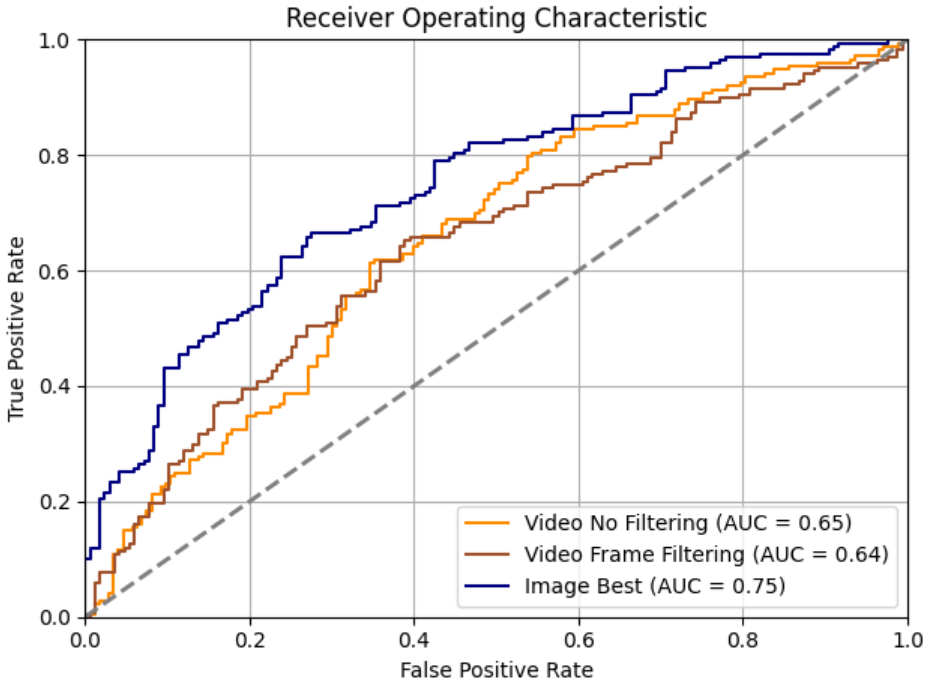


Figure 6.2: ROC curves for the baseline models across all three experiments, including AUC scores.

video requires handling more variation, such as changes in facial expression and head pose, whereas a single, well-aligned image may offer a more stable and informative input for the task.

6.5.3 FLACL Hyperparameter Tuning

FLACL introduces contrastive learning parameters, including the NT-Xent temperature τ. Table 6.6 summarizes the hyperparameter tuning results on validation folds. The image-based FLACL model reached an AUC of 0.77, while the video-based model achieved only 0.55, highlighting the benefit of image-level input and pretraining.

Test Results

Using the tuned parameters, the image-based FLACL model consistently outperformed the video-based variant. Figure 6.4 illustrates the effect of training on the image backbone: training increases the separation between kin and non-kin pairs, shifting positive pairs to higher similarity values and improving discriminative

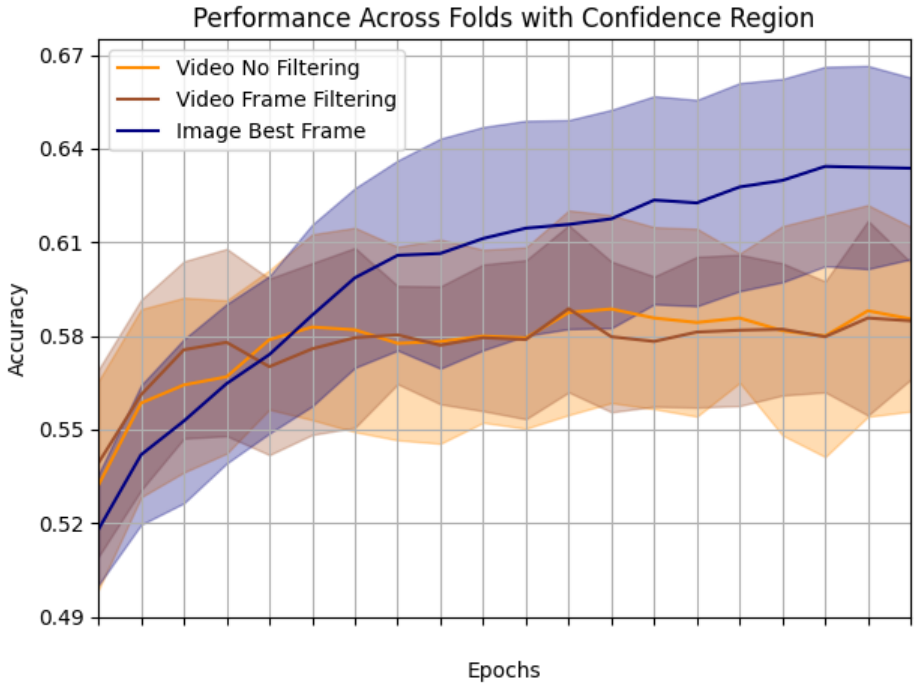


Figure 6.3: Accuracy confidence interval of KinFusionNet models over ten-fold cross validation.

power.

We compare the performance of all proposed models and benchmark them against results reported by Kefalas et al. [5]. The scores shown reflect the performance of each model after cross-validation and threshold optimization via the ROC curve.

To better understand how each model handles specific relationship types, we include per-label accuracy scores. Table 6.7 reports the distribution of relationship categories in the KAN-AV dataset.

Across all models, the image-based FLACL achieves the highest overall accuracy at 70.1% $\pm$ 5.24. KinFusionNet with frame filtering follows closely at 69.5% $\pm$ 4.82, with overlapping confidence intervals. The video-based FLACL performs considerably lower at 58.5% $\pm$ 4.50.

Compared to results reported by Kefalas et al. [5], several image-based models match or exceed previously reported scores. Detailed per-relationship accuracies for all models are shown in Table 6.8.

Table 6.6: Hyperparameter tuning for FLACL models on validation folds. AUC values are threshold-independent.

Backbone	BS	Epo	LR	WD	Mom	Wa	LRSS	Gam	τ	AUC
Video	16	100	3e-3	0	0.95	50	10	0.8	0.1	0.55
Image	32	50	1e-3	5e-4	0.95	10	10	0.8	0.1	0.77

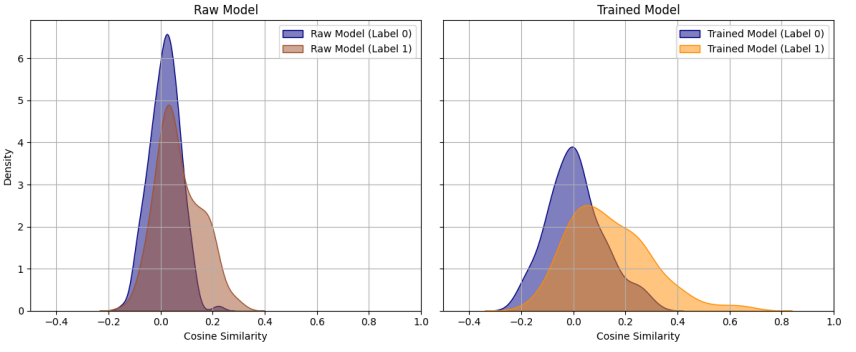


Figure 6.4: Distribution difference between the untrained and trained image backbone.

6.6 Discussion

This paper investigated the impact of pre-trained models on facial kinship classification, particularly in settings where labeled data is limited. To study this, we designed and evaluated two fine-tuning strategies: KinFusionNet, a Siamese-style classification model, and FLACL, a supervised contrastive learning framework. Both were tested using video-based and image-based inputs across three experiments, allowing us to analyze the influence of data modality, pretraining strategy, and model design.

6.6.1 Major Findings

A central finding of this work is the strong contrast between video-based and image-based kinship recognition. Across all experiments, video-based models consistently underperformed, reaching only marginally above-chance levels (58–64% accuracy). Even after introducing frame filtering to mitigate noise, performance gains were minimal. This suggests that the challenge is not restricted to noisy inputs but extends to the inherent difficulty of extracting stable, discriminative kinship cues from temporally dynamic data. The MARLIN video backbone, while well-suited for facial video reconstruction, may lack both the scale and diversity of pretraining necessary for kinship-specific generalization. These results

Table 6.7: The distribution of kinship relationships in the KAN-AV Dataset. Abbreviations used: *NR* (Non-Related), *FS* (Father–Son), *BB* (Brother–Brother), *FD* (Father–Daughter), *MD* (Mother–Daughter), *BS* (Brother–Sister), *SS* (Sister–Sister), and *MS* (Mother–Son).

Labels	**NR**	**FS**	**BB**	**FD**	**MD**	**BS**	**SS**	**MS**
Distribution	0.50	0.101	0.101	0.075	0.066	0.060	0.060	0.034

Table 6.8: Summary of test results for all models and configurations, including accuracy by relationship category and non-related (NR) pairs.

Model	**Filter**	**Overall Acc.**	**FS Acc.**	**BB Acc.**	**FD Acc.**	**MD Acc.**	**BS Acc.**	**SS Acc.**	**MS Acc.**	**NR Acc.**
Baseline KinFusionNet (Video)	no	63.6% ± 3.91	63.0%	61.3%	57.9%	67.0%	60.3%	58.8%	57.8%	64.5%
Baseline KinFusionNet (Video)	yes	64.0% ± 4.19	62.9%	60.6%	59.7%	68.4%	59.2%	58.7%	62.8%	66.2%
Baseline KinFusionNet (Image)	no	69.5% ± 4.82	66.8%	68.9%	63.1%	72.6%	68.3%	66.7%	64.2%	70.8%
FLACL (Video)	no	58.5% ± 4.50	59.2%	57.9%	55.2%	61.7%	59.3%	56.8%	52.7%	59.3%
FLACL (Image)	no	**70.1%** ± 5.24	66.9%	70.3%	64.2%	73.5%	69.1%	68.7%	65.8%	71.4%

highlight a broader challenge in kinship recognition: temporal information, although intuitively valuable, can be difficult to leverage effectively without highly specialized representations.

In contrast, switching to single high-quality frames substantially improved results. Both KinFusionNet and FLACL achieved their strongest performance in the image-based setup, with FLACL reaching 70.1% overall accuracy. This demonstrates that, under current modeling strategies, static image inputs can capture kinship-relevant features more effectively than video sequences. The image-based ResNet backbone likely benefited from extensive pretraining on large-scale identity recognition tasks, providing a richer and more transferable feature space. Interestingly, this finding challenges the assumption that video data inherently provides more useful kinship signals, suggesting instead that the quality and diversity of pretraining may outweigh the potential benefits of temporal information.

The comparison between KinFusionNet and FLACL further reveals important insights. While KinFusionNet showed solid baseline performance, FLACL demonstrated the ability to reshape the embedding space in a label-aware manner. Cosine similarity distributions before and after training confirmed that the supervised contrastive objective successfully increased separation between related and unrelated pairs. This was especially beneficial for certain kinship categories, such as father–daughter and brother–sister. However, FLACL also exhibited weaknesses, particularly in distinguishing non-related pairs, where accuracy lagged behind other relationship categories. This limitation may stem from its training setup: small batch sizes structured per family improved kin recognition but reduced the diversity of negative samples, limiting its ability to robustly model non-kin bound-

aries.

When placed in the broader research context, both KinFusionNet (image-based) and FLACL performed comparably to or slightly better than results reported by Kefalas et al. [5]. The overlap in confidence intervals, however, prevents claiming a clear state-of-the-art advantage. Instead, the results point to the conclusion that leveraging strong image-based backbones with careful fine-tuning remains a highly effective strategy for kinship recognition, while contrastive methods such as FLACL show promise but require further refinement.

6.6.2 Limitations

Several limitations of this work should be noted. First, the evaluation relied on a single backbone per modality: MARLIN for video and ResNet for images. This design provided a clear comparison between image- and video-based approaches but also restricts the scope of conclusions regarding backbone choice. The relatively lower performance of video-based models therefore cannot be attributed solely to the temporal modality; it may also reflect characteristics specific to MARLIN's pretraining objectives and dataset. As such, the results highlight the importance of backbone selection when applying transfer learning to kinship verification.

Second, the dataset itself presented challenges. While the KAN-AV dataset is balanced across families, many of the raw video frames were of low resolution, blurred, or poorly lit. The frame filtering procedure helped remove the most problematic inputs, but it did not enhance the quality of retained frames. Applying super-resolution or advanced preprocessing could further improve facial detail and strengthen model performance.

Third, the training design of FLACL introduced its own trade-offs. Organizing mini-batches by family improved positive kin coverage but reduced exposure to diverse non-kin examples. This imbalance likely contributed to the weaker performance on non-related pairs. Alternative sampling strategies—such as mixing family-specific batches with a proportion of random non-kin pairs—could address this limitation and improve generalization.

Finally, although our models achieved performance comparable to prior work, the observed accuracy remains modest. Around 30% of pairs were still misclassified, highlighting the inherent difficulty of the kinship task and the subtlety of visual kinship cues. This suggests that further advances may require both richer pretraining (e.g., identity and kinship joint objectives) and larger, more diverse kinship datasets.

6.6.3 Future Work

Building on the findings of this research, several directions could be explored to advance FKR. One of the most immediate challenges is the limited availability of high-quality video datasets. Acquiring or constructing a larger, more diverse kinship video dataset would enable a more comprehensive evaluation of video-based models and potentially allow architectures like MARLIN to realize their full potential.

In addition to data improvements, future work could also focus on enhancing model interpretability. Deep learning methods often offer little transparency into which facial features are being used to determine kinship, making it difficult to verify or trust model predictions. Incorporating explainability techniques, such as saliency maps or attention-based visualizations, could help reveal which facial regions the model relies on, providing insight into both its strengths and limitations.

Another promising research direction lies in further optimizing the contrastive learning framework. This might include combining it with supervised classification objectives or expanding it to support multi-positive training, where the model learns from multiple related individuals within the same family. These extensions could improve the model's ability to generalize and better capture the subtle visual patterns that define familial relationships.

6.7 Conclusion

This paper examined the effectiveness of pre-trained models in facial kinship recognition under limited supervision, evaluating two fine-tuning strategies—KinFusionNet and FLACL—across both video and image-based inputs. The aim was to assess how fine-tuning strategies and data modalities affect kinship classification performance.

The results show that image-based approaches consistently outperform video-based models in terms of accuracy and generalization. The best-performing model was the image-based FLACL network, which achieved an overall accuracy of 70.1% on the KAN-AV dataset. This performance was closely followed by the KinFusionNet image model using frame filtering. These results suggest that, under current modeling strategies, a single high-quality frame provides more stable and informative input for kinship verification than full video sequences.

By contrast, video-based models struggled to achieve comparable performance. Despite leveraging temporally rich data, neither KinFusionNet nor FLACL demonstrated consistent improvements, and frame filtering produced only marginal gains. One likely factor is the MARLIN backbone, which was pre-trained

for video reconstruction rather than kinship-relevant tasks, limiting its ability to capture subtle relational cues. In contrast, the ResNet backbone benefited from large-scale identity pretraining, making it more transferable to kinship classification. Contrastive learning in FLACL did help refine feature representations—particularly for certain kinship categories—and this benefit was further evident when the fine-tuned FLACL backbone improved performance in a baseline classifier. However, it remained less effective at distinguishing non-kin pairs, which limited overall accuracy.

In conclusion, this study shows that high-quality image inputs, combined with strong pre-trained backbones and targeted fine-tuning strategies, provide a reliable basis for kinship verification. Video, while intuitively appealing, did not yet offer measurable benefits, underscoring the need for improved pretraining and model architectures that can fully exploit temporal information. Future work could explore richer video backbones, multimodal fusion strategies, or kinship-specific pretraining to advance performance beyond the limits observed here.

7 General Discussion and Conclusion

Contents

7.1 Synthesis of the Main Findings

Across the dissertation, a central theme emerges: relational patterns can be effectively modeled across highly different domains (maritime behavior, text authorship, and facial imagery), but their success strongly depends on the nature of the signal, the availability of supervision, and the robustness of feature representations under distribution shift.

Chapter 2 demonstrates that supervised sequence modeling can successfully capture coordinated maritime behavior associated with drug trafficking. The proposed Long Short-Term Memory (LSTM) model achieves a PR-AUC of 0.67, with a recall of 0.97 and precision of 0.71 for detecting drop-off events. These results indicate that the model is highly sensitive to relevant behavioral patterns in AIS trajectories, correctly identifying nearly all drop-off events while producing a moderate number of false positives. Compared to prior work that relies mainly on unsupervised anomaly detection, these findings show that supervised learning can explicitly encode interaction patterns between vessels, particularly in fishing vessel contexts. At the same time, the precision–recall trade-off highlights the inherent noise and partial observability in AIS data, which limits perfect discrimination between illicit and legitimate maritime interactions.

Chapter 3 shows that AV benefits significantly from combining semantic and stylistic representations. Three architectures are evaluated: the Feature Interaction Network, Pairwise Concatenation Network, and Siamese Network. All models integrate RoBERTa embeddings with stylometric features such as sentence length, word frequency, and punctuation patterns. The best-performing model, the Siamese Network with style features, achieves an accuracy of 0.8472 and an F1-score of 0.7228. Across all architectures, adding stylometric features consistently improves performance, particularly recall. For the Feature Interaction Network, recall increases from 0.6681 to 0.6961, and for the Pairwise Concatenation Network from 0.6121 to 0.7052. These consistent gains demonstrate that stylometric features provide complementary information to contextual embeddings, even in models already using strong pretrained language representations. The results further suggest that performance gains are more strongly driven by feature augmentation than by architectural variation, reinforcing the importance of hybrid representations in AV.

Chapter 4 extends this analysis to cross-domain AV, focusing on generalization under domain shift. In the non-holdout setting, where training and testing domains differ but remain within the same dataset, the model achieves an overall score of 0.654, outperforming the PAN 2022 baseline of 0.600. Strong cross-domain performance is observed for several pairs, including essay–email (0.694

vs. 0.590), essay–text message (0.638 vs. 0.599), and email–text message (0.633 vs. 0.614), with the largest improvement of 0.104 on essay–email pairs. In the more challenging holdout setting, where entire domains are unseen during training, performance remains relatively stable for some domains, including 0.659 for speech-to-text, 0.642 for SE queries, and 0.599 for text messages, but drops significantly for handwritten text (0.489). These results indicate that stylistic representations generalize well across digital communication formats but are sensitive to modality shifts such as handwriting, where structural differences are substantially larger.

Chapter 5 investigates which facial features are most informative for KR using 40 StyleGAN2-derived features. The best-performing model, a random forest classifier, achieves the highest accuracy when using full (unmasked) face representations compared to masked variants, indicating that both facial regions contribute to prediction. Feature importance analysis shows that age-related and facial hair attributes dominate predictive performance. Features such as young, no beard, mustache, and arched eyebrows consistently rank among the most important across models. These results suggest that kinship prediction in this setting is strongly influenced by demographic and grooming-related cues rather than localized identity regions such as the eyes, highlighting a divergence between human perceptual assumptions and machine-learned signals.

Chapter 6 evaluates the effectiveness of pre-trained models and contrastive learning for KR under limited supervision. Two approaches are compared: KinFusionNet, a Siamese-style classifier, and FLACL, a supervised contrastive learning framework. Using both image-based and video-based inputs on the KAN-AV dataset, results show that image-based models consistently outperform video-based models. The best-performing configuration is the image-based FLACL model, achieving 70.1% accuracy and an AUC of 0.77. KinFusionNet with frame filtering performs competitively, reaching 69.5% accuracy, closely approaching FLACL performance. Backbone analysis shows that ResNet, pretrained on large identity datasets, transfers more effectively to KR than MARLIN, which is optimized for video reconstruction tasks. This is reflected in the performance gap between FLACL variants, where accuracy drops from 70.1% (image) to 58.5% (video), and AUC decreases from 0.77 to 0.55. These results highlight both the limitations of temporal modeling in KR and the importance of strong identity-focused pretraining.

7.2 Overall Contribution

While the individual chapters address different application domains, together they provide a broader perspective on the challenges and opportunities of machine learning in real-world scenarios. At first sight, maritime drug trafficking detection, AV, and KR appear unrelated. However, all three domains require models to infer relationships that are not directly observable from the available data. Instead, these relationships are expressed through indirect signals, such as vessel movement patterns in AIS trajectories, stylistic and semantic properties of text, and facial representations derived from images or video. This shared structure forms the conceptual backbone of the dissertation.

A central insight emerging from the dissertation is that relational inference is possible across very different domains, but that the appropriate representation is highly dependent on both the type of relationship and the data modality. This is evident in the use of sequential AIS trajectories for modeling coordinated maritime behaviour in Chapter 2, in the combination of semantic embeddings and stylometric features for AV in Chapters 3 and 4, and in pretrained facial representations and feature-based encodings for KR in Chapters 5 and 6. Across these cases, the results consistently show that no single representation type is universally optimal; instead, performance depends on aligning the representation with the structure of the underlying data and the nature of the relational signal.

A second key observation is that generalization is the central challenge for practical deployment. Chapter 4 explicitly demonstrates this in the cross-domain AV setting, where performance varies substantially under domain shift and degrades most strongly under modality changes such as handwritten text. However, the same issue is implicitly present in the other domains as well. In maritime surveillance, vessel behaviour is inherently non-stationary across regions, time periods, and operational contexts, which limits the stability of learned patterns. Similarly, in KR, performance is affected by changes in demographic composition, image quality, and recording conditions. Taken together, these results show that robustness under distribution shift is not a domain-specific issue, but a general constraint that shapes the practical applicability of relational inference systems.

A third insight is that complementary information sources can improve robustness, although this effect is not uniform across all settings. In AV, combining semantic representations with stylometric features consistently improves performance, suggesting that these signals capture different but partially overlapping aspects of writing style. In KR, pretrained image representations and contrastive learning similarly provide complementary benefits by capturing identity-related structure beyond handcrafted features. However, the results also indicate that

such complementarity is not guaranteed: its effectiveness depends on whether the combined signals capture genuinely independent aspects of the relational structure rather than redundant or noisy information. This suggests that complementarity is most effective when it reflects meaningful diversity in the underlying representation space rather than simply increasing feature quantity.

From a broader perspective, the dissertation contributes to the growing body of research that seeks to move machine learning beyond controlled laboratory settings and toward practical deployment in complex environments. The results show that relational inference can be successfully applied across highly diverse modalities, including trajectories, text, and images, while also revealing the limitations that arise when data are imperfect, heterogeneous, and subject to change. Collectively, the chapters therefore contribute not only to maritime security, AV, and KR individually, but also to a broader understanding of how machine learning systems can be designed to operate effectively in real-world scenarios. The overarching message is that robust relational inference requires a combination of informative representations, complementary information sources, generalization capabilities, and domain knowledge. These principles extend beyond the specific applications studied in this dissertation and provide general guidance for designing machine learning systems intended for real-world use.

Research Questions Answered

Across the dissertation, the research questions are addressed as follows. Research question 1 is answered by demonstrating how relational patterns can be modeled across vessel trajectories, textual data, and facial imagery using domain-specific architectures. Research question 2 is addressed by showing that meaningful representations can be learned under conditions of limited and imbalanced data through transfer learning and hybrid feature representations. Research question 3 is primarily addressed in the AV setting, where models are shown to remain robust under cross-domain evaluation and distribution shift. Research question 4 is addressed through the use of complementary feature representations within AV and KR settings, showing that combining different representation types can support relational inference in practice. Research question 5 is addressed in the KR setting, where interpretability analysis reveals which facial features contribute to model predictions in high-stakes biometric applications.

7.3 Practical Implications

The results presented throughout this dissertation have several practical implications. In the maritime domain, automated analysis of AIS data can support mari-

time authorities by identifying potentially suspicious vessel behavior and reducing the amount of manual monitoring required. In AV, automated methods may assist forensic investigations, plagiarism detection, and online safety applications. In KR, machine learning approaches may support research in biometrics and family analysis.

Importantly, the models developed in this dissertation are not intended to replace human experts. Instead, they function as decision-support systems that help users process large amounts of information and identify cases that merit further investigation. By reducing workload and improving efficiency, such systems may allow experts to focus their attention on the most relevant cases while maintaining responsibility for final decisions.

7.4 Ethical Considerations

The application of machine learning methods to real-world relational inference problems raises important ethical considerations. Although the domains studied in this dissertation differ substantially, they share concerns regarding privacy, responsible use of data, and the societal consequences of automated decision-making. Because the methods developed in this work operate on human-generated or human-related data, ethical reflection is essential when considering their deployment in practice.

In the maritime domain, the use of machine learning for vessel monitoring introduces ethical and legal considerations related to surveillance and law enforcement decision-making. AIS trajectories are originally intended for navigation safety rather than behavioural inference, and their secondary use for detecting suspicious activity raises questions about proportionality and transparency. The LSTM model developed in this work achieves high recall (0.97) but lower precision (0.71), meaning that a non-trivial number of false positives will occur. In a law enforcement context, such false positives may lead to additional surveillance, inspections, or intervention of vessels that are not actually involved in illicit activity. This makes the risk of disproportionate or misdirected action a central ethical concern. As a result, the system should not be interpreted as a decision-making tool, but as a support mechanism requiring human oversight, auditability of model outputs, and clear accountability structures. These considerations align with emerging regulatory expectations under the EU AI Act [212], where such systems may fall under high-risk categories due to their potential impact on individuals and commercial activity.

In AV, ethical concerns primarily relate to privacy and consent. False authorship attribution in real-world applications can have serious consequences, particularly

in forensic, academic, or workplace settings where writing style may be used as evidence. The risk of de-anonymisation or unintended inference of authorship from private communications also raises privacy concerns, especially when applied without explicit consent. Furthermore, performance differences across domains suggest potential biases related to genre, writing medium, and text length, meaning that certain groups or communication styles may be systematically more difficult to classify. These limitations highlight the importance of restricting such systems to assistive contexts (e.g., plagiarism detection support) rather than standalone adjudicative use.

KR raises ethical issues related to biometric data and personal identity. Facial images and video data constitute sensitive personal data under the GDPR [213], and the KAN-AV dataset and FIW-related kinship benchmarks used in this research originate from curated collections of publicly available imagery, but still raise questions regarding downstream use and representativeness. Automated kinship inference can have direct implications for individuals and families, particularly if used outside research contexts. Another limitation of this work is that demographic performance gaps were not explicitly evaluated, meaning that potential disparities across gender, ethnicity, or age groups cannot be quantified in this study and remain an open risk.

Across all three domains, a shared ethical theme emerges: systems that infer relationships between entities or individuals should not be treated as autonomous decision-makers. Instead, their primary role lies in decision support under uncertainty, where outputs must be interpreted in context by human experts. This is particularly important given that all three domains involve non-trivial error modes—false positives in maritime detection, misattribution in AV, and biased or proxy-driven predictions in KR.

From a regulatory perspective, these applications fall within domains increasingly addressed by frameworks such as the GDPR and the EU AI Act, particularly due to their use of surveillance data, biometric information, and high-impact inference tasks. These frameworks emphasize principles such as data minimisation, transparency, human oversight, and accountability, all of which are directly relevant to the systems developed in this dissertation.

More broadly, the increasing use of relational inference in complex environments highlights the need to balance predictive performance with responsible deployment. Beyond improving model accuracy, it is essential to ensure that such systems are used in ways that respect privacy, acknowledge uncertainty, and prevent over-reliance on automated outputs in high-stakes contexts involving people, organizations, or legal decisions.

7.5 Limitations and Future Work

Despite these contributions, several limitations remain at the level of empirical scope and evaluation. Domain shift is primarily evaluated in AV, while comparable cross-domain experiments are not conducted in the maritime setting or in KR, where experiments focus on controlled variations in modality and model architecture. As a result, the generalization behavior of the proposed approaches in these settings remains less explored under varying environmental or demographic conditions. Similarly, interpretability is mainly studied in KR, leaving open questions regarding explainability and feature attribution in textual and behavioral domains.

Future work could address these limitations by extending evaluation settings across domains. In the maritime domain, this could include cross-region evaluation of vessel behavior under different operational environments. For KR, further research could explore kinship-specific pretraining strategies using video data to better capture temporal facial dynamics. In AV, extending models to longer-context settings would allow better modeling of document-level structure beyond fixed-length inputs. Finally, interpretability methods could be generalized beyond KR to provide more transparent decision-making across all three application areas.

Bibliography

[1] K. Wolsing, L. Roepert, J. Bauer, and K. Wehrle, "Anomaly Detection in Maritime AIS Tracks: A Review of Recent Approaches", *Journal of Marine Science and Engineering*, volume 10, number 1, page 112, Jan. 2022. DOI: 10.3390/jmse10010112.

[2] I. M. Organization, "AIS transponders", https://www.imo.org/en/OurWork/Safety/Pages/AIS.aspx.

[3] Webis Group, "PAN @ CLEF Shared Tasks on Authorship Analysis, Computational Ethics, and Originality", https://pan.webis.de/shared-tasks.html, Accessed: 2026-03-09, 2024.

[4] J. P. Robinson, M. Shao, Y. Wu, and Y. Fu, "Family in the Wild (FIW): A Large-scale Kinship Recognition Database", *CoRR*, volume abs/1604.02182, 2016.

[5] T. Kefalas et al., "KAN-AV Dataset for Audio-Visual Face and Speech Analysis in the Wild", *Image and Vision Computing*, volume 140, page 104 839, Dec. 2023. DOI: 10.1016/j.imavis.2023.104839.

[6] J. P. Robinson, Z. Khan, Y. Yin, M. Shao, and Y. Fu, "Families in Wild Multimedia: A Multimodal Database for Recognizing Kinship", *IEEE Transactions on Multimedia*, volume 24, pages 3582–3594, 2022. DOI: 10.1109/TMM.2021.3103074.

[7] X. Wu, X. Zhang, X. Feng, M. Bordallo López, and L. Liu, "Audio-Visual Kinship Verification: A New Dataset and a Unified Adaptive Adversarial Multimodal Learning Approach", *IEEE Transactions on Cyber-*

netics, volume 54, pages 1523–1536, Mar. 2024. DOI: 10.1109/TCYB.2022.3220040.

[8] T. H. in Sydney and J. G. in Singapore, "Cocaine Haul Sparks Hunt for Australian trio rescued at sea", https://www.bbc.com/news/world-australia-64645184, Feb. 2023.

[9] S. Mehlbaum, A. K. v. den, A. Verweij, and A. Wester, *Zuiver op de graat?: Over de betrokkenheid van de visserij bij maritieme drugssmokkel*. Politie & Wetenschap, 2021.

[10] M. v. I. e. Waterstaat, "AIS: Verkeersinformatie Scheepvaart", https://www.rijkswaterstaat.nl/water/scheepvaart, Mar. 2022.

[11] D. Nguyen, R. Vadaine, G. Hajduch, R. Garello, and R. Fablet, "GeoTrackNet-A Maritime Anomaly Detector using Probabilistic Neural Network Representation of AIS Tracks and A Contrario Detection", *IEEE Transactions on Intelligent Transportation Systems*, volume 23, number 6, pages 5655–5667, Dec. 2019.

[12] E. de Coning, *Transnational Organized Crime in the Fishing Industry*. United Nations Office on Drugs and Crime (UNODC), 2011, pages 126–138.

[13] R. O. Lane, D. A. Nevell, S. D. Hayward, and T. W. Beaney, "Maritime anomaly detection and threat assessment", in *2010 13th International Conference on Information Fusion*, 2010, pages 1–8. DOI: 10.1109/ICIF.2010.5711998.

[14] J. Venskus, P. Treigys, and J. Markevičiūtė, "Unsupervised marine vessel trajectory prediction using LSTM network and wild bootstrapping techniques", *Nonlinear Analysis: Modelling and Control*, volume 26, pages 718–737, Jul. 2021. DOI: 10.15388/namc.2021.26.23056.

[15] S. K. Singh and F. Heymann, "Machine Learning-Assisted Anomaly Detection in Maritime Navigation using AIS Data", in *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2020, pages 832–838. DOI: 10.1109/PLANS46316.2020.9109806.

[16] D. Liu, X. Wang, Y. Cai, Z. Liu, and Z.-J. Liu, "A Novel Framework of Real-Time Regional Collision Risk Prediction Based on the RNN Approach", *Journal of Marine Science and Engineering*, volume 8, number 3, pages 139–149, 2020. DOI: 10.3390/jmse8030224.

[17] Z. Fang, H. Yu, R. Ke, S.-L. Shaw, and G. Peng, "Automatic Identification System-Based Approach for Assessing the Near-Miss Collision Risk Dynamics of Ships in Ports", *IEEE Transactions on Intelligent Transportation Systems*, volume 20, number 2, pages 534–543, 2019. DOI: 10.1109/TITS.2018.2816122.

[18] S. Arasteh et al., "Fishing Vessels Activity Detection from Longitudinal AIS Data", in *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*, series SIGSPATIAL '20, Seattle, WA, USA: Association for Computing Machinery, 2020, pages 347–356. DOI: 10.1145/3397536.3422267.

[19] T. Eriksen, H. Greidanus, and C. Delaney, "Metrics and provider-based results for completeness and temporal resolution of satellite-based AIS services", *Marine Policy*, volume 93, pages 80–92, 2018. DOI: https://doi.org/10.1016/j.marpol.2018.03.028.

[20] N. C. Greig, E. M. Hines, S. Cope, and X. Liu, "Using Satellite AIS to Analyze Vessel Speeds Off the Coast of Washington State, U.S., as a Risk Analysis for Cetacean-Vessel Collisions", *Frontiers in Marine Science*, volume 7, page 109, 2020. DOI: 10.3389/fmars.2020.00109.

[21] I. M. Organization, "RESOLUTION MSC.74(69) (adopted on 12 May 1998) Adoption of New and Amended Performance Standards", https://www.imorules.com/MSCRES_74.69.html, 1998.

[22] J. H. Ford, D. Peel, D. Kroodsma, B. D. Hardesty, U. Rosebrock, and C. Wilcox, "Detecting suspicious activities at sea based on anomalies in Automatic Identification Systems Transmissions", https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6084947/.

[23] M. S. G. BV, "Made smart group navigator: The Fastest Nautical Charts", https://www.madesmart.nl/, 2014.

[24] R. van den Goorbergh, M. van Smeden, D. Timmerman, and B. Van Calster, "The harm of class imbalance corrections for risk prediction models: illustration and simulation using logistic regression", *Journal of the American Medical Informatics Association*, volume 29, number 9, pages 1525–1534, Jun. 2022. DOI: 10.1093/jamia/ocac093.

[25] M. S. K. Yanmin Sun Andrew Wong, "Classification of imbalanced data: a review", *International Journal of Pattern Recognition and Artificial Intelligence*, volume 23, pages 687–719, Nov. 2011. DOI: 10.1142/S0218001409007326.

[26] B. Krawczyk, "Learning from imbalanced data: Open challenges and future directions", *Progress in Artificial Intelligence*, volume 5, pages 221–232, Apr. 2016. DOI: 10.1007/s13748-016-0094-0.

[27] C.-H. Yang, G.-C. Lin, C.-H. Wu, Y.-H. Liu, Y.-C. Wang, and K.-C. Chen, "Deep Learning for Vessel Trajectory Prediction Using Clustered AIS Data", *Mathematics*, volume 10, page 2936, Aug. 2022. DOI: 10.3390/math10162936.

[28] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory", *Neural computation*, volume 9, number 8, pages 1735–1780, 1997.

[29] J. Bergstra, D. Yamins, and D. D. Cox, "Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures", in *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, series ICML'13, Atlanta, GA, USA: JMLR.org, 2013, I–115–I–123.

[30] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization", *International Conference on Learning Representations*, Dec. 2014.

[31] F. Chollet et al., "Keras", https://github.com/fchollet/keras, 2015.

[32] M. R. Rezaei-Dastjerdehei, A. Mijani, and E. Fatemizadeh, "Addressing Imbalance in Multi-Label Classification Using Weighted Cross Entropy Loss Function", in *2020 27th National and 5th International Iranian Conference on Biomedical Engineering (ICBME)*, 2020, pages 333–338. DOI: 10.1109/ICBME51989.2020.9319440.

[33] E. van de Bijl, J. Klein, J. Pries, S. Bhulai, M. Hoogendoorn, and R. D. van der Mei, "The Dutch Draw: Constructing a Universal Baseline for Binary Prediction Models", 2025.

[34] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research*, volume 12, pages 2825–2830, 2011.

[35] J. Brownlee, "A Gentle Introduction to Mini-Batch Gradient Descent and How to Configure Batch Size", https://machinelearningmastery.com/gentle-introduction-mini-batch-gradient-descent-configure-batch-size/, Aug. 2019.

[36] J. Bevendorff et al., "Overview of pan 2020: Authorship verification, celebrity profiling, profiling fake news spreaders on twitter, and style change detection", in *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 11th International Conference of the CLEF Association, CLEF 2020, Thessaloniki, Greece, September 22–25, 2020, Proceedings 11*, Springer, 2020, pages 372–383.

[37] E. Stamatatos et al., "Overview of the Authorship Verification Task at PAN 2022", English, *CEUR workshop proceedings*, volume 3180, pages 2301–2313, Sep. 2022. [Online]. Available: https://ceur-ws.org/Vol-3180/.

[38] P. Juola, "Verifying authorship for forensic purposes: A computational protocol and its validation", *Forensic Science International*, volume 325, page 110 824, 2021.

[39] K. Lagutina, N. Lagutina, E. Boychuk, V. Larionov, and I. Paramonov, "Authorship verification of literary texts with rhythm features", in *2021 28th Conference of Open Innovations Association (FRUCT)*, IEEE, 2021, pages 240–251.

[40] B. Stein, N. Lipka, and P. Prettenhofer, "Intrinsic plagiarism analysis", *Language Resources and Evaluation*, volume 45, pages 63–82, 2011.

[41] T. Yılmaz and T. Scheffler, "Song authorship attribution: a lyrics and rhyme based approach", *International Journal of Digital Humanities*, volume 5, number 1, pages 29–44, 2023.

[42] A. Misini, A. Kadriu, and E. Canhasi, "Authorship classification techniques: Bridging textual domains and languages", *International Journal on Information Technologies and Security*, volume 16, number 1, pages 27–38, 2024.

[43] Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach", 2019. [Online]. Available: https://arxiv.org/abs/1907.11692.

[44] M. T. Zamir, M. A. Ayub, A. Gul, N. Ahmad, and K. Ahmad, "Stylometry Analysis of Multi-authored Documents for Authorship and Author Style Change Detection", 2024. [Online]. Available: https://arxiv.org/abs/2401.06752.

[45] E. Stamatatos et al., "Overview of the Authorship Verification Task at PAN 2023", in *Conference and Labs of the Evaluation Forum*, 2023.

[46] M. L. Brocardo, I. Traore, S. Saad, and I. Woungang, "Authorship verification for short messages using stylometry", in *2013 International Conference on Computer, Information and Telecommunication Systems (CITS)*, 2013, pages 1–6. DOI: 10.1109/CITS.2013.6705711.

[47] M. L. Brocardo, I. Traore, S. Saad, and I. Woungang, "Authorship verification for short messages using stylometry", in *2013 International Conference on Computer, Information and Telecommunication Systems (CITS)*, IEEE, 2013, pages 1–6.

[48] D. C. Castro, Y. A. Arcia, M. P. Brioso, and R. Muñoz, "Authorship verification, average similarity analysis", in *Proceedings of the international conference recent advances in natural language processing*, 2015, pages 84–90.

[49] N. Potha and E. Stamatatos, "An improved impostors method for authorship verification", in *International conference of the cross-language evaluation forum for European languages*, Springer, 2017, pages 138–144.

[50] N. Potha and E. Stamatatos, "Improved algorithms for extrinsic author verification", *Knowledge and Information Systems*, volume 62, number 5, pages 1903–1921, 2020.

[51] T. Alsanoosy, B. Shalbi, and A. Noor, "Authorship Attribution for English Short Texts", *Engineering, Technology & Applied Science Research*, volume 14, number 5, pages 16 419–16 426, 2024.

[52] A. Abbasi, A. R. Javed, F. Iqbal, Z. Jalil, T. R. Gadekallu, and N. Kryvinska, "Authorship identification using ensemble learning", *Scientific reports*, volume 12, number 1, page 9537, 2022.

[53] A. Abedzadeh, R. Ramezani, and A. Fatemi, "A Weighted TF-IDF-based Approach for Authorship Attribution", in *2021 11th International Conference on Computer Engineering and Knowledge (ICCKE)*, IEEE, 2021, pages 188–193.

[54] J. Tyo, B. Dhingra, and Z. C. Lipton, "Siamese Bert for Authorship Verification." In *CLEF (Working Notes)*, 2021, pages 2169–2177.

[55] J. Tyo, B. Dhingra, and Z. C. Lipton, "On the state of the art in authorship attribution and authorship verification", *arXiv preprint arXiv:2209.06869*, 2022.

[56] A. Manolache, F. Brad, E. Burceanu, A. Barbalau, R. Ionescu, and M. Popescu, "Transferring bert-like transformers' knowledge for authorship verification", *arXiv preprint arXiv:2112.05125*, 2021.

[57] K. Jones, J. R. Nurse, and S. Li, "Are you robert or roberta? deceiving online authorship attribution models using neural text generators", in *Proceedings of the International AAAI Conference on Web and Social Media*, volume 16, 2022, pages 429–440.

[58] X. Wang and M. Iwaihara, "Integrating RoBERTa Fine-Tuning and User Writing Styles for Authorship Attribution of Short Texts", in *Web and Big Data*, L. H. U, M. Spaniol, Y. Sakurai, and J. Chen, Eds., Cham: Springer International Publishing, 2021, pages 413–421.

[59] M. Najafi and E. Tavan, "Text-to-Text Transformer in Authorship Verification Via Stylistic and Semantical Analysis." In *CLEF (Working Notes)*, 2022, pages 2607–2616.

[60] B. Boenninghoff, R. Nickel, and D. Kolossa, "O2D2: Out-Of-Distribution Detector to Capture Undecidable Trials in Authorship Verification—Notebook for PAN at CLEF 2021", in *CLEF 2021 Labs and Workshops, Notebook Papers*, G. Faggioli, N. Ferro, A. Joly, M. Maistro, and F. Piroi, Eds., CEUR-WS.org, Sep. 2021.

[61] J. Weerasinghe and R. Greenstadt, "Feature Vector Difference based Neural Network and Logistic Regression Models for Authorship Verification—Notebook for PAN at CLEF 2020", in *CLEF 2020 Labs and Workshops, Notebook Papers*, L. Cappellato, C. Eickhoff, N. Ferro, and A. Névéol, Eds., CEUR-WS.org, Sep. 2020.

[62] A. Kipnis, "Higher Criticism as an Unsupervised Authorship Discriminator." In *CLEF (Working Notes)*, 2020.

[63] S. Konstantinou, J. Li, and A. Zinonos, "Different Encoding Approaches for Authorship Verification", in *Proceedings of the Working Notes of*

CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022, G. Faggioli, N. Ferro, A. Hanbury, and M. Potthast, Eds., series CEUR Workshop Proceedings, volume 3180, CEUR-WS.org, 2022, pages 2532–2540.

[64] J. A. Martinez-Galicia, D. Embarcadero-Ruiz, A. Ríos-Orduña, and H. Gómez-Adorno, "Graph-based siamese network for authorship verification", in *CLEF 2022 Labs and Workshops, Notebook Papers*, 2022, pages 2594–2606.

[65] M. Crespo-Sanchez, H. Gómez-Adorno, I. Lopez-Arevalo, E. Aldana-Bobadilla, K. Salas-Jimenez, and J. Cortes-Lopez, "A content spectral-based analysis for authorship verification." In *CLEF (Working Notes)*, 2022, pages 2416–2425.

[66] M. Huang et al., "Authorship verification Based On Fully Interacted Text Segments." In *CLEF (Working Notes)*, 2022, pages 2491–2495.

[67] Z. Lei, H. Qi, Y. Han, Z. Peng, and M. Huang, "Application of BERT in author verification task." In *CLEF (Working Notes)*, 2022, pages 2560–2564.

[68] Y. Ye, Y. Han, Z. Peng, M. Huang, L. Kong, and Z. Han, "Authorship verification using convolutional neural network", in *CLEF (Working Notes)*, 2022.

[69] J. Schler, M. Koppel, S. Argamon, and J. Pennebaker, "Effects of Age and Gender on Blogging.", Jan. 2006, pages 199–205.

[70] R. Flesch, "A New Readability Yardstick", *Journal of Applied Psychology*, volume 32, number 3, pages 221–233, Jun. 1948. DOI: 10.1037/h0057532.

[71] F. Chollet et al., "Keras", https://keras.io, 2015.

[72] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation", *Journal of Machine Learning Research*, volume 3, pages 993–1022, Mar. 2003.

[73] P. Shrestha, S. Sierra, F. González, M. Montes, P. Rosso, and T. Solorio, "Convolutional Neural Networks for Authorship Attribution of Short Texts", in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, M. Lapata, P. Blunsom, and A. Koller, Eds., Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pages 669–674.

[74] T. Neal, K. Sundararajan, A. Fatima, Y. Yan, Y. Xiang, and D. Woodard, "Surveying stylometry techniques and applications", *ACM Computing Surveys (CSuR)*, volume 50, number 6, pages 1–36, 2017.

[75] V. Cammarota, S. Bozza, C.-A. Roten, and F. Taroni, "Stylometry and forensic science: A literature review", *Forensic Science International:*

Synergy, volume 9, page 100 481, 2024. DOI: https://doi.org/10.1016/j.fsisyn.2024.100481.

[76] E. Stamatatos, "A survey of modern authorship attribution methods", *Journal of the American Society for information Science and Technology*, volume 60, number 3, pages 538–556, 2009.

[77] A. Misini, A. Kadriu, and E. Canhasi, "A survey on authorship analysis tasks and techniques", *Seeu Review*, volume 17, number 2, pages 153–167, 2022.

[78] X. He, A. H. Lashkari, N. Vombatkere, and D. P. Sharma, "Authorship attribution methods, challenges, and future research directions: A comprehensive survey", *Information*, volume 15, number 3, page 131, 2024.

[79] K. Luyckx and W. Daelemans, "Authorship attribution and verification with many authors and limited data", in *Proceedings of the 22nd international conference on computational linguistics (COLING 2008)*, 2008, pages 513–520.

[80] K. Lagutina et al., "A survey on stylometric text features", in *2019 25th Conference of Open Innovations Association (FRUCT)*, IEEE, 2019, pages 184–195.

[81] R. Futrzynski, "Author classification as pre-training for pairwise authorship verification." In *CLEF (Working Notes)*, 2021, pages 1945–1952.

[82] G. Barlas and E. Stamatatos, "Cross-domain authorship attribution using pre-trained language models", in *IFIP International Conference on Artificial Intelligence Applications and Innovations*, Springer, 2020, pages 255–266.

[83] J. Bevendorff et al., "Overview of PAN 2021: authorship verification, profiling hate speech spreaders on twitter, and style change detection", in *International conference of the cross-language evaluation forum for european languages*, Springer, 2021, pages 419–431.

[84] M. Kestemont et al., "Overview of the Authorship Verification Task at PAN 2022", in *CLEF 2022 Working Notes*, CEUR-WS.org, 2022.

[85] J. Bevendorff et al., "Overview of pan 2023: Authorship verification, multi-author writing style analysis, profiling cryptocurrency influencers, and trigger detection: Condensed lab overview", in *International Conference of the Cross-Language Evaluation Forum for European Languages*, Springer, 2023, pages 459–481.

[86] E. Stamatatos et al., "Overview of the authorship verification task at PAN 2022", in *CEUR workshop proceedings*, volume 3180, 2022, pages 2301–2313.

[87] J. Tyo, B. Dhingra, and Z. C. Lipton, "On the State of the Art in Authorship Attribution and Authorship Verification", 2022. [Online]. Available: https://arxiv.org/abs/2209.06869.

[88] P. Petropoulos, "Contrastive Learning for Authorship Verification using BERT and Bi-LSTM in a Siamese Architecture." In *CLEF (Working Notes)*, 2023, pages 2734–2741.

[89] M. Guo, Z. Han, H. Chen, and H. Qi, "A Contrastive Learning of Sample Pairs for Authorship Verification." In *CLEF (Working Notes)*, 2023, pages 2608–2612.

[90] M. Kestemont et al., "Overview of the author identification task at PAN-2018: cross-domain authorship attribution and style change detection", in *Working Notes Papers of the CLEF 2018 Evaluation Labs. Avignon, France, September 10-14, 2018/Cappellato, Linda [edit.]; et al.*, 2018, pages 1–25.

[91] Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach", Jul. 2019. DOI: 10.48550/arXiv.1907.11692.

[92] X. Hu, W. Ou, S. Acharya, S. H. Ding, R. D'Gama, and H. Yu, "TDRLM: Stylometric learning for authorship verification by Topic-Debiasing", *Expert Systems with Applications*, volume 233, page 120 745, 2023. DOI: https://doi.org/10.1016/j.eswa.2023.120745.

[93] B. van Leeuwen, S. Bhulai, and R. D. van der Mei, "Combining style and semantics for robust authorship verification", *Machine Learning with Applications*, volume 22, page 100 732, 2025. DOI: https://doi.org/10.1016/j.mlwa.2025.100732.

[94] A. Hermans, L. Beyer, and B. Leibe, "In Defense of the Triplet Loss for Person Re-Identification", 2017. [Online]. Available: https://arxiv.org/abs/1703.07737.

[95] PAN Lab, "PAN 2022 Authorship Verification Dataset", https://pan.webis.de/data.html, Accessed: 2026-03-06, 2022.

[96] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library", in *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pages 8024–8035.

[97] R. Peñas and P. Rodrígo, "Evaluating systems for authorship verification", in *CLEF (Online Working Notes/Labs/Workshop)*, Introduced the c@1 score for authorship verification evaluations, 2011.

[98] B. E. van Leeuwen, A. Gansekoele, J. Pries, E. van de Bijl, and J. Klein, "Explainable Kinship: The Importance of Facial Features in Kinship Recognition", in *IARIA Congress Proceedings*, 2022, pages 54–60.

[99] J. Brownlee, *Deep Learning for Computer Vision: Image Classification, Object Detection, and Face Recognition in Python*. Machine Learning

Mastery, 2019. [Online]. Available: https://books.google.nl/books?id=DOamDwAAQBAJ.

[100] R. Szeliski, *Computer Vision: Algorithms and Applications*. Springer London, 2010. [Online]. Available: https://books.google.nl/books?id=bXzAlkODwa8C.

[101] S. A. Papert, "The Summer Vision Project", 1966. [Online]. Available: http://hdl.handle.net/1721.1/6125.

[102] "A brief history of facial recognition - NEC New Zealand", Accessed: 2022-10-21, May 2020. [Online]. Available: https://www.nec.co.nz/market-leadership/publications-media/a-brief-history-of-facial-recognition/.

[103] E. Seselja, "How the Red Cross and a radio reconnected a family torn apart by conflict - ABC News", Accessed: 2022-10-21, Aug. 2021. [Online]. Available: https://www.abc.net.au/news/2021-08-29/red-cross-reconnect-family-separated-by-conflict-after-16-years-/100413214.

[104] F. Schroff, T. Treibitz, D. Kriegman, and S. Belongie, "Pose, illumination and expression invariant pairwise face-similarity measure via Doppelgänger list comparison", in *International Conference on Computer Vision*, 2011, pages 2494–2501. DOI: 10.1109/ICCV.2011.6126535.

[105] H. Lamba, A. Sarkar, M. Vatsa, R. Singh, and A. Noore, "Face recognition for look-alikes: A preliminary study", in *International Joint Conference on Biometrics (IJCB)*, 2011, pages 1–6. DOI: 10.1109/IJCB.2011.6117520.

[106] N. L. Segal, J. L. Graham, and U. Ettinger, "Unrelated look-alikes: Replicated study of personality similarity and qualitative findings on social relatedness", *Personality and Individual Differences*, volume 55, number 2, pages 169–174, 2013.

[107] G. Guo and X. Wang, "Kinship Measurement on Salient Facial Features", *IEEE Transactions on Instrumentation and Measurement*, volume 61, number 8, pages 2322–2325, 2012.

[108] M. F. Dal Martello and L. T. Maloney, "Lateralization of kin recognition signals in the human face", *Journal of Vision*, volume 10, number 8, 2010.

[109] M. F. Dal Martello and L. T. Maloney, "Where are kin recognition signals in the human face?" *Journal of Vision*, volume 6, number 12, 2006.

[110] J. Lu, X. Zhou, Y. Tan, Y. Shang, and J. Zhou, "Neighborhood Repulsed Metric Learning for Kinship Verification", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 36, number 2, pages 331–345, 2014.

[111] L. M. DeBruine, F. G. Smith, B. C. Jones, S. C. Roberts, M. Petrie, and T. D. Spector, "Kin recognition signals in adult faces", *Vision Research*, volume 49, number 1, pages 38–43, 2009.

[112] G. Kaminski, S. Dridi, C. Graff, and E. Gentaz, "Human ability to detect kinship in strangers' faces: effects of the degree of relatedness", *Proceedings of the Royal Society B: Biological Sciences*, volume 276, number 1670, pages 3193–3200, 2009.

[113] A. Alexandra, P. Fanny, M. Allan, M. Ulrich, and R. Michel, "Identification of visual paternity cues in humans", *Biology Letters*, volume 10, number 4, 2014.

[114] L. T. Maloney and M. F. Dal Martello, "Kin recognition and the perceived facial similarity of children", *Journal of Vision*, volume 6, number 10, 2006.

[115] H. Yan, J. Lu, W. Deng, and X. Zhou, "Discriminative Multimetric Learning for Kinship Verification", *IEEE Transactions on Information Forensics and Security*, volume 9, number 7, 2014.

[116] R. Fang, K. D. Tang, N. Snavely, and T. Chen, "Towards computational models of kinship verification", in *IEEE International Conference on Image Processing (ICIP)*, 2010, pages 1577–1580.

[117] X. Qin, X. Tan, and S. Chen, "Tri-subjects kinship verification: Understanding the core of a family", in *IAPR International Conference on Machine Vision Applications (MVA)*, 2015, pages 580–583.

[118] J. Zhang, S. Xia, H. Pan, and A. K. Qin, "A genetics-motivated unsupervised model for tri-subject kinship verification", in *IEEE International Conference on Image Processing (ICIP)*, 2016, pages 2916–2920.

[119] J. P. Robinson, M. Shao, Y. Wu, H. Liu, T. Gillis, and Y. Fu, "Visual Kinship Recognition of Families in the Wild", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 40, number 11, pages 2624–2637, 2018.

[120] R. F. Rachmadi, I. K. E. Purnama, S. M. S. Nugroho, and Y. K. Suprapto, "Family-aware convolutional neural network for image-based kinship verification", *International Journal of Intelligent Engineering and Systems*, volume 13, number 6, pages 20–30, 2020.

[121] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering", in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[122] L. Dulčić, "Face Recognition with FaceNet and MTCNN - Ars Futura", Accessed: 2022-10-21. [Online]. Available: https://arsfutura.com/magazine/face-recognition-with-facenet-and-mtcnn/.

[123] R. Fang, K. Tang, N. Snavely, and T. Chen, "Towards computational models of kinship verification", in *IEEE International Conference on Image Processing*, 2010, pages 1577–1580.

[124] G. Guo and X. Wang, "Kinship measurement on salient facial features", *IEEE Transactions on Instrumentation and Measurement*, volume 61, number 8, 2012.

[125] J. Liang, Q. Hu, C. Dang, and W. Zuo, "Weighted graph embedding-based metric learning for kinship verification", *IEEE Transactions on Image Processing*, volume 28, number 3, pages 1149–1162, 2019.

[126] R. Fang, A. C. Gallagher, T. Chen, and A. Loui, "Kinship classification by modeling facial feature heredity", in *IEEE International Conference on Image Processing*, 2013, pages 2983–2987.

[127] T. H. Nguyen, H. H. Nguyen, and H. Dao, "Recognizing families through images with pretrained encoder", 2020.

[128] M. Xu and Y. Shang, "Kinship Verification Using Facial Images by Robust Similarity Learning", *Mathematical Problems in Engineering*, pages 1–8, 2016.

[129] "The Closed Eyes in the Wild (CEW) dataset", Accessed: 2022-10-21. [Online]. Available: http://parnec.nuaa.edu.cn/_upload/tpl/02/db/731/template731/pages/xtan/TSKinFace.html.

[130] Seldon, "Transfer learning for machine learning", Accessed: 2022-10-21, Jun. 2021. [Online]. Available: https://www.seldon.io/transfer-learning/.

[131] S. Ronaghan, "The Mathematics of Decision Trees, Random Forest and Feature Importance in Scikit-learn and Spark", Accessed: 2022-10-21. [Online]. Available: https://towardsdatascience.com/the-mathematics-of-decision-trees-random-forest-and-feature-importance-in-scikit-learn-and-spark-f2861df67e3.

[132] "Advantages of a Decision Tree for Classification - Python", Accessed: 2022-10-21. [Online]. Available: https://pythonprogramminglanguage.com/what-are-the-advantages-of-using-a-decision-tree-for-classification/.

[133] S. M. Piryonesi and T. E. El-Diraby, "Role of Data Analytics in Infrastructure Asset Management: Overcoming Data Size and Quality Problems", *Journal of Transportation Engineering, Part B: Pavements*, volume 146, number 2, 2020.

[134] H. Ishwaran, "The effect of splitting on random forests", *Springer*, volume 99, number 1, pages 75–118, 2015.

[135] S. Nembrini, I. R. König, and M. Wright, "The revival of the Gini importance?" *Bioinformatics*, volume 34, number 21, pages 3711–3718, 2018.

[136] N. Liberman, "Decision Trees and Random Forests", Accessed: 2022-10-21. [Online]. Available: https://towardsdatascience.com/decision-trees-and-random-forests-df0c3123f991.

[137] "4.2. Permutation feature importance — scikit-learn 1.0.1 documentation", Accessed: 2022-10-21. [Online]. Available: https://scikit-learn.org/stable/modules/permutation_importance.html.

[138] "Naive Bayes Classifier", Accessed: 2022-10-21. [Online]. Available: https://www.simplilearn.com/tutorials/machine-learning-tutorial/naive-bayes-classifier.

[139] "Naive Bayes Classifier: Pros & Cons, Applications & Types Explained", Accessed: 2022-10-21. [Online]. Available: https://www.upgrad.com/blog/naive-bayes-classifier/.

[140] A. Bakharia, "Visualising Top Features in Linear SVM with Scikit Learn and Matplotlib", Accessed: 2022-10-21. [Online]. Available: https://aneesha.medium.com/visualising-top-features-in-linear-svm-with-scikit-learn-and-matplotlib-3454ab18a14d.

[141] "SVM: Advantages Disadvantages and Applications", Accessed: 2022-10-21. [Online]. Available: https://statinfer.com/204-6-8-svm-advantages-disadvantages-applications/.

[142] "Advantages of Support Vector Machines (SVM)", Accessed: 2022-10-21. [Online]. Available: https://iq.opengenus.org/advantages-of-svm/.

[143] J. Cervantes, X. Li, W. Yu, and K. Li, "Support vector machine classification for large data sets via minimum enclosing ball clustering", *Neurocomputing*, volume 71, number 4, pages 611–619, 2008.

[144] "scikit-learn 1.0 documentation", Accessed: 2022-10-21. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.

[145] "Advantages and Disadvantages of Logistic Regression", Accessed: 2022-10-21. [Online]. Available: https://iq.opengenus.org/advantages-and-disadvantages-of-logistic-regression/.

[146] "Advantages and Disadvantages of Logistic Regression - GeeksforGeeks", Accessed: 2022-10-21. [Online]. Available: https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/.

[147] Z. Zhao, L. Alzubaidi, J. Zhang, Y. Duan, and Y. Gu, "A Comparison Review of Transfer Learning and Self-Supervised Learning: Definitions, Applications, Advantages and Limitations", *Expert Systems with Applications*, volume 242, page 122 807, May 2024. DOI: 10.1016/j.eswa.2023.122807.

[148] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations", Jul. 2020. DOI: 10.48550/arXiv.2002.05709.

[149] L. M. DeBruine, F. G. Smith, B. C. Jones, S. Craig Roberts, M. Petrie, and T. D. Spector, "Kin Recognition Signals in Adult Faces", *Vision Research*, volume 49, pages 38–43, Jan. 2009. DOI: 10.1016/j.visres.2008.09.025.

[150] G. Kaminski, S. Dridi, C. Graff, and E. Gentaz, "Human Ability to Detect Kinship in Strangers' Faces: Effects of the Degree of Relatedness", *Proceedings of the Royal Society B: Biological Sciences*, volume 276, pages 3193–3200, Sep. 2009. DOI: 10.1098/rspb.2009.0677.

[151] R. L. Burch and G. G. Gallup, "Perceptions of Paternal Resemblance Predict Family Violence", *Evolution and Human Behavior*, volume 21, pages 429–435, Nov. 2000. DOI: 10.1016/S1090-5138(00)00056-8.

[152] L. A. Zebrowitz and J. M. Montepare, "Social Psychological Face Perception: Why Appearance Matters", *Social and Personality Psychology Compass*, volume 2, pages 1497–1517, May 2008. DOI: 10.1111/j.1751-9004.2008.00109.x.

[153] M. F. Dal Martello and L. T. Maloney, "Where Are Kin Recognition Signals in the Human Face?" *Journal of Vision*, volume 6, page 2, Nov. 2006. DOI: 10.1167/6.12.2.

[154] P. Gao et al., "What Will Your Child Look Like? DNA-Net: Age and Gender Aware Kin Face Synthesizer", Nov. 2019. DOI: 10.48550/arXiv.1911.07014.

[155] M. Bordallo Lopez, A. Hadid, E. Boutellaa, J. Goncalves, V. Kostakos, and S. Hosio, "Kinship Verification from Facial Images and Videos: Human versus Machine", *Machine Vision and Applications*, volume 29, pages 873–890, Jul. 2018. DOI: 10.1007/s00138-018-0943-x.

[156] R. Fang, K. D. Tang, N. Snavely, and T. Chen, "Towards Computational Models of Kinship Verification", in *2010 IEEE International Conference on Image Processing*, Sep. 2010, pages 1577–1580. DOI: 10.1109/ICIP.2010.5652590.

[157] G. Guo and X. Wang, "Kinship Measurement on Salient Facial Features", *IEEE Transactions on Instrumentation and Measurement*, volume 61, pages 2322–2325, Aug. 2012. DOI: 10.1109/TIM.2012.2187468.

[158] S. Xia, M. Shao, and Y. Fu, "Toward Kinship Verification Using Visual Attributes", in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, Nov. 2012, pages 549–552.

[159] M. Kostinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, "Large Scale Metric Learning from Equivalence Constraints", in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI: IEEE, Jun. 2012, pages 2288–2295. DOI: 10.1109/CVPR.2012.6247939.

[160] J. Lu, X. Zhou, Y.-P. Tan, Y. Shang, and J. Zhou, "Neighborhood Repulsed Metric Learning for Kinship Verification", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 36, pages 331–345, Feb. 2014. DOI: 10.1109/TPAMI.2013.134.

[161] H. Yan, J. Lu, and X. Zhou, "Prototype-Based Discriminative Feature Learning for Kinship Verification", *IEEE Transactions on Cybernetics*, volume 45, pages 2535–2545, Nov. 2015. DOI: 10.1109/TCYB.2014.2376934.

[162] H. Yan, J. Lu, W. Deng, and X. Zhou, "Discriminative Multimetric Learning for Kinship Verification", *IEEE Transactions on Information Forensics and Security*, volume 9, pages 1169–1178, Jul. 2014. DOI: 10.1109/TIFS.2014.2327757.

[163] X. Zhou, J. Hu, J. Lu, Y. Shang, and Y. Guan, "Kinship Verification from Facial Images under Uncontrolled Conditions", in *Proceedings of the 19th ACM International Conference on Multimedia*, Scottsdale Arizona USA: ACM, Nov. 2011, pages 953–956. DOI: 10.1145/2072298.2071911.

[164] M. Shao, S. Xia, and Y. Fu, "Genealogical Face Recognition Based on UB KinFace Database", in *CVPR 2011 WORKSHOPS*, Jun. 2011, pages 60–65. DOI: 10.1109/CVPRW.2011.5981801.

[165] S. Xia, M. Shao, J. Luo, and Y. Fu, "Understanding Kin Relationships in a Photo", *IEEE Transactions on Multimedia*, volume 14, pages 1046–1056, Aug. 2012. DOI: 10.1109/TMM.2012.2187436.

[166] H. Dibeklioglu, A. A. Salah, and T. Gevers, "Like Father, Like Son: Facial Expression Dynamics for Kinship Verification", in *2013 IEEE International Conference on Computer Vision*, Sydney, Australia: IEEE, Dec. 2013, pages 1497–1504. DOI: 10.1109/ICCV.2013.189.

[167] K. Zhang, Y. Huang, C. Song, H. Wu, and L. Wang, "Kinship Verification with Deep Convolutional Neural Networks", in *Procedings of the Brit-*

ish Machine Vision Conference 2015, Swansea: British Machine Vision Association, 2015, pages 148.1–148.12. DOI: 10.5244/C.29.148.

[168] L. Li, X. Feng, X. Wu, Z. Xia, and A. Hadid, "Kinship Verification from Faces via Similarity Metric Based Convolutional Neural Network", in *Image Analysis and Recognition*, Cham: Springer International Publishing, 2016, pages 539–548. DOI: 10.1007/978-3-319-41501-7_60.

[169] J. Liang, Q. Hu, C. Dang, and W. Zuo, "Weighted Graph Embedding-Based Metric Learning for Kinship Verification", *IEEE Transactions on Image Processing*, volume 28, pages 1149–1162, Mar. 2019. DOI: 10.1109/TIP.2018.2875346.

[170] W. Li, J. Lu, A. Wuerkaixi, J. Feng, and J. Zhou, "Reasoning Graph Networks for Kinship Verification: From Star-Shaped to Hierarchical", *IEEE Transactions on Image Processing*, volume 30, pages 4947–4961, 2021. DOI: 10.1109/TIP.2021.3077111.

[171] H. Yan and S. Wang, "Learning Part-Aware Attention Networks for Kinship Verification", *Pattern Recognition Letters*, volume 128, pages 169–175, Dec. 2019. DOI: 10.1016/j.patrec.2019.08.023.

[172] F. Liu, Z. Li, W. Yang, and F. Xu, "Age-Invariant Adversarial Feature Learning for Kinship Verification", *Mathematics*, volume 10, page 480, Jan. 2022. DOI: 10.3390/math10030480.

[173] S. Wang, Z. Ding, and Y. Fu, "Cross-Generation Kinship Verification with Sparse Discriminative Metric", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 41, pages 2783–2790, Nov. 2019. DOI: 10.1109/TPAMI.2018.2861871.

[174] L. Zhang, Q. Duan, D. Zhang, W. Jia, and X. Wang, "AdvKin: Adversarial Convolutional Network for Kinship Verification", *IEEE Transactions on Cybernetics*, volume 51, pages 5883–5896, Dec. 2021. DOI: 10.1109/TCYB.2019.2959403.

[175] E. O. Belabbaci et al., "High-Order Knowledge-Based Discriminant Features for Kinship Verification", *Pattern Recognition Letters*, volume 175, pages 30–37, Nov. 2023. DOI: 10.1016/j.patrec.2023.09.008.

[176] O. Laiadi, A. Ouamane, A. Benakcha, A. Taleb-Ahmed, and A. Hadid, "Tensor Cross-View Quadratic Discriminant Analysis for Kinship Verification in the Wild", *Neurocomputing*, volume 377, pages 286–300, Feb. 2020. DOI: 10.1016/j.neucom.2019.10.055.

[177] F. Ramazankhani, M. Yazdian-Dehkord, and M. Rezaeian, "Feature Fusion and NRML Metric Learning for Facial Kinship Verification", *JUCS - Journal of Universal Computer Science*, volume 29, pages 326–348, Apr. 2023. DOI: 10.3897/jucs.89254.

[178] I. Serraoui, O. Laiadi, A. Ouamane, F. Dornaika, and A. Taleb-Ahmed, "Knowledge-Based Tensor Subspace Analysis System for Kinship Verification", *Neural Networks*, volume 151, pages 222–237, Jul. 2022. DOI: 10.1016/j.neunet.2022.03.020.

[179] Z. Wang, J. Chen, and J. Hu, "Multi-View Cosine Similarity Learning with Application to Face Verification", *Mathematics*, volume 10, page 1800, Jan. 2022. DOI: 10.3390/math10111800.

[180] H. Yan and C. Song, "Multi-Scale Deep Relational Reasoning for Facial Kinship Verification", *Pattern Recognition*, volume 110, page 107 541, Feb. 2021. DOI: 10.1016/j.patcog.2020.107541.

[181] E. Boutellaa, M. B. López, S. Ait-Aoudia, X. Feng, and A. Hadid, "Kinship Verification from Videos Using Spatio-Temporal Texture Features and Deep Learning", Aug. 2017. DOI: 10.48550/arXiv.1708.04069.

[182] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep Face Recognition", in *Procedings of the British Machine Vision Conference 2015*, Swansea: British Machine Vision Association, 2015, pages 41.1–41.12. DOI: 10.5244/C.29.41.

[183] H. Dibeklioğlu, A. A. Salah, and T. Gevers, "Are You Really Smiling at Me? Spontaneous versus Posed Enjoyment Smiles", in *Computer Vision – ECCV 2012*, volume 7574, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pages 525–538. DOI: 10.1007/978-3-642-33712-3_38.

[184] N. Kohli, D. Yadav, M. Vatsa, R. Singh, and A. Noore, "Supervised Mixed Norm Autoencoder for Kinship Verification in Unconstrained Videos", *IEEE Transactions on Image Processing*, volume 28, pages 1329–1341, Mar. 2019. DOI: 10.1109/TIP.2018.2840880.

[185] W. Wang, S. You, S. Karaoglu, and T. Gevers, "A Survey on Kinship Verification", *Neurocomputing*, volume 525, pages 1–28, Mar. 2023. DOI: 10.1016/j.neucom.2022.12.031.

[186] M. Iman, H. R. Arabnia, and K. Rasheed, "A Review of Deep Transfer Learning and Recent Advancements", *Technologies*, volume 11, page 40, Apr. 2023. DOI: 10.3390/technologies11020040.

[187] T. Mensink, J. Uijlings, A. Kuznetsova, M. Gygli, and V. Ferrari, "Factors of Influence for Transfer Learning across Diverse Appearance Domains and Task Types", Nov. 2021. DOI: 10.48550/arXiv.2103.13318.

[188] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "VGGFace2: A dataset for recognising faces across pose and age", 2018. [Online]. Available: https://arxiv.org/abs/1710.08092.

[189] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning Face Representation from Scratch", 2014. [Online]. Available: https://arxiv.org/abs/1411.7923.

[190] G. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled Faces in the Wild: A Database forStudying Face Recognition in Unconstrained Environments", *Tech. rep.*, Oct. 2008.

[191] B. van Leeuwen, A. Gansekoele, J. Pries, E. van de Bijl, and J. Klein, "Explainable kinship: The importance of facial features in kinship recognition", in *IARIA Congress 2022: The 2022 IARIA Annual Congress on Frontiers in Science, Technology, Services, and Applications*, Jul. 2022, pages 54–60.

[192] L. Ericsson, H. Gouk, C. C. Loy, and T. M. Hospedales, "Self-Supervised Representation Learning: Introduction, Advances, and Challenges", *IEEE Signal Processing Magazine*, volume 39, pages 42–62, May 2022. DOI: 10.1109/MSP.2021.3134634.

[193] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum Contrast for Unsupervised Visual Representation Learning", Mar. 2020. DOI: 10.48550/arXiv.1911.05722.

[194] Z. Cai et al., "MARLIN: Masked Autoencoder for Facial Video Representation LearnINg", in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, BC, Canada: IEEE, Jun. 2023, pages 1493–1504. DOI: 10.1109/CVPR52729.2023.00150.

[195] H. Dibeklioglu, "Visual Transformation Aided Contrastive Learning for Video-Based Kinship Verification", in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice: IEEE, Oct. 2017, pages 2478–2487. DOI: 10.1109/ICCV.2017.269.

[196] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement", 2018. [Online]. Available: https://arxiv.org/abs/1804.02767.

[197] X. Wu et al., "Facial Kinship Verification: A Comprehensive Review and Outlook", *International Journal of Computer Vision*, volume 130, pages 1494–1525, Jun. 2022. DOI: 10.1007/s11263-022-01605-9.

[198] J. Deng, J. Guo, J. Yang, N. Xue, I. Kotsia, and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 44, number 10, pages 5962–5979, Oct. 2022. DOI: 10.1109/tpami.2021.3087709.

[199] W. Lior, H. Tal, and M. Itay, "Face recognition in unconstrained videos with matched background similarity", in *CVPR 2011*, 2011, pages 529–534. DOI: 10.1109/CVPR.2011.5995566.

[200] S. J. Pan and Q. Yang, "A Survey on Transfer Learning", *IEEE Transactions on Knowledge and Data Engineering*, volume 22, pages 1345–1359, Oct. 2010. DOI: 10.1109/TKDE.2009.191.

[201] Z. Tong, Y. Song, J. Wang, and L. Wang, "VideoMAE: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training", in *Advances in Neural Information Processing Systems*, 2022.

[202] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", 2015. [Online]. Available: https://arxiv.org/abs/1512.03385.

[203] ResearchGate, "Structure of a ResNet block: Summing outputs from previous layers with the outputs of residual functions", Deep learning based image processing approaches for image deblurring - Scientific Figure on ResearchGate, [Online; accessed 16-Sept-2025], 2025. [Online]. Available: https://www.researchgate.net/figure/Structure-of-a-ResNet-block-Summing-outputs-from-previous-layers-with-the-outputs-of_fig2_347996133.

[204] M. Kim, A. K. Jain, and X. Liu, "AdaFace: Quality Adaptive Margin for Face Recognition", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[205] D. Li and X. Jiang, "Kinship Verification Method of Face Image Deep Feature Fusion", *Academic Journal of Science and Technology*, volume 5, pages 57–62, Feb. 2023. DOI: 10.54097/ajst.v5i1.5348.

[206] K. Gupta, T. Ajanthan, A. van den Hengel, and S. Gould, "Understanding and Improving the Role of Projection Head in Self-Supervised Learning", 2022. [Online]. Available: https://arxiv.org/abs/2212.11491.

[207] X. Zhang, X. Min, X. Zhou, and G. Guo, "Supervised contrastive learning for facial kinship recognition", in *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*, IEEE, 2021, pages 01–05.

[208] N. Bendib, "Supervised Contrastive Learning and Feature Fusion for Improved Kinship Verification", 2023. [Online]. Available: https://arxiv.org/abs/2302.09556.

[209] K. Sohn, "Improved Deep Metric Learning with Multi-class N-pair Loss Objective", in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., volume 29, Curran Associates, Inc., 2016.

[210] T. Hempel, A. A. Abdelrahman, and A. Al-Hamadi, "6d Rotation Representation For Unconstrained Head Pose Estimation", in *2022 IEEE International Conference on Image Processing (ICIP)*, 2022, pages 2496–2500. DOI: 10.1109/ICIP46576.2022.9897219.

[211] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-Reference Image Quality Assessment in the Spatial Domain", *IEEE Transactions on Image Processing*, volume 21, number 12, pages 4695–4708, 2012. DOI: 10.1109/TIP.2012.2214050.

[212] "Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act) (Text with EEA relevance)", Jul. 2024. [Online]. Available: http://data.europa.eu/eli/reg/2024/1689/oj.

[213] "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance)", May 2016. [Online]. Available: http://data.europa.eu/eli/reg/2016/679/oj.

Summary

This dissertation examines relational inference in real-world machine learning applications across three domains: maritime security, authorship verification (AV), and kinship recognition (KR). Despite differences in data modality and application context, these domains share common challenges that arise in practical deployment, including limited supervision, distribution shift, and heterogeneous real-world data. These conditions are characteristic of settings where observations are noisy or collected under uncontrolled conditions. To address these challenges, this dissertation adopts a multi-domain approach in which relational patterns are studied through behavioral trajectories, textual representations, and facial imagery. The focus is on how meaningful relationships can be extracted under realistic constraints.

Chapter Summaries

Chapter 2 introduces a supervised sequence learning approach for detecting maritime drug trafficking events based on AIS vessel trajectories. By modeling temporal dependencies between vessels using a Long Short-Term Memory (LSTM) architecture, coordinated "drop-off" behavior is learned as a predictive pattern rather than treated solely as an anomaly. The model achieves strong recall, making it suitable for surveillance scenarios where early detection is important. Precision limitations reflect the difficulty of distinguishing illicit coordination from complex but legitimate maritime interactions. The chapter establishes supervised temporal modeling as a method for maritime security analytics.

Chapter 3 investigates AV through the combination of deep semantic embeddings and stylometric features. Three neural architectures are evaluated: Feature Interaction, Pairwise Concatenation, and Siamese networks. All models integrate

RoBERTa-based embeddings with handcrafted stylistic features such as sentence length, word frequency, and punctuation patterns. The results show consistent performance improvements when stylistic features are included, particularly in recall. The best-performing Siamese model demonstrates that relatively simple architectures can be effective when combined with complementary feature types. The chapter focuses on multi-perspective representations combining semantic and stylistic information.

Chapter 4 evaluates model performance under domain shift. Two evaluation settings are considered: a non-holdout setting with partial domain overlap and a holdout setting with unseen domains. Results show that performance remains relatively stable under moderate shifts but decreases in structurally different domains such as handwritten text. Despite this, the proposed models outperform strong baselines from the PAN shared task. The study uses multiple writing domains, including essays, emails, text messages, speech-to-text data, search engine queries, and handwritten text, to evaluate robustness across modalities and writing styles.

Chapter 5 investigates facial attributes relevant for KR using interpretable models based on 40 StyleGAN2-derived features. The analysis shows that age-related cues and facial hair characteristics are the most influential features in classification. Localized facial regions contribute less than expected in comparison. These findings are derived from experiments on KR datasets using feature-based classification models, and they provide a breakdown of which visual attributes are most associated with model predictions.

Chapter 6 evaluates pretrained deep learning models and contrastive learning approaches for KR. Two methods are compared: KinFusionNet, a Siamese-style architecture, and FLACL, a supervised contrastive learning framework. Both image-based and video-based inputs are used. Results show that image-based representations consistently outperform video-based ones. The experiments further show that backbone choice influences performance, with models pretrained on identity-focused datasets transferring more effectively than those trained on video reconstruction tasks. The study highlights the role of representation learning in KR under limited supervision.

Dissertation Scope

The dissertation integrates three main strands of contribution. First, it provides empirical analyses of relational inference across maritime behavior analysis, AV, and KR, demonstrating how different types of relational structure can be modeled using domain-specific representations. This reflects that relational inference is

possible across different domains, but that the appropriate representation strongly depends on the type of relation and the data modality.

Second, it systematically evaluates the role of representation design, including sequential modeling, semantic and stylistic feature combinations, and pretrained visual embeddings, across multiple tasks and architectures. The results suggest that combining different sources of information can improve robustness, although this effect depends on whether the signals provide genuinely complementary rather than redundant or noisy information.

Third, it includes comparative evaluations of model robustness under changing conditions, including domain shift in AV and modality variation in KR, highlighting that generalization is the central challenge for practical deployment and that similar issues are also relevant for maritime applications.

Across all contributions, the models should primarily be interpreted as decision-support tools rather than autonomous decision-making systems, which aligns with the ethical considerations.

List of Abbreviations

AA	Authorship Attribution
AC	Authorship Classification
AIS	Automatic Identification System
AUC	Area Under the Curve
AUC-PR	Area Under the Precision-Recall Curve
AUROC	Area Under the Receiver Operating Characteristic Curve
AV	Authorship Verification
BBC	British Broadcasting Corporation
BCE	Binary Cross-Entropy
BERT	Bidirectional Encoder Representations from Transformers
BoW	Bag-of-Words
CNN	Convolutional Neural Network
COG	Course Over Ground
DD	Dutch Draw (Baseline Model)
DT	Decision Tree
F1	F1 Score

FDR	False Discovery Rate
FI	Feature Importance
FIN	Feature Interaction Network
FIW	Families in the Wild
FKR	Facial Kinship Recognition
FLACL	Family Label Aware Contrastive Learning
FM	Fowlkes-Mallows Index
FN	(Number of) False Negative(s)
FOR	False Omission Rate
FP	(Number of) False Positive(s)
GAN	Generative Adversarial Network
GNB	Gaussian Naive Bayes
HDG	Heading
IMO	International Maritime Organisation
KAN-AV	Kinship Audio-Visual Dataset
KNN	K-Nearest Neighbors
KR	Kinship Recognition
LDA	Latent Dirichlet Allocation
LFW	Labeled Faces in the Wild
LODO	Leave-one-domain-out
LR	Logistic Regression
LSTM	Long Short-Term Memory
LSVM	Linear Support Vector Machine
MCC	Matthews Correlation Coefficient
ML	Machine Learning
MLP	Multi-Layer Perceptron
MMSI	Maritime Mobile Service Identity

NLP	Natural Language Processing
NPV	Negative Predictive Value
NT-Xent	Normalized Temperature-scaled Cross-Entropy Loss
PAN	Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection
PCN	Pairwise Concatenation Network
PPV	Positive Predictive Value
PR	Precision-Recall
ReLU	Rectified Linear Unit
RF	Random Forest
RGB	Red Green Blue (color representation)
RGB	Red Green Blue
RNN	Recurrent Neural Network
RoBERTa	Robustly Optimized BERT Pretraining Approach
ROC	Receiver Operating Characteristic
ROC-PR	Receiver Operating Characteristic - Precision Recall
ROT	Rate of Turn
SHAP	SHapley Additive exPlanations
SMBO	Sequential Model-Based Optimization
SN	Siamese Network
SOG	Speed Over Ground
SSL	Self-Supervised Learning
StyleGAN	Style-based Generative Adversarial Network
StyleGAN2	Style Generative Adversarial Network 2
SVM	Support Vector Machine
T5	Text-To-Text Transfer Transformer
TF-IDF	Term Frequency - Inverse Document Frequency

TMP	The Maritime Police
TN	(Number of) True Negative(s)
TP	(Number of) True Positive(s)
TPR	True Positive Rate
TSKinFace	Tri-Subjects Kinship Face Database
UTC	Coordinated Universal Time
VRNN	Variational Recurrent Neural Network
WBCE	Weighted Binary Cross-Entropy
YTF	YouTube Faces Dataset

Dankwoord

Begin september 2020 zag ik een vacature langskomen: een PhD-positie met als titel "Machine Learning for Combatting Cybercrime". Ik was natuurlijk niet op zoek naar een PhD-positie, mijn doel was eerst mijn master binnen te halen. Toch geïntrigeerd door het onderwerp stuurde ik een mailtje: "Het ziet er erg interessant uit, maar ik moet eerst nog een jaartje studeren om mijn master te halen. In de PhD-positie zelf heb ik dus geen interesse, maar ik ben wel op zoek naar een project voor mijn masterthesis."En zodoende belandde ik in een team begeleid door Rob van de Mei en Sandjai Bhulai, bestaande uit Jan, Joris, Etienne en Arwin. De stage ging snel voorbij en aan het einde kreeg ik een aanbod: of ik bij het CWI wilde blijven voor een PhD. Wie mij goed kent die weet: als er voor mij een deur opent met een uitdaging erachter, dan ga ik die graag aan.

Vier (en een half) jaar later zit ik hier, kijkend naar mijn proefschrift dat af is. Na al het harde werk is daar eindelijk het resultaat. Natuurlijk had ik dit nooit kunnen bereiken zonder de begeleiding en steun van iedereen om mij heen.

Als allereerste wil ik dan ook mijn promotoren bedanken, Rob en Sandjai. Het wekelijkse overleg was vaak erg gezellig. Meetings van een half uur begonnen meestal met twintig minuten bijpraten over de afgelopen week, waarna pas de inhoudelijke update volgde. Er waren ups en downs, zoals te verwachten is tijdens een PhD-traject, maar met jullie hulp heb ik mij door alle obstakels heen geslagen. Jullie bleven altijd positief en vol vertrouwen. Kortom hebben jullie mij tijdens mijn PhD-traject erg fijn begeleid, dank daarvoor!

Daarnaast wil ik ook graag mijn collega's bedanken. Van begeleiders naar collega's naar padelmaatjes — jullie zijn mij voorgegaan en inmiddels mogen jullie jezelf allemaal doctor noemen; ik ben blij dat ik jullie mag volgen. Arwin, Etienne, Jan en Joris, erg bedankt voor de begeleiding, de gezelligheid en de aanmoediging.

Verder wil ik al mijn andere collega's van het CWI en de VU bedanken voor de gezelligheid op kantoor. Ik was er niet heel vaak, maar als ik er was, was het altijd fijn. Ik kijk met plezier terug op de conferentietripjes naar Nice en Kopenhagen. Samen naar presentaties luisteren, maar ook tijd nemen om naar het strand te gaan, te dansen en karaoke te zingen. Ook de CWI-uitjes zijn natuurlijk niet te vergeten. Bedankt allemaal voor de fijne tijd.

Ook buiten het werk heb ik veel steun ontvangen. Ik ben erg dankbaar voor de onvoorwaardelijke steun van mijn ouders, Piet en Debbie, bij het kiezen van mijn eigen pad. Jullie hebben altijd vertrouwen in mij uitgestraald en mij gemotiveerd om in mijzelf te geloven en door te zetten.

Ik wil ook mijn partner, Janus, bedanken voor het altijd achter mij staan. We delen veel mooie momenten, maar juist ook bij tegenslagen sta je altijd voor mij klaar. Ik kan op je rekenen om mij weer beter te laten voelen als ik er even doorheen zit. Daarnaast zorg je er ook voor dat ik juist de mooie en goede momenten vier, en dat we even stilstaan bij wat ik heb bereikt. Ik heb veel van je geleerd en ben je erg dankbaar.

Ook heb ik ook heel veel gehad aan mijn vrienden. Bedankt voor alle leuke dingen die we de afgelopen vier jaar hebben gedaan; zonder er even tussenuit te kunnen, had ik het niet volgehouden. Elke keer weer die vraag: "Hoe gaat het met je PhD?" met mijn standaardantwoord: "Niet nu, het is mijn vrije avond." Ondanks dat jullie daardoor soms geen idee hadden waar ik mee bezig was, twijfelden jullie geen moment aan of ik het kon halen. Specifiek wil ik Leen en Sofia bedanken, mijn paranimfen. Ik ben erg dankbaar voor onze vriendschappen.

Er zijn nog zoveel meer mensen die ik zou kunnen opnoemen. In de afgelopen vier jaar heb ik alleen maar aanmoediging en vertrouwen gevoeld van iedereen om mij heen. Zeker wanneer ik zelf twijfelde aan mijn kennis en kunnen, is jullie steun voor mij van onschatbare waarde geweest. Ik hoop dat jullie van mijn proefschrift kunnen genieten.

About the Author

Britt was born on 26 September 1997 in Velserbroek, the Netherlands. She developed a strong interest in mathematics during her secondary education in Driehuis, which led her to pursue a Bachelor's degree in Mathematics at Vrije Universiteit Amsterdam. During her bachelor's programme, Britt completed a minor in programming, where she discovered her enthusiasm for computer science and problem-solving.

Britt subsequently enrolled in the Master's programme in Artificial Intelligence, where she further developed her interest in combining theoretical foundations with practical applications. During her master's studies, Britt completed an internship at the Centrum Wiskunde & Informatica (CWI). This experience ultimately led to the opportunity to pursue a PhD, which she enthusiastically accepted.

Throughout her academic career, Britt has been actively involved in extracurricular activities. These included working as a teaching assistant for the Bachelor's programme in Artificial Intelligence and serving as chair of her study association. While Britt has a strong affinity for technology, she also values collaboration and working with people. During her PhD, she found a balance between conducting research and contributing to the academic community through activities such as attending conferences, supervising master's theses, and following additional courses to further broaden her knowledge.