EXPLOITING GROUP STRUCTURES TO INFER SOCIAL INTERACTIONS FROM VIDEOS

A Thesis
Submitted to the Faculty
in partial fulfillment of the requirements for the
degree of

Doctor of Philosophy

in

Computer Science

by Maksim Bolonkin

Guarini School of Graduate and Advanced Studies
Dartmouth College
Hanover, New Hampshire

August, 2021

Examining Committee:

(chair) **V.S. Subrahmanian**, Dartmouth College

Bo Zhu, Dartmouth College

Soroush Vosoughi, Dartmouth College

Dimitris Metaxas, Rutgers University

F. Jon Kull, Ph.D.
Dean of the Guarini School of Graduate and Advanced Studies

Abstract

In this thesis, we consider the task of inferring the social interactions between humans by analyzing multi-modal data. Specifically, we attempt to solve some of the problems in interaction analysis, such as long-term deception detection, political deception detection, and impression prediction. In this work, we emphasize the importance of using knowledge about the group structure of the analyzed interactions. Previous works on the matter mostly neglected this aspect and analyzed a single subject at a time.

Using the new Resistance dataset, collected by our collaborators, we approach the problem of long-term deception detection by designing a class of histogram-based features and a novel class of meta-features we call LiarRank. We develop a LiarOrNot model to identify spies in Resistance videos. We achieve AUCs of over 0.70 outperforming our baselines by 3% and human judges by 12%.

For the problem of political deception, we first collect a dataset of videos and transcripts of 76 politicians from 18 countries making truthful and deceptive statements. We call it the Global Political Deception Dataset. We then show how to analyze the statements in a broader context by building a Video-Article-Topic graph. From this graph, we create a novel class of features called Deception Score that captures how controversial each topic is and how it affects the truthfulness of each statement. We show that our approach achieves 0.775 AUC outperforming competing baselines.

Finally, we use the Resistance data to solve the problem of dyadic impression prediction. Our proposed Dyadic Impression Prediction System (DIPS) contains four major innovations: a novel class of features called emotion ranks, sign imbalance features derived from signed graphs theory, a novel method to *align* the facial expressions of subjects, and finally, we propose the concept of a *multilayered* stochastic network we call Temporal Delayed Network. Our DIPS architecture beats eight baselines from the literature, yielding statistically significant improvements of 19.9-30.8% in AUC.

Preface

The work presented in this thesis was conducted in the Dartmouth Security and AI Laboratory. The research was in part funded by the US Army Research Office (ARO Grant W911NF1610342). The new Resistance dataset extensively used in this work was collected as part of the Multidisciplinary University Research Initiative by our collaborators from the University of Arizona, University of California Santa Barbara, Rutgers University, Stanford University, and the University of Maryland.

Acknowledgments

My graduate studies have been a long journey full of hesitations and tough decisions. This thesis would never be possible without all the support I received from my mentors, colleagues, friends, and family.

First and foremost, I am forever grateful to my mentor and advisor, V.S. Subrahmanian. From the first day I entered your office to discuss SCAN project and the prospect of working in your lab until the moment you delivered the news of successful defense, I felt unconditional support and encouragement. Thank you for sharing your knowledge, your massive experience, and your valuable wisdom with me. It was a great honor to be your student, and I hope to keep up with the high standards set by you.

I'm thankful to my thesis committee Soroush Vosoughi, Bo Zhu, and Dimitris Metaxas, for their valuable feedback on this work as well as multiple suggestions on possible extensions of my research.

Many thanks go to my collaborators on SCAN MURI projects: Zhe Wu, Bharat Singh, Chao Chen, Viney Regunath, Srijan Kumar, Jure Leskovec, Judee Burgoon, Norah Dunbar, Dongkai Chen, Lezi Wang, Sayak Chakrabarty, and Cristian Molinaro. I want to express separate gratitude to Chongyang Bai, with whom I worked side-by-side on most SCAN projects. Chongyang, it has been a great pleasure to work with you. I have learned a lot from our collaboration. I will also fondly remember all

DSAIL colleagues: Qian Han, Rui Liu, Tommy White, Yanhai Xiong, Bowen Dai, and Chiara Pulice.

I cannot leave out great researchers I had the luck to work with during my first years at Dartmouth: Lorenzo Torresani, Du Tran, Manohar Paluri. Lorenzo, I cannot express how much I am thankful for what I learned from you.

My time at Dartmouth would not be as beneficial if not for many great educators I had the honor to learn from: Wojciech Jarosz, Hany Farid, Prasad Jayanti, Sergei Bratus, Thomas Cormen. You taught me a lot about subjects but even more about how to be a fantastic teacher.

I owe my sanity and my successful completion to my friends who supported me through these long years in Upper Valley: Shruti Agarwal, Naofumi Tomita, Ruchir Patel, Jack Messerly. Shruti, you were always a shining beacon of hope and support. I will always miss our discussions and walks for coffee. Nao, where would I be without our occasional trips for groceries? Ruchir, it is not easy to imagine coming to the second floor of Sudikoff and not seeing you there studying. I will miss our chats about life, computer science, and politics. Jack, I am grateful to you for motivating me when I needed that and the lumbar support that stayed with me after your departure. I am also happy to have amazing friends rooting for me overseas: Linara Adilova, Anna Tuniants, Alisher Alimov. I always knew that I could talk to you when I needed that and receive emotional support or research feedback.

Special thanks to Kenneth Bernier. Kenny, you witnessed the harshest part of the whole process, and I felt your support all the time. I was lucky to have you by my side when finishing this chapter of my life, and I am looking forward to more adventures together.

Finally, and probably most importantly, my family: my mom Larisa Bolonkina, my sister Anastasia Gainutdinova, my brother-in-law Kirill Gainutdinov, and my nephew Roman Gainutdinov. I love you all, and I can't thank you enough for having faith in me when I lacked it.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Aristotle called humans "political animals"[2] implying that we are more social than any other species. Indeed, many social psychologists think that humans evolved to be social [91, 28], as that provides a lot of evolutionary benefits such as protection from predators, psychological well-being, increased mating capabilities, and others. It is not surprising that *homo sapiens* also developed ways of exploiting these interactions: we try to make good impressions to make friends, we express our displeasure to affect others, we lie to each other, we dominate or submit whenever it is beneficial for our goals.

Social interactions permeate human lives. Each day we engage in various interactions that can vary in multiple ways: in the number of people interacting (from talking one-on-one with someone to giving a talk in front of a thousand people), in the medium of interaction (e.g., face-to-face discussion in the same room, video conferences, or even text messaging), in the incentives behind the exchange (adversarial or collaborative), and others.

Naturally, humans always wanted to be able to predict other humans' behavior. These days it is not unusual to see articles in popular and social media titled "How to Tell If Someone Is Lying to You"[3] or "How to Tell If Someone Likes You"[4]. By observing many human subjects, psychologists have discovered some cues that can help uncover a person's thoughts, attitudes, inner state and potentially predict her future actions. In the age of Big Data, it is not surprising to see Machine Learning heavily influencing the area of automated behavior analysis. It has been successfully applied to deception detection [95, 158, 122], anxiety prediction [114, 59, 55], dominance and leadership

[2] Aristotle., and Trevor J. Saunders. Politics. Books I and II . Oxford: Clarendon Press, 1995. Print.

[3] https://time.com/5443204/signs-lying-body-language-experts/

[4] https://www.scienceofpeople.com/someone-likes-you/

analysis [131, 131, 10, 5, 105, 106], personality traits prediction [51, 16, 107], and other areas of computational social science.

This research has some potential real-life applications. Deception detection is already used in law enforcement [57] but has potential use in the military, intelligence agencies, and even in commercial organizations. There are numerous use cases for this task: from interrogating potential terrorists and verifying the good intentions of visa applicants to assessing the integrity of business partners or even questioning a cheating partner. Of course, improvements in deception detection accuracy raise several moral and ethical concerns. We discuss them in Chapter 6.

Impression prediction and social dynamics analysis can be used to understand the interaction and attitudes within a group of people (e.g., a project team or sports team), which can help improve a group's performance [108, 47, 90]. Another area of interest is automated job interview assessment [69, 102] where video recordings of one-on-one interviews are used to predict whether the candidate will get hired. This task is helpful to reduce human bias during the hiring process and improve hiring managers' skills. We can also apply the learned properties in a generative setting: building more human-like virtual assistants possessing artificial emotional intelligence and exhibiting realistic group behavior.

As we have mentioned above, computational social science received a lot of attention in the past decade. Nevertheless, these tasks remain challenging for several reasons.

First, Machine Learning and especially Deep Learning approaches are hungry for data, but the data is lacking. Existing datasets for deception detection are either restricted in modalities (text-only [116] or audio-only [101]), or relatively small in size (on the order of 100–200 samples or less) [122, 52, 63]. This problem arises because it is (1) hard to define precisely what deception is, (2) expensive to run human experiments, (3) hard to find publicly available data with low noise levels and existing reliable annotations. The same challenges exist for other social interaction datasets.

Second, because of the lack of data, researchers cannot efficiently use contemporary computational techniques (e.g., Deep Learning) to learn useful representations automatically. They have to refer to psychological and social sciences for inspiration. The majority of works on deception detection and interaction analysis use features previously discovered by psychologists to be relevant for the task at hand. These features include prosodic and acoustic properties such as pitch [101, 173], speaking activity features such as the number of turns or utterance rate [31, 15], statistical

linguistic properties such as word categories frequency [134, 84], visual features such as facial expression units [158, 138, 107].

Very few works make use of the group nature of social interactions, however. In fact, very few datasets reflect this common property in the first place. Only several deception detection datasets depict group deception [162, 39, 31] despite the latter being very common in real-life (from international summits to business negotiations). Yu et al. [162] use the network properties of the group to detect a cluster of liars. Most recent works by Kumar et al. [86] and Wang et al. [155] use gaze network properties to predict multiple social features such as deception, dominance, and others. But most of the other papers on deception detection use only the potential liar's individual features to predict dishonesty. In the area of personality traits prediction and interpersonal attitudes analysis, there are several prominent group interaction datasets [103, 102], possibly the most widely used is ELEA [130]. But most of the works in this area still make predictions based on single subject's data.

This dissertation aims to address these shortcomings by proposing ways of exploiting the network nature of social interactions and showing that this leads to improvement on some of the challenging tasks, such as deception detection and impression prediction.

1.2 Problem Statement

Given videos of people participating in social interaction (playing a game or making public statements), we want to predict hidden properties of a given person's behavior: whether the person is deceptive or what is the person's attitude towards other group members.

These problems received significant attention from the research community. Figure 1.1 shows a generic solution pipeline that usually includes the following steps: (1) separating modalities from the video of the subject, (2) extracting features from the corresponding modalities, for instance, MFCC for audio or LIWC word frequencies for text, (3) training a Machine Learning model to make a prediction for the classification task at hand. Different modalities are fused either before the model training step (usually by concatenating, but sometimes using more sophisticated techniques [105]) or after the model training step (by means of late fusion [10]).

The scope of this dissertation is to explore different ways of using the group na-

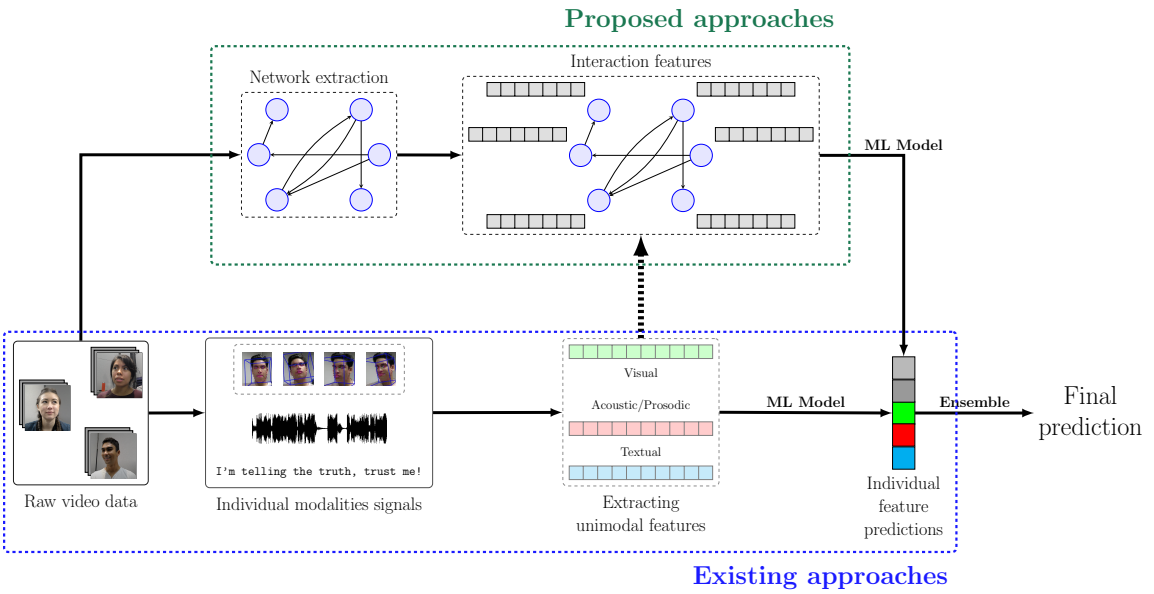


Figure 1.1: Scope of the dissertation. The majority of automated social behavior analysis works are considering each individual in the group, extracting some relevant features and making predictions about traits or actions in question. We propose to exploit the social nature of the interactions and produce features and models that take mutual influence into account.

ture of the interactions (Figure 1.1). We propose ways of building features that use group-level information for each considered task and show how that affects the final prediction accuracy. We consider the following three tasks: deception detection in group settings, deception detection in public statements, dyadic impression prediction.

1.3 Contributions

The overall contributions of this work can be summarised as follows:

- we proposed a way to exploit the group information for the task of long-term deception detection,
- we showed how to build a network for the seemingly individual task of deception detection in political and public statements and showed how we could use it to improve the performance on this task,
- to evaluate our approach on the previous task, we collected a multimodal dataset of deceptive and truthful statements by public figures,

- finally, we developed several new features and a multi-layer network approach to solve the task of dyadic impression prediction.

We briefly summarize each of these contributions below. And the rest of the document is organized as follows: in Chapter 2 we review existing work on the problems in this dissertation and highlight where we stand out; Chapter 3 discusses the problem of long-term deception detection and how using group information can improve the detection accuracy; Chapter 4 considers the task of political deception detection and proposes a network-based solution that improves on other multi- and uni-modal features; in Chapter 5 we try predicting dyadic impressions in a group setting by incorporating mutual influence in the group into the set of novel features as well as using a multi-layer network to train a Graph Convolutional Network; finally, in Chapter 6 we provide concluding remarks and suggest potential extensions to our work.

1.3.1 Long-term deception detection (Chapter 3)

In this chapter, we consider a task of deception detection in long group interaction videos. We use a novel Resistance dataset [45] containing videos of groups of subjects playing a social game involving deception (see Section 2.3). This task brings several challenges:

- Unlike most of the other datasets and works on the topic of deception detection, which consider subjects in a monologue or interview setting, we deal with a highly interactive setting. This causes several issues. For example, processing the audio becomes complicated due to multiple people speaking concurrently. Subjects in videos can move a lot, look away from camera, chat with other players, or gesture. On one hand these features provide valuable information about the subject's state of mind, on the other hand these same features make it harder to automatically analyze the videos.
- Unlike most of the other similar datasets, which contain short videos of 10–120 seconds, the Resistance dataset contains very long videos spanning 30–60 minutes.

This chapter proposes a computational method that we call LiarOrNot to predict deception in the game.

In addition to the fact that long-term deception in group settings has been rarely studied, LiarOrNot makes the following innovations. Building on well-known basic image (VGG Face) and audio features (mel-frequency cepstral coefficients),

1. we introduce a class of histogram-based features that build on well known low-level (eye/head movement, facial action units) and high-level (emotion features from Amazon Rekognition) features,
2. we introduce a novel class of "meta-features" called LiarRank that builds on the basic features, and
3. we introduce an ensemble based prediction model.

We show that LiarOrNot achieves an AUC of 0.705 in this hard test, significantly outperforming other feature classes and past work. Additionally, as the Resistance dataset was collected across three very different countries and because there may be cultural differences in deception, our results are more robust across cultures than past studies.

1.3.2 Political deception detection (Chapter 4)

This chapter deals with the following problem: having a video of a public figure (usually a politician) making a statement, we want to predict whether the statement is truthful or deceptive. Similar to other works [84, 76] we use fact-checking of the underlying fact as a proxy for the veracity of the statement: we consider it truthful if fact-checkers found it to be factually correct and deceptive otherwise.

This task has its challenges:

- First, there is an obvious lack of publicly available datasets for this task: the sole available dataset [100] contains only text and restricts itself to only US politicians,
- Second, it is obvious that the veracity of the statement highly depends on the content of that statement. Still, it is not immediately clear how to analyze the content in the broader context.

In an attempt to overcome these challenges, we make the following novel contributions:

1. We collect a new cross-cultural, multilingual dataset of public figures from several countries worldwide making truthful and deceptive statements. We call this dataset the Global Political Deception Dataset. It is the first multimodal dataset of this kind with subjects drawn from a variety of cultures and linguistic backgrounds;

2. We propose a way to build a graph with nodes representing videos of politicians, topics of the messages, and news articles about those politicians: this method allows putting the content of the analyzed statement into a broader context and assess how likely it is to be deceptive;

3. We develop a novel class of features we call Deception Score that brings together properties of the video (how likely the person in it to be deceptive) with the assessment of how likely the message from the video to be deceptive;

4. We show that our proposed Deception Score in conjunction with basic features greatly improve upon basic features alone and a comprehensive set of baselines.

1.3.3 Dyadic impression prediction (Chapter 5)

In this chapter, we consider the problem of dyadic impression prediction: given videos showing group interaction, we want to predict whether subject p_i likes or dislikes subject p_j. Specifically, we would like to use nonverbal cues such as facial action units [12], facial emotions [89], gaze relationships [11], and more in order to predict subject p_i's impression about subject p_j's likability. We capture likability through six survey questions designed to elicit p_i's impression of p_j.

In order to predict these six types of dyadic impressions, we have developed a framework called DIPS (Dyadic Impression Prediction System). DIPS involves the following features:

1. *Emotion Ranks* is a novel class of features intended to capture interpersonal emotional response throughout the video. It is well-known (e.g. [36]) in social science that the emotions of a subject p_i about a subject p_j may be influenced by the emotions of others toward p_j. We first consider emotions of a pair (p_i, p_j) and define *emotion score* as the intensity of a given emotion on p_i's face weighted by the probability that that emotion is directed toward p_j. Emotion rank takes these dyadic emotion scores as input and uses interaction networks to account

for the fact that user p_i's impression of p_j might depend not only on his facial emotions, but also that of others, as well as his attitude toward those other individuals. This leads to a mutual influence of emotions and network interactions that we aggregate. *Past work on predicting impressions [98] only considers direct gaze relationships and does not consider such network interactions.*

2. *Sign Imbalance features.* Classical social network theory has identified the importance of triangles in friend/enemy networks. Such networks are called *signed networks* [43, 50] in which edges can be positive or negative. Balance theory forms an important part of social network theory going back to the time of Heider in the 1950s [66]. It suggests that for a triad to be balanced, the products of any pair of edge weights must be positive. Important phenomena explained by balance theory include the ideas that "a friend of my friend is my friend" and "an enemy of my enemy is my friend". In this thesis we propose a novel class of features that measures the degree of *imbalance* thereby quantifying the effect of a third party p_k on the impression that p_i has of p_j. *To our knowledge, this is the first time that social balance theory has been used for predicting impressions from nonverbal data.*

3. *Emotion and Facial Action Units Alignment.* Social science theory [36] posits that p_i's impression of p_j and p_j's impression of p_i are not independent. We can observe this in our daily lives - if a person doesn't like you that might cause an unconscious response: you may not like them back. We might therefore get clues about p_i's impression of p_j by looking at p_j's facial emotions. We define a novel class of *alignment vectors* that capture the alignment — with possible temporal delays in order to account for subjects' response times — between the facial emotions and action units of subjects p_i and p_j.

4. Finally, we introduce the novel concept of a *Temporal Delayed Network* which is a *multi-layer* network [37, 83] where each layer represents the social group at a particular time point. Within a single layer, nodes correspond to players and edges correspond to different interactions between players (e.g. look at, talk to, listen to). Within a layer, edges are labeled with the probability of the stated interaction. Across layers, edges connect the same individuals in different layers, as well as track delayed interaction information. *To our knowledge, this is the first time that multi-layer networks have been used in predicting impressions of subjects.* Using this TDN as an underlying graph, we build a Graph Convolution

Network [82] with an attention mechanisms [148] to learn representations and predict dyadic impressions of p_i toward p_j.

We also build an ensemble model out of our proposed features. We evaluate our framework on the Resistance dataset and show that it outperforms baselines that use only single-subject features by a significant margin on all of the variables.

Chapter 2

Background

2.1 Automated Deception Detection

2.1.1 Psychological and physiological research

Humans noticed that an act of lying brings some changes to a liar's physiology. Ford [57] records an ancient Chinese technique of making a suspect put dry rice in the mouth and then spit it out. If the rice was still dry, the person was presumed to lie, as the feeling of guilt was associated with lack of salivating. One of the first systematic attempts at detecting lies was based on understanding human physiology and finding correlations with psychophysiological effects of lying [1]. Physiological measurement-based methods measure blood pressure, heart/respiratory rate, and galvanic skin response to detect deception. Other methods include blood flow measurements [23] or brain imaging using fMRI [70]. Results showing that deception is linked to an increase in blood flow in the eye [2] which in turn can be detected via thermal imaging, have led to new methods to identify deception in airports [113, 156, 71, 123]. When Rajoub et al. [123] performed experiments with thermal videos, the method was very successful (87% accuracy), but the same identity was part of the training and test set. However, when different identities were evaluated, the performance dropped to 60%, highlighting that deception detection on new identities is difficult. Moreover, this experimental study was done on a small scale in short videos and only involved 25 people. In contrast, our approach focuses on deception prediction in people never seen before and is evaluated on a dataset that is an order of magnitude larger. Improved results are obtained when cues such as facial expressions, subtle body movements, hesitations or pauses in speech, gaze aversion, dilation of pupils are combined with physiological features [2]. Samuel et al. [129] used electromyography (EMG) readings of the masseter muscle, along with electrocardiography and galvanic skin response, to detect deceptive behavior with relative success. Prior work also shows that there is a correlation of deception with emotions like fear, guilt, and delight [151]. The hypoth-

esis is that significant cognitive load leads to subtle changes in some of these cues, which can help in deception analysis [24, 40]. Compared to our proposed method, these methods are not automatic, and the cues are typically annotated by humans, which are later used in a classifier.

Burgoon et al. [26] demonstrated that deceitful and truthful speech could be detected based on linguistic cues, deceivers being more concise, having less rich language and less complex sentence structures, demonstrating lack of specificity. Zhou et al. [170] performed analysis of emails and found that deceivers' and truth tellers' linguistic patterns differ, and deceivers' language changes over time to adapt to the changing interaction.

2.1.2 Deception detection datasets

Several unimodal datasets for deception detection were developed over the past ten years. Cross-Cultural Deception dataset [116] contains a set of essays on controversial topics written by subjects from four countries with different culture: US, India, Mexico, and Romania. Subjects were required to write short essays not necessarily representing their own belief on the given topic. Open Domain Deception Dataset [117] also relied on crowd-sourcing: subjects from Amazon Mechanical Turk were requested to provide several casual truths and lies in the form of one sentence. This dataset also contains the demographic data about the subjects. Another dataset [146] contains Electroencephalogram (EEG) readings of subjects when they lie or tell the truth. Mihalcea et al. [96] collected a dataset of short answers to three promts written by Amazon Mechanical Turk workers: 100 truthful and 100 deceptive statements on each of the three topics. Nasril et al.[101] created a ReGIM-Lab Lie Detection DataBase where subjects participated in an interview and the answers (truthful and deceitful) were recorded. Zhang et al. [168] were among first to use fine-grained image analysis to detect deception in facial and emotional expressions in static images. To distinguish genuine facial expressions from simulated ones, they proposed a set of features relying on 58 manually labeled facial points, which makes the approach not fully automated.

Several more datasets contain two or more modalities: usually speech and text, but sometimes also video or physiological readings (EEG, ECG, etc.) Levitan et al. collected a Columbia X-Cultural Deception (CXD) Corpus of dialogs where subjects were lying or telling the truth about some casual topics [88]. This corpus explores

acoustic, prosodic, and linguistic sides of the deception. Early work in the computer vision community used head and hand tracking to predict the state of the human subject in a video (relaxed, agitated or over-controlled) to infer whether a person is deceitful [136, 143]. These methods, however, were tested on a very small database (5 videos with a total duration of 5 minutes 33 seconds), were person-specific and prone to overfitting. Michael et al. [95] proposed a feature called motion patterns, incorporating both head/hand movement and automatic facial landmarks tracking.

One of the most widely used datasets is a Real-Life Trial dataset [122] that contains videos of people telling truths or lying in a high stake situation: testifying in a court of law. This dataset comprises videos, transcripts, and manually annotated facial expressions. Two research groups attempted to create a low-stake lies dataset by using the "Box of Lies" game: one player describes an object in the box that only she can see, and another player must guess whether the first player is telling the truth or lying. Soldner et al. [52] used publicly available recordings of the "Box of Lies" game played by United States celebrities on The Tonight Show with Jimmy Fallon. This dataset contains videos of the players annotated for deception at utterance level, since the deceptive description of an object can be false in only minor details (for example, the player can change the color of the object leaving all other details truthful). Another similar dataset called "Bag of Lies" [63] contains video, audio, and eye gaze of subjects playing the same game in laboratory settings. For a subset of subjects EEG readings are also available.

In the recent years several datasets emerged for deception detection in group setting. All of them are based on a variances of a social game "Mafia" (other versions are "Werewolf" and "The Resistance"). In these games some players form a secret clique aiming at deceiving the rest of the players and staying hidden to win the game. Some of these datasets are formed out of publicly available videos of a TV-game with the same name [39], others conducted the games in controlled setting and recorded the videos [31, 45]. The Resistance dataset [45] also includes a set of surveys quantifying players feelings towards each other. Two other datasets contain transcripts of a similar game played online through text-messaging service [162] or on the online message board [38].

Apart from high stake lies (courtroom testimonies) and low stake lies (games and mock interviews), another domain of interest for automated deception detection is politics. It has been shown that the false news are spreading much faster and reach larger audience than truthful news [150]. Thus it is becoming important to be able to

automatically assess the truthfulness of the statements made by public figures such as politicians and high-level government officials.

One dataset aiming at this task is CT-CWC-18 corpus used in CLEF-2018 CheckThat! Lab Task [100]. It contains transcripts of political debates from 2016 US Presidential campaign alongside with labels of truthfulness according to fact-checking resources[1]. Another dataset [76] contains 180 videos of US politicians from both major parties making public statements that are labeled by PolitiFact[2] with one of the following verdicts: True, Mostly True, Half True, Mostly False, False, Pants on Fire. This dataset, although truly multimodal, is not publicly available at the moment of writing this thesis.

In this work we wish to contribute to the field by introducing a new dataset that we call Global Political Deception Dataset containing videos of public figures from around the world making truthful or false statements. We selected the subjects from a wide array of countries and cultures.

2.1.3 Deception detection methods

Since the first available deception detection datasets were mostly unimodal (usually text or audio), early work in this area was mostly concentrated on these modalities. Hirschberg et al. [67] used lexical (LIWC categories) and acoustic/prosodic features to distinguish deceptive speech from truthful. Mihalcea et al. [96] used similar linguistic features, in particular LIWC 2001 categories frequencies to detect deceptive language in texts. Nasril et al. [101] proposed an audio-only approach analyzing pitch and MFCC features of the subjects' voice. Fernandes at al. [54] suggested using spectral and cepstral features of speech signals to detect deception.

The release of the Real-Life Trial dataset [122] prompted more research groups to look into multi-modal deception detection. The dataset authors [134] used a set of features from each modality: unigrams and LIWC categories frequency for lingustic features, manually annotated facial expressions and hand gestures, automatically extracted Facial Action Units, as well as pitch and silence/speech histograms for acoustic features. Same group later extended this work to include thermal imaging data (by collecting the relevant dataset) [3]. Wu et al. [158] used multi-modal approach: visual, acoustic, textual features as well as facial expression annotations to build an ensemble models

[1] https://www.factcheck.org/
[2] https://www.politifact.com/

to detect deception in the Real-Life Trial dataset. Similar general approach but different uni-modal features were used by Rill-Garcia et al. [128] to detect deception in the Real-Life Trial dataset and the Spanish Language Abortion/Best Friend datasets. Several works made an attempt at using Deep Learning for the task of deception detection [138, 60, 78, 42] with the latter proposing an adversarial learning technique to overcome scarcity of data in the Real-Life Trial dataset.

Several works considered deceptive behavior in group settings. Yu et al. [162] used sentiment analysis to infer players' attitude towards each other and to build a network to identify a group of deceitful players. Other works dealing with deception in group setting only use player-level features such as linguistic and lexical patterns [38], facial expressions [39, 138], and speech patterns [31]. The latter exploits some rudimentary group-level information by including such features as the number of successful interruptions, number of turns during the discussion, and others. Wang et al. [152] used an attention mechanism to further analyze the deceptive facial behaviors. Kumar et al. [86, 155] exploited the social interaction networks extracted from the Resistance dataset to identify deceivers.

In this thesis (Chapter 3) we consider very long videos to detect deceptive behavior. We use visual and audio features such as Facial Action Units, Emotion expressions, Head/Eye Movement, VGG Face embeddings, and MFCC to predict long-term deception in a group setting on the Resistance dataset [45]. We also propose a LiarRank meta-feature to exploit group nature of the interaction.

Since the politics and public relations are a natural source of deceitful speech, some researchers directed their interests to this direction. Clementson [34] analyzed how well humans are able to detect lies and truths in political debates and found that humans are significantly truth-biased on average (presume truth by default) but also have high levels of partisan bias (more often think that politicians from an opposing party lie). Another study [33, 32] found that humans are relatively good at detecting when politicians dodge the questions, which negatively affects trustworthiness of the politician in the eyes of viewers, but this effect also highly depends on political attitudes of the viewer.

Kamboj et al. [76] uses several sets of features from multiple modalities (visual: Facial Expressions, Pose, gaze; linguistic: Glove represenataions, LIWC categories frequency, polarity of statements, POS tagging, unigram; acoustic: OpenSMILE features for emotions predictions) as well as their combinations. The authors test their approach on the dataset of US politicians statements. Kopev et al. [84] attempts

at detecting false statements in US Presidential debates. The authors exploit both audio and textual modalities of the data: they use BERT [41] embeddings, TF-IDF vectors [72], and LIWC categories frequencies for the text, and OpenSMILE feaures for acoustic analysis. Windsor et al. [157] considers a very specific case of a political speech: Bill Clinton's testimonies and speeches related to his impeachment process. The authors explored how syntactic (using Coh-Metrix linguistic analysis tool), sentiment (using LIWC dictionary), acoustic (using OpenSMILE and openEAR tools for audio analysis), facial expressions (using iMotions biometric platform), and fusion of the unimodal features affect the deception prediction and which of them provide reliable cues to the deception.

These works only examine each individual politician at the moment of making the statement in question and analyze the behavior of that politician as well as the linguistic properties of the statement. We propose to consider the content of the statement in the broader context: what this politician is talking about, how controversial is the topic, and how likely that politician is to lie on that topic.

2.2 Impression Prediction

2.2.1 Social and psychological science efforts

Since the late last century, social and psychology scientists have been studying dyadic and group impressions between individuals.

In general, positive and negative impressions can be discovered from observing personality traits. Reysen [126] developed a likeability scale asking subjects to rate other people on 11 variables, such as attractiveness, friendliness, similarity to a subject, likeliness to be a friend with a subject. He also observed that genuine laughter is a strong predictor of likeability [126, 127]. One of the most intensely researched personality trait groups are The Big Five [61]: a group of five factors that can be repeatably recognized in various sample data, including extraversion, agreeableness, conscientiousness, emotional stability, and culture.

Researchers have also looked into cause and effects interactions that result in the liking or disliking of another. Davydenko et al. [36] observed that the way people act (e.g. extroverted or introverted) affects how other participants judge them, as well as serve as a part of the feedback loop: a person interacting with an extroverted acting

partner is perceived as nicer and displaying more positive social behavior.

Seiter et al. [133] showed that even background nonverbal behavior can alter the impression of a person. In an experiment where one debater non-verbally expresses her agreement or disagreement with another speaking debater, Seiter shows that the nonspeaking debater is perceived as less likable if she expresses moderate or constant disagreement compared to a neutral expression.

According to Floyd and Burgoon [56], nonverbal expressions, such as smiling or gaze attention, can be more provocative than verbal expressions. The experiments conducted showed that the specific combination of gaze attention and smiling is the most impactful combination to the receiver of said expressions compared to other possible combinations.

Besides nonverbal actions and behavior, Hareli et al. [64] showed that knowledge about the emotional response of a person can be used by people in forming impressions of that person. In other words, they discovered that people associate personality impressions with emotional reactions. This "reverse engineering" of a situation creates outcomes such as deeming someone negatively when he or she is angry in an unpleasant situation.

One important theory espoused by various social scientists about the concern of recognizing an individual's impression of another is the Interpersonal Adaptation Theory (IAT), which can be considered to be a summation of impression theories [27]. The theory states that people have preset expectations when interacting with others. The individual's behavior can thus be predicted if these conditions are met or not met. For instance, if Person B seems aggressive to Person A when Person A expected nothing of the sort, then it can be predicted by IAT that Person A will have a negative impression of B.

Besides creating prediction theories over impression detection, social scientists have also analyzed sentiment change over time among strangers. Bruce and McDonald [22] suggest that a person will continue to like a stranger if there was an initial positive reaction. Interestingly enough, an initial negative reaction does not lead to one disliking another for long periods of time. Instead, people tend to forget unlikable faces, and thus the possibility of an impression change from dislike to like appears to be higher than constant displeasure. This finding also reflects in the data used for this thesis.

To detect the change from dislike to like in individuals, various features were built

to reflect social science theories. We also look at group dynamics for insight. For instance, Nisbett [104] finds that sentiment is correlated with the subject's popularity in the group. In other words, even if someone has a negative initial opinion of another person, that person's popularity can decrease the negative sentiment from the former.

Another important quality of human interaction is rapport. Tickle-Degnen and Rosenthal [141] identify three essential components of rapport: mutual attentiveness, positivity, and coordination between participants. Rapport, in Tickle-Degnen and Rosenthal's terms, means mutual responsiveness between individuals. Specifically, nonverbal behavior is seen as a quintessential component of identifying rapport. During a helpful context, a person mirroring another person has a strong correlation for mutual positive attitudes of each other. While context is important in weighing each of the three components, it should be recognized that all are necessary for positive liking to take place. Furthermore, if there is a positive relationship between two individuals, social science research suggests a certain amount of facial mimicry occurs. Murata [99] suggests, for instance, that people will emulate someone's grin if they like that person: smiling means more smiling. Overall, it would seem a person's gaze or attention drives this mimicry. It is also noted by Murata [99] that disliked people are mimicked less.

From the research published by social scientists, we decided to narrow our scopes for the sake of practicality. Our research is focused more on individuals forming negative impressions because previous social science research suggests that negativity is far easier to identify than its opposite. Floyd et al. [56] supports this reasoning by mentioning that people's expression of dislike is more "uncontrollable" and external while acceptance is generally internally expressed and more controlled in gesture.

2.2.2 Computational efforts

Over the past decade, there was also an increasing interest in automated analysis and modeling of human-to-human interaction in dyadic or group settings as well as computer-mediated interactions.

Several datasets were proposed as benchmarks for a variety of human interaction related tasks. SEMAINE [93] and its modifications contain videos of people conversing with human-driven, semi-automatic, and automatic virtual agents, and annotations of perceived personality traits. ELEA [130] captures collaborative group interaction, the Big Five traits, and leadership behavior annotated by external observers as well

as perceived leadership and likeness reported by group participants. MATRICS [103] uses several modalities (motion capture, gaze tracking, head acceleration, video, audio, Kinect sensor) to record a small group of people participating in a task-oriented discussion. ChaLearn First Impressions dataset [120] comprises a set of YouTube videos with human-annotated the Big Five scores. VLOG [18] is another dataset of YouTube videos with crowdsourced personality impressions. In addition to that, several datasets were published with a focus on job interviews and hireability prediction [69, 102].

The majority of the work in this area is focused on predicting apparent personality traits, the most common ones are the Big Five and leadership style, which is partly due to the availability of the relevant data. Joshi et al. [74] used Pyramid of Histogram of Gradient to predict perceived traits in videos of humans interacting with virtual agents expressing various personalities. Chávez-Martínez et al. [30] use multimodal features for multi-label prediction of moods and traits in the VLOG dataset exploiting high correlations between moods and perceived traits. Sheng et al. [51] used a variety of semantic visual and audio features as well as dyadic and group features for personality traits prediction in small groups and provided some insights on the importance of those features for predicting particular traits. Çeliktutan et al. [172, 173] used visual features such as Histogram of Gradients and Histogram of Optical Flow as well as audio features such as MFCC to predict perceived traits in videos of humans interacting with virtual agents. Kindiroglu et al. [81] used multitask and transfer learning for extraversion and leadership prediction on ELEA and VLOG datasets with the same set of high-level audio-visual features. Kampman et al. [77] proposed an end-to-end deep model for multimodal impression prediction. Beyan et al. [16] used DNN based features for leadership and high/low extraversion prediction. In other work [17] they used dynamic images and activity-based information for personality traits inference. Mawalim et al. [92] investigated the effectiveness of multimodal features such as acoustic, head motion, and linguistic features for the personality traits prediction. Anselmi et al. [4] used deep neural networks to infer self-reported personality traits from highly constrained still face images. Zhang et al. [164] proposed an end-to-end deep neural network for joint prediction of apparent personality and emotions, showing that joint task improves upon separate traits or emotions prediction. Muller et al. [98] proposed a framework for detecting low rapport in a small group setting using speech, facial, and body movement features. Bai et al. [10] suggested a framework for detecting the most dominant person in a group and relative dominance between two people by exploiting interaction dynamics within

the group. Okada et al. [107] proposed co-occurrence pattern mining in multimodal features such as speech, head movement, body movement, and gaze for leadership and the Big Five prediction. Zhang et al. [165] also considered co-occurring visual events for the Big Five traits predictions. Recently some efforts also went into explainability and interpretability of the impression predicting systems [48].

Some research focused on assessing communication skills in group setting [108, 103]. Lin et al. [90] built a conversational Graph Convolutional Network from participant acoustic and lexical features to predict group performance outcomes in the ELEA dataset. Eloy et al. [47] used face and upper movement multi-dimensional recurrence quantification analysis to model team level dynamics and predict the collaborative outcome.

One of the motivations for modeling human personality and interpersonal communication is building better devices aimed at assisting humans. Zhang et al. [167] explored the way of assessing personal affect and team cohesion in small groups using wearable devices capable of recording data about movement, face-to-face interaction, location, and audio communication.

Our work differs from previous research in several key aspects. First, we predict dyadic impressions as opposed to group impressions. In other words, we predict how humans in the group perceive each other, and not how the group as a whole perceives a person, or how a group of external observers perceives that person. Second, we actively employ the group interaction nature of our dataset and propose a set of features and a multi-layer network model aimed at capturing social dynamics.

2.3 The Resistance dataset

Our collaborators designed and collected the new Resistance dataset [45] based on the social role-playing, card-based party game The Resistance (some variations of the game are also known as Mafia or Werewolf). In this section, we will describe the nature of the game and the dataset based on it.

2.3.1 Game description.

Figure 2.1 outlines the process of the game as it was conducted during the data collection. At the beginning of each session, the facilitator of the game introduces

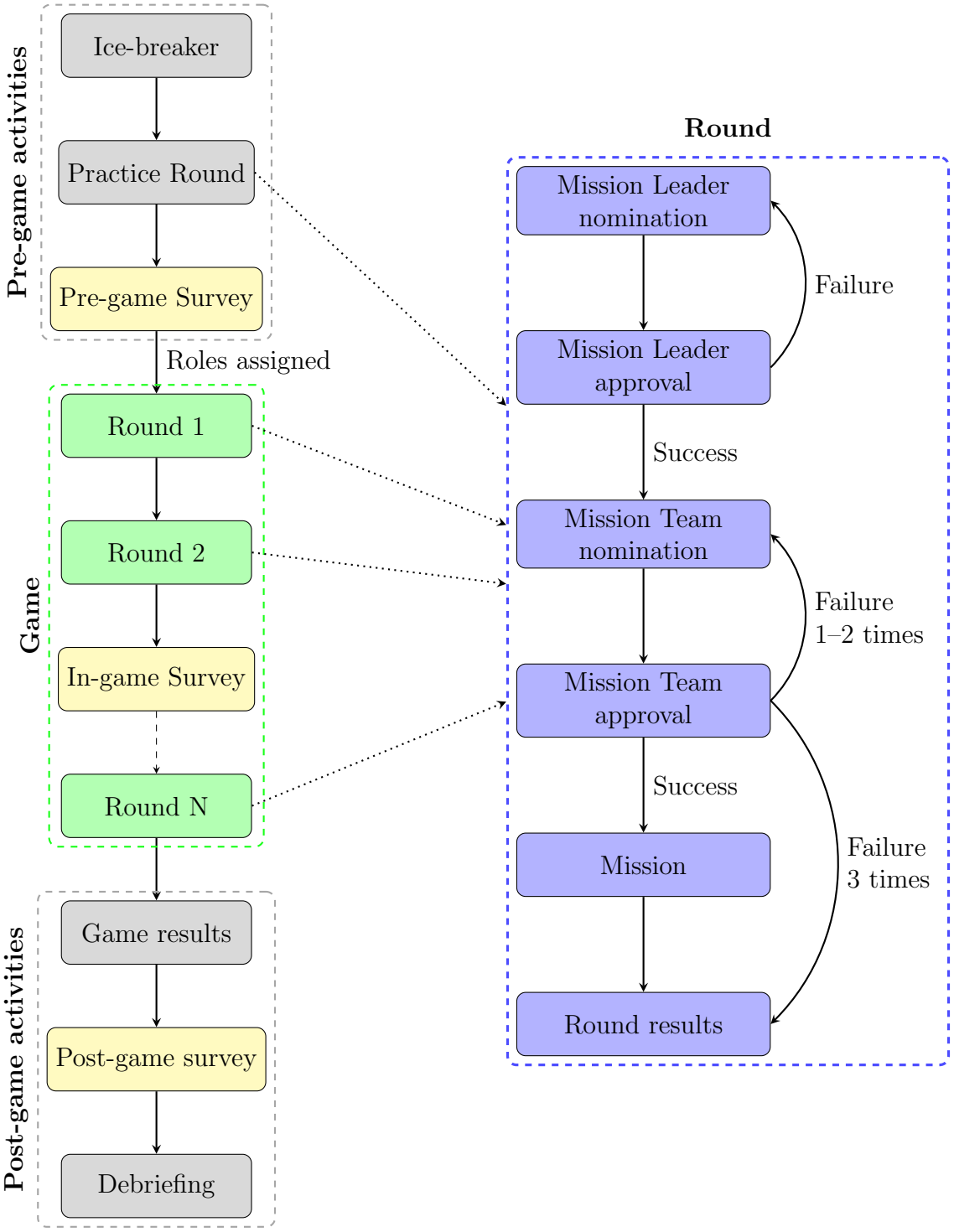


Figure 2.1: The Resistance game process. The game session starts with pre-game warm-up activities, proceeds to the game consisting of several rounds, and ends with a post-game period. Each round goes through several steps: mission leader nomination and election, mission team nomination and approval, the mission itself.

Number of players	5	6	7	8
Number of spies	2	2	3	3
Number of villagers	3	4	4	5

Table 2.1: Rules of the Resistance game: number of spies depending on the total number of the players

themselves and invites the participants to introduce themselves as well. This interaction serves as an ice-breaker activity and allows players to get to know each other better before the game starts. Each participant is asked to state her name, her major, and one interesting thing about herself. After that, a participant sitting across the newly introduced participant has to ask her a follow-up question.

After all players introduced themselves, the facilitator explains the rules of the Resistance game, and participants play one practice round, which is identical to the first round of the actual game, except all votes, including mission vote, are public. The practice round is intended to ensure the players understand the rules of the game.

Number of players	R1	R2	R3	R4	R5	R6	R7	R8
5	2	3	4	4	3	3	-	-
6	3	3	4	4	4	5	5	5
7	3	3	4	4	5	4	4	4
8	3	3	4	4	5	5	5	5

Table 2.2: Rules of the Resistance game: number of players on a mission in every round depending on the number of players in the game

Number of players	R1	R2	R3	R4	R5	R6	R7	R8
5	1	1	1	2	1	1	-	-
6	1	1	1	1	1	2	2	2
7	1	1	1	2	2	2	2	2
8	1	1	1	2	2	2	2	2

Table 2.3: Rules of the Resistance game: number of failure votes required to fail the mission in every round depending on the number of players in the game

After the practice round, the players are asked to fill in a survey (Table 2.5). When all participants complete the pre-game survey, the game starts, and each player is randomly assigned a role: a "villager" or a "spy". The number of spies in the game depends on the total number of players in that game (Table 2.1). Spies are also

secretly informed about who other spies are. Villagers are only aware of their own role.

Then, the game proceeds in rounds (every round is called a "mission"). Each round consists of the following steps:

1. First, players nominate and discuss a mission leader among themselves. Each round, a mission leader has to change so that no player can be a mission leader in two consecutive rounds.

2. Then, players vote to approve or reject the nominated leader. The vote happens secretly (using tablets in front of the players) and then publicly (by raising hands), but only secret votes count towards the game. If the nominee obtains a majority of votes, she is approved, and the game proceeds to the next step. If the nominee fails to secure the majority of votes, her nomination fails, and the Step 1 starts again. These steps repeat until someone receives the majority of player votes. Each time the game facilitator announces the secret ballot results (number of votes for approval and against it) without specifying individual players' votes.

3. Then, the elected mission leader nominates several players for the mission. The number of players to go on a mission is announced before each round. It depends on the total number of players in the game and the round number (see Table 2.2).

4. After a discussion, all players vote to approve or reject the proposed mission team. As in Step 2, players vote secretly and publicly, but only secret vote counts. If the nominated team receives the support of the majority, then it is approved, and the round proceeds to the mission. If the team cannot obtain the majority of votes, it is rejected. In this case, the mission leader has to nominate another group of players, and steps 3–4 repeat. If nominated teams get rejected three times, the round ends with the point going to the spies. As in step 2, the game facilitator announces the final vote toll without detailing individual votes.

5. Finally, the approved mission team "goes on a mission": each member of the mission team secretly votes to succeed or fail the mission. The number of fail votes necessary to fail the mission is announced at the beginning of the round and depends on the round and the number of players in the game (Table 2.3). The game facilitator announces vote toll. If the necessary number of "fail"

votes is achieved, the mission fails, and the spies get the point for the round. Otherwise mission is successful, and villagers get the point.

The game lasts for 4–8 rounds and is limited by the total session time. If spies and villagers have equal scores by the end of the allotted time, the facilitator runs a breaking round. The team with the highest score at the end of the game wins and receives additional monetary incentives after the game. Moreover, elected mission leaders also receive a financial reward which incentivizes players to nominate themselves as leaders during the rounds.

After every even-numbered round, the players fill in an in-game survey (Table 2.5).

After the last round and the announcement of the results of the game, all players are asked to complete a post-game survey (Table 2.5). In order to keep answers unbiased, the players are prohibited from sharing their roles with others until they leave the room. Finally, all participants are debriefed about the goals and details of the study.

By the nature of the game, spies and villagers have different incentives throughout the process. Spies want to stay stealthy as long as possible and get elected on as many missions as possible to fail them and earn points for the team. Villagers want to collectively identify spies as early as possible to prevent them from getting on the missions. Thus, this game has both a collaborative and adversary nature.

2.3.2 Dataset description.

The Resistance dataset contains data from 95 games with a total of 697 players. These games were conducted under IRB authorization at eight geographical locations

Country	University	# of games	# of players
	UCSB	11	78
USA	University of Arizona	9	61
	University of Maryland	10	70
Zambia	The University of Zambia	15	117
Hong Kong	Hong Kong Polytechnic University	15	115
Singapore	Nanyang Technological University	12	84
Israel	Tel Aviv University	9	64
Fiji	University of the South Pacific	14	108

Table 2.4: The Resistance data geographical distribution.

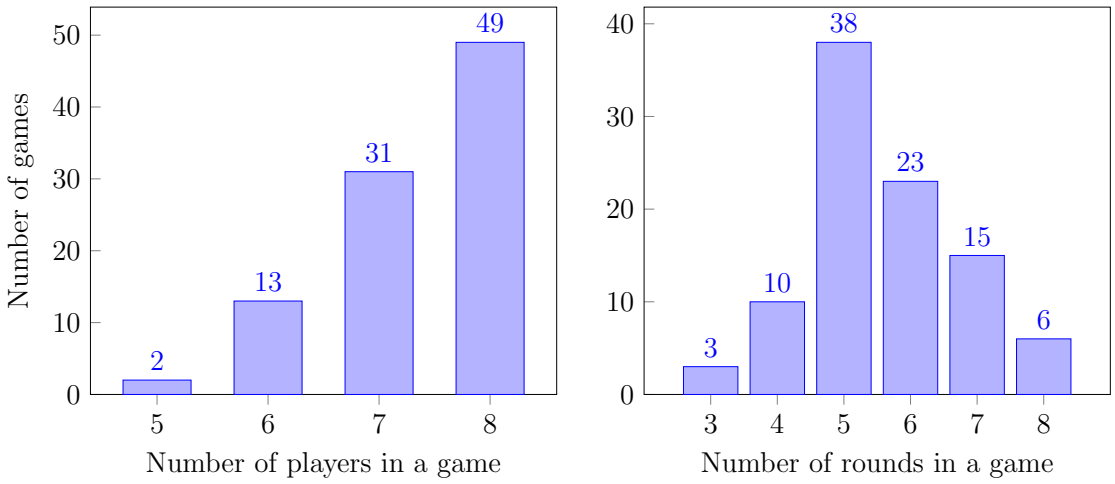


Figure 2.2: The Resistance game players and rounds distribution.

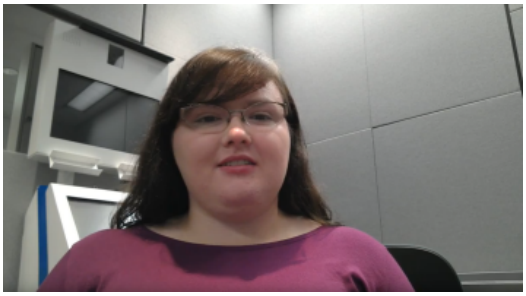
spanning different continents and cultures (Table 2.4). Most of the participants were recruited among college students through college message boards. The average age of the players is 22.4 with the youngest participant to be 16 and the oldest to be 48. Gender distribution is the following: 47.2% of players were male, and 52.8% were female. Participants reported various ethnicities, with the most popular being Asian (39.2%) and White (18.2%).

Figure 2.2 shows the distribution of games with respect to the number of players and the number of rounds: most of the games lasted for 5–6 rounds and had 7–8 players. The average duration of a game was 46 minutes, with lengths varying from 29 minutes to 66 minutes.

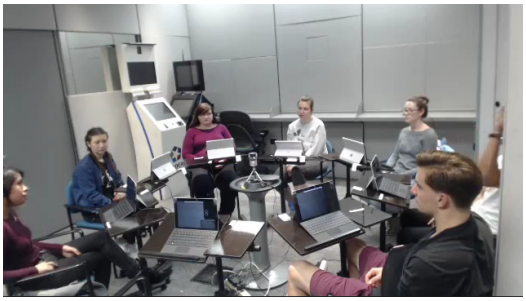
The Resistance dataset comprises video footage of the game sessions, information on the game progress, and survey results filled by the players. Figure 2.3 shows how several cameras capture every game. Frontal video of every player is recorded with tablet cameras in front of the player. The 360-degree camera records the video of all players at once. One or more overhead cameras record the overall view of the room where the game is conducted. For this research, we only use the high-definition quality frontal videos recorded on the tablet cameras (Figure 2.3a). These videos were recorded with 29.98 fps frame-rate, 1920×1080 resolution, and $16 : 9$ aspect ratio.

In addition to audio-visual data from the cameras, the dataset also includes surveys that players fill in before the game (pre-game survey), after the game (post-game survey), and after every two rounds of the game (in-game survey) as shown in Table 2.5.

Pre-game survey records the basic sociological data about the player (age, sex, ethnicity, major, native language and the county of origin, level of English language fluency, and others), psychological questionnaire (questions related to the Big Five traits, questions intended to quantify player's cultural background), questions about player's perception of other players (trust, dominance, likeability, nervousness). In the in-game surveys, players score each other on a number of traits: how dominant each player is, how nervous or anxious each player looks, how much the player trusts other players, and who the player thinks spies might be. Post-game surveys contain more fine-grained questions about the same traits. For example, for likeability trait, the survey asks to rate each player on being cold vs. warm, friendly vs. unfriendly, etc.; for trust, the players need to rate each player on being useful vs. useless, honest vs. dishonest, deceptive vs. truthful, etc. The post-game survey also asks game-related questions (e.g. whether the player was engaged or bored, whether the other players were engaged and how they behaved) and introspective questions (e.g. how often the player lied, which non-verbal clues the player used to identify liars, how the game made the player feel).



(a)



(b)

(c)

Figure 2.3: Setup of the game. Players sit in a circle. In front of each player, there is a tablet with a camera (a). There are also one or more overhead cameras (b) and a 360-degree camera (c).

Question type	Pre-game	In-game	Post-game
Sociological questions: age, sex, ethnicity, college major, country of origin, etc.	×		
Psychological survey: the Big Five, group vs. individual, etc.	×		
Personal qualities of other players: likeable, friendly, etc.	×	×	×
Trust-related qualities of other players: trustworthy, honest, etc.	×	×	×
Dominance-related qualities of other players: dominant, quiet, active, etc.	×	×	×
Nervousness-related qualities of other players: nervous, anxious, tense, etc.	×	×	×
Suspected spies		×	
Game-related questions: enjoyed the game, felt engaged, etc.			×
Introspection questions: how often lied, how the game made feel, etc.			×
Other players' engagement: how interested, how involved, etc.			×

Table 2.5: Survey questions. Some questions only appear in pre- or post-game questionnaires. Some questions appear at all stages of the game but with different granularity: in-game survey asks for overall judgement of a quality while post-game survey asks about specific manifestations separately.

Game information contains records of the player's actions during the game: whether the player was a spy or a villager, whether the player was elected a mission leader or a mission team member in every round, and all player's secret votes.

Because of the incremental data collection process, not all of the Resistance dataset videos were available at the time of experiments for our projects. Actual numbers of videos used for each of the tasks are specified in corresponding chapters (Chapters 3 and 5).

Chapter 3

Long-Term Deception Detection

Past work on automated deception in video [168, 122, 158] focuses on videos of a single person in a short (15–200 second) clip. In this chapter, we present a fully automated system (LiarOrNot) in which we take a frontal video of a subject interacting with a group and predict whether that person is being deceptive in the long term, i.e. across the duration of a 30–65 minute video. The Resistance game and its variants such as Mafia and Werewolf naturally induce long term deception in a highly interactive group setting. The Resistance game usually involves 5-8 players, 2-3 of whom are designated "spies" who win the game if they are not discovered. Thus, they must be deceptive throughout the game, but must intermix lies with truth in order to stay undiscovered by others. We develop methods to predict "spies" and "honest" players in the game as a proxy for the deception detection task.

In addition to the fact that long-term deception in group settings has been rarely studied, LiarOrNot makes the following innovations. Building on well-known image (VGG Face) and audio features (Mel-frequency cepstral coefficients),

(i) we introduce a class of histogram-based features that build on well known low-level (eye/head movement, facial action units) and high-level (emotion features from Amazon Rekognition) features,

(ii) we introduce a novel class of "meta-features" called LiarRank that builds on the basic features, and

(iii) we introduce an ensemble based prediction model.

Our 10-fold cross validations split the *entire* set of videos into training and testing sets based on games. Hence, LiarOrNot predicts on games and people that are completely disjoint from those seen in training. We show that LiarOrNot achieves an AUC of 0.705 in this hard test, significantly outperforming other feature classes and past work. Additionally, as our data set was collected across three very different countries

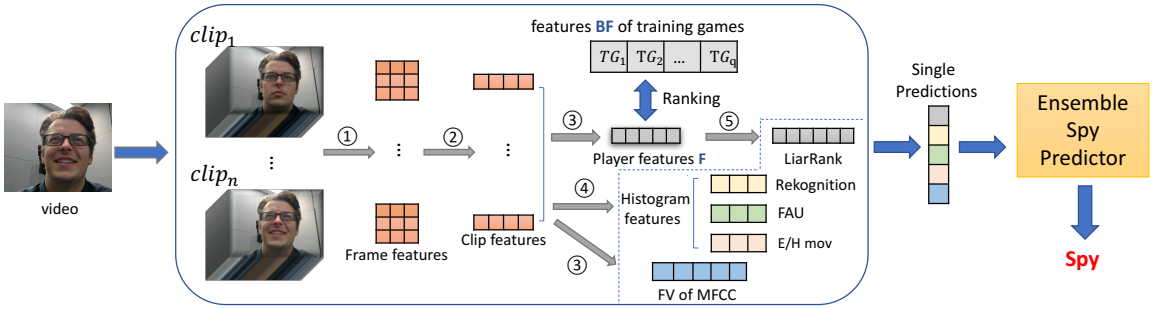


Figure 3.1: LiarOrNot Architecture. Steps: Uniformly sample n clips from a player's video, then (1) extract frame features, including VGG Face, emotions, facial action units and eye/head movements, (2) aggregate frame features and sub-second MFCC features to clip features, (3) and (4) aggregate previous features to player features, (4) is histograms of low-dimension high-level features, and (3) is Fisher Vectors (FV), (5) build LiarRank of player features. Finally, predictions made from each feature type are used in our ensemble spy predictor to generate the final prediction.

and because there may be cultural differences in deception, our results are more robust across cultures than past studies (though much additional work needs to be done to capture African and Latin American cultures as well).

3.1 Methodology

Architecture. Figure 3.1 shows the LiarOrNot architecture. Let $\mathcal{TG} = \{TG_1, \ldots, TG_n\}$ be the set of training game videos (e.g. in some fold of cross validation) and let TG_{n+1} be any game (either in $\mathcal{TG}$ or not). In any game TG_j, let p_i^j be the i'th player in that game. In our data, i varies from 1 through a max of 8. Each player p_j^i's frontal camera captures a video v_j^i of that player of length 30–65 minutes. *Each player appeared in exactly one game.* Since we wish to predict whether a player p_j^i is deceptive or not, each player needs to have an associated feature vector $fv(p_j^i)$ which we define as either a basic feature vector $bf(p_j^i)$ or a LiarRank meta-feature vector $sr(p_j^i)$.

The rest of this section is organized as follows. We first explain the concept of LiarRank, showing how to associate a LiarRank meta-feature vector $sr(p_j^i)$ with player p_j^i. We then explain how the "basic" features are derived. Finally, we explain our ensemble predictor. Throughout this section, we use the "dot" notation to denote the connection between representations and level of aggregation, e.g. $fr.f_i$ denotes feature f_i of the frame fr, and $Cl.\boldsymbol{f}$ denotes feature vector $\boldsymbol{f}$ of clip Cl.

Algorithm 1: LIARRANK($\mathcal{TG}, TG_{n+1}, p^{\ell}_{n+1}, f_h$)

Input : Training set $\mathcal{TG} = \{TG_1, \ldots, TG_n\}$, Player p^{ℓ}_{n+1} from some game TG_{n+1}, basic feature f_h

Output: $sr_h(p^{\ell}_{n+1})$

```
for j ∈ [1, ..., n] do
    Vals(f_h, j) = {p^ℓ_{n+1}.f_h} ∪ ⋃_{i=1}^{8} {p^i_j.f_h}
    Sort Vals(f_h, j) in descending order
    r_j = position of p^ℓ_{n+1}.f_h's value in Vals(f_h, j)
end
return the vector ⟨r_1, ..., r_n⟩
```

3.1.1 LiarRank Features

Suppose $BF = \{f_h\}_{h=1}^{k}$ is any set of basic features. Given any basic feature f_h, we will first define the LiarRank $sr_h(p^i_j)$ of player p^i_j w.r.t. feature f_h. The LiarRank vector $sr(p^i_j)$ is then the vector $\langle sr_1(p^i_j), \ldots, sr_k(p^i_j)\rangle$ obtained by concatenating these individual feature-ranks.

The LiarRank algorithm shown above (Algorithm 1) takes as input, a training set $\mathcal{TG} = \{TG_1, \ldots, TG_n\}$, a game TG_{n+1} (which could be in $\mathcal{TG}$ or not), as well as a player and a single feature f_h. It returns a vector of length n (i.e. number of games in the training set) which captures the position of players p^{ℓ}_{n+1}'s value for feature f_h w.r.t. the corresponding values for other players in each of the n games. To do this, it computes the value of the feature for the player p^{ℓ}_{n+1} as well as every player who participated in any of the training games. The resulting set of features values is stored in the set $Vals(f_h)$. This set of values is then sorted in descending order. The first item in the descending order has position (or rank) 1, the second has position (or rank) 2, etc. The LiarRank of player p^{ℓ}_{n+1} w.r.t. feature f_h is its position in the sorted $Vals(f_h)$ list. *Intuitively, LiarRank of player p^{ℓ}_{n+1} w.r.t. feature f_h is the relative rank of player p^{ℓ}_{n+1} had she participated in that game.*

The above defines the LiarRank vector of a player w.r.t. a feature. The LiarRank vector of a player is the concatenation of the feature vectors. (further illustrated in Figure 3.2). There is some similarity between LiarRank and the rank transform proposed in [163] and the local binary pattern descriptor (LBP) in computer vision.

Example. We illustrate LiarRank via a simple example using Figure 3.2. In this example, there are three games and we are considering feature f_1. The values of this feature for the players in the games are shown in the three tables labeled TG_1, TG_2,

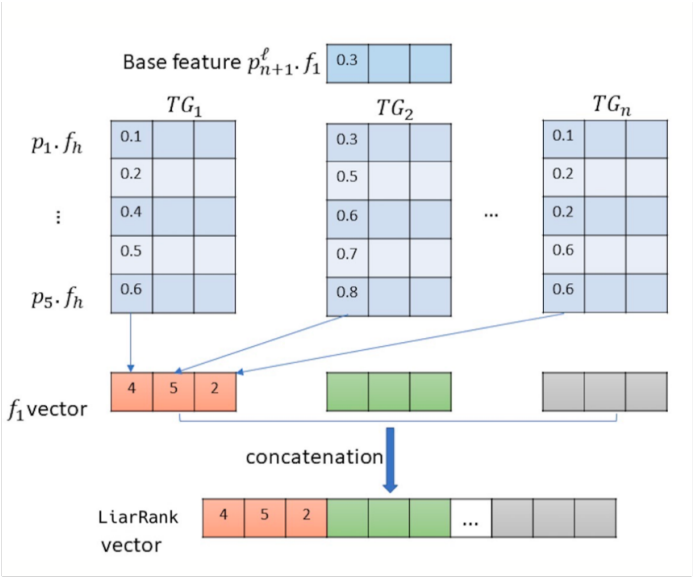


Figure 3.2: LiarRank meta-feature calculation process.

TG_3 and that value of this feature for the player p^l_{n+1} is 0.3. As there are three base features in this highly simplified example for each player, the LiarRank $sr(p^l_{n+1}, f_1)$ vector will be of length three. The rank of the first player w.r.t. the players in TG_1 is $r_1 = 4$ because the set $Vals(f_1, 1)$ after sorting is $\{0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$ and $p^l_{n+1}.f_1 = 0.3$ is the fourth entry in this list. When we consider the second game, the set $Vals(f_1, 1)$ after sorting is $\{0.8, 0.7, 0.6, 0.5, 0.4, 0.3\}$ setting $r_2 = 5$. In the same way, $r_3 = 2$. Thus the LiarRank vector for p^l_{n+1} w.r.t. f_1 is $<4, 5, 2>$. Vectors of length 3 each for features f_2, f_3 are similarly calculated and concatenated to obtain the final LiarRank vector for player p^l_{n+1}.

3.1.2 Basic Features

Sampling. We sample 10-second clips at an interval of 30 seconds per video. Since games are 30-65 minutes long, different videos may consist of different numbers of clips. From each clip, we further sample a set of $m = 300$ frames for Eye/Head Estimations and $m = 20$ frames for the rest of visual features (see below). As low/high-level video features as well as audio features for each player may vary substantially over the length of the video, we define features at both the frame-level and clip-level. For each $(ClipId, FrameId)$ pair, we extract a set of basic features.

Basic Frame Features. Once frames are extracted, we extract the following basic features:

- **VGG Face** [112] is a deep neural network pretrained on a large scale face recognition dataset. To obtain VGG Face features, we first detect the player's face in the frame using OpenFace [12], then input the cropped rectangle containing the detected face to the VGG Face network and extract activations of the fully connected layer ("fc6"). We expect these 4096-dimensional activation vectors to contain high level information about faces. We perform PCA on these representations to reduce dimensionality to 512.

- **Facial Action Units (FAU)**. We use OpenFace [12] to predict the intensity of 18 Facial Action Units [13, 46] on a 5-point scale. This software detects faces and relevant points on them and then uses pre-trained deep neural models to extract FAUs and their intensities.

- **Eye/Head Estimations (EHEs)**. OpenFace is also used for estimating eye gaze angle, head position and rotation. We calculate eye and head movement features as a difference between key point coordinates of successive frames.

- **Basic Expressions**. Amazon Rekognition, a proprietary cloud-based service[1] provides estimations of seven emotion intensities (happy, sad, angry, confused, disgusted, surprised, calm) and three facial attributes (open eyes, open mouth, smile) for a face in the frame. Each emotion and facial attribute feature is a real value from 0 to 100. These seven emotions are widely considered as basic in psychological literature [142, 29] and used as targets in automatic emotion recognition models [53].

- **Mel-Frequency Cepstral Coefficients (MFCC)** [35] features are widely used for speech recognition tasks. For each sampled audio clip, we use a sliding window with a window-size of 25 ms and step size of 10 ms, then we get a series of MFCC features corresponding to short intervals.

Basic Clip-level features. We aggregate frame-level features into clip-level features with average-pooling. If a clip Cl is a set of sampled frames, then the value of a clip-level feature f_h for clip Cl is given by $Cl.f_h = \frac{1}{|Cl|}\Sigma_{fr \in Cl} fr.f_h$. Clip-level

[1] https://aws.amazon.com/rekognition/

features smooth variations in frame level features, especially as those variations can be substantial for some features, e.g. emotion features.

Player-level features. As the goal is to extract features at a per-player level, we aggregate clip-level features into player-level features using Fisher Vectors (for VGG Face representations), or histograms (for Facial Action Units, Eye/Head movement, and Amazon Rekognition features).

Fisher Vector features. Fisher vector (FV) is a bag-of-words based model heavily used for object recognition in images [118]. Note that each video may have a different number of clips. Fisher Vectors aggregate the clip level features of an arbitrarily long video into a fixed length encoding. It first builds a K-component GMM model $(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i, w_i : i = 1, 2, ..., K)$ from all the clip-level features in training data, where $\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i, w_i$ are the mean vector, diagonal of a covariance matrix, and mixture weights for the i^{th} component, respectively. Given a player Pl and clips that were extracted for this player $Cl_t \in Pl$, we first extract clip-level features $\{Cl_1.\boldsymbol{f}, Cl_2.\boldsymbol{f}, ..., Cl_{|Pl|}.\boldsymbol{f}\}$, where $Cl_t.\boldsymbol{f}$ is a clip-level feature vector for the t^{th} clip, and $|Pl|$ is the number of clips for the player Pl. It's Fisher Vector is computed as:

$$\mathcal{G}_{\boldsymbol{\mu}_i} = \frac{1}{|Pl|\sqrt{w_i}} \sum_{t=1}^{|Pl|} \gamma_t(i) \left(\frac{Cl_t.\boldsymbol{f} - \boldsymbol{\mu}_i}{\boldsymbol{\sigma}_i} \right)$$

$$\mathcal{G}_{\boldsymbol{\sigma}_i} = \frac{1}{|Pl|\sqrt{2w_i}} \sum_{t=1}^{|Pl|} \gamma_t(i) \left(\frac{(Cl_t.\boldsymbol{f} - \boldsymbol{\mu}_i)^2}{\boldsymbol{\sigma}_i^2} - 1 \right)$$

where, $\gamma_t(i)$ is the posterior probability. We then concatenate all the $\mathcal{G}_{\boldsymbol{\mu}_i}$ and $\mathcal{G}_{\boldsymbol{\sigma}_i}$ to form the $2DK$-dimension Fisher Vector $Pl.\boldsymbol{f}$, where D is the dimensionality of a clip-level feature vector $Cl_t.\boldsymbol{f}$.

For audio, we found that computing the Fisher Vector directly from frame-based features (small audio intervals near a frame) and bypassing the clip-level features performs better than averaging MFCC features over a clip. In this case, the bag of frame-level features will consist of all the feature vectors from all the clips of the video (and not just the initial 10 seconds in the 30 second window).

Histogram features. We compute three types of histogram features for every basic feature such as Facial Action Units, Eye/Head movement, and Amazon Rekognition features. These are histograms of frame-level features, histograms of clip-level features, and combination of the first two.

For a player Pl and a basic frame feature f_h, we have a set of all feature values for all frames $\{fr_{st}.f_h\}$, where $fr_{st} \in Cl_t$ and $Cl_t \in Pl$ (or a set of clip-level features $\{Cl_1.f_h, Cl_2.f_h, \ldots, Cl_{|Pl|}.f_h\}$ where $Cl_i \in Pl$.). We build a histogram of frame-level features $\mathcal{V}_h^{frames} = \langle v_h^1, v_h^2, \ldots, v_h^b \rangle$ where v_h^i are frequencies of values $fr_{st}.f_h$ falling into the i^{th} bin, and b is the number of bins (similarly $\mathcal{V}_h^{clips} = \langle v_h^1, v_h^2, \ldots, v_h^b \rangle$ for a histogram of clip-level features). We form a histogram feature by concatenating histograms for all or some of basic features $Pl.\boldsymbol{f} = \langle \mathcal{V}_{h_1}^{frames}, \mathcal{V}_{h_2}^{frames}, \ldots \rangle$ (or $Pl.\boldsymbol{f} = \langle \mathcal{V}_{h_1}^{clips}, \mathcal{V}_{h_2}^{clips}, \ldots \rangle$ for clip-level histograms). For the same player Pl and a basic feature f_h, we have a set of clip-level features $\{Cl_1.f_h, Cl_2.f_h, \ldots, Cl_{|Pl|}.f_h\}$ where $Cl_i \in Pl$. We build a histogram $\mathcal{V}_h^{clips} = \langle v_h^1, v_h^2, \ldots, v_h^b \rangle$ where v_h^i are frequencies of values $Cl_t.f_h$ falling into i^{th} bin and b is the number of bins. We form a histogram feature by concatenating histograms for all or some of basic features $Pl.\boldsymbol{f} = \langle \mathcal{V}_{h_1}^{clips}, \mathcal{V}_{h_2}^{clips}, \ldots \rangle$.

Finally, we also build combined histogram features by concatenating frame-level histograms and clip-level histograms of the same combination of features

$$Pl.\boldsymbol{f} = \left\langle \mathcal{V}_{h_1}^{frames}, \mathcal{V}_{h_1}^{clips}, \mathcal{V}_{h_2}^{frames}, \mathcal{V}_{h_2}^{clips}, \ldots \right\rangle.$$

Optimal number of bins b is determined through cross-validation.

3.1.3 Ensemble classifier

The previous steps associate with each player p_j^i a feature vector $fv(p_j^i)$ represented by the basic features or associated LiarRank features listed above at the player level (aggregating from frame- and clip-levels as described above). Thus, there are five types of features: LiarRank of Fisher Vector of VGG Face, Facial Action Units, Rekognition Emotions, Eye/Head movement, and MFCC. We trained a suite of classifiers and used them to produce a late fusion model. Each classifier returns a *score* denoting the probability of a subject being a spy. If S_i is the score returned by a classifier for the ith feature type for $i \in \{1, \ldots, 5\}$, then the final score S is obtained by late fusion

of named models:

$$S = \sum_{i=1}^{5} \alpha_i S_i \ ,$$

where $\sum_{i=1}^{5} \alpha_i = 1$. Late fusion weights α_i are obtained by grid-search and cross-validation. For each of the five types of features, we select the best classifier, and combine them as above via late fusion.

3.2 Experimental results

We use videos of 285 players from 44 Resistance games. We split the dataset into 10 folds by games, i.e. all players from a game are in either the training or the testing part of a fold. Our classifier suite includes: k-Nearest Neighbors (KNN), Logistic Regression (LR), Gaussian Naive Bayes (NB), Linear SVM (L-SVM), and Random Forest (RF). As a performance metric we report the mean AUC over 10 folds.

For LiarRank feaures, we had to employ the greedy feature selection (FS) technique due to the very large dimensionality of the feature vectors. This procedure works as follows. We start with an empty feature vector as a base. At the beginning of each iteration, the base vector's length is L ($L \geq 0$). We form new feature vectors by adding to the base one feature from the LiarRank vector at a time if that feature is not in the base yet. We then train a classifier on the newly formed features of the length $L + 1$. We then select the best-performing feature and use it as a base for the next iteration. This process repeats until there is no increase in performance metrics on the validation set. To accommodate a 10-fold setting, we perform this process in the nested loop. For each fold, we set it aside as a testing fold. We then perform 9-fold cross-validation to determine the best-performing feature vector that we subsequently test on the testing fold.

3.2.1 Prediction using single-feature classifiers

LiarRank. Table 3.1 shows performance of different aggregations from VGG Face-based and MFCC-based features including LiarRank. As a baseline we use the feature obtained by averaging all frame-level VGG Face features. This baseline does not even achieve 0.55 AUC, which means simple averaging is not a good strategy to capture the relevant behavior of a player over a long video.

Features	RF	L-SVM	NB	LR	KNN
Average VGG Face (baseline)	0.516	0.533	0.549	0.546	0.50
VGG Face clip-level voting	0.503	0.520	0.550	0.527	0.479
FV of VGG Face	0.468	0.573	0.502	0.584	0.502
FV of VGG Face + FS	0.506	0.470	0.491	0.467	0.522
LiarRank of FV of VGG Face + FS	0.639	0.647	**0.663**	0.652	0.603
FV of MFCC frame-level	0.606	0.395	0.56	0.608	0.579
FV of MFCC clip-level	0.586	0.441	0.533	0.579	0.595

Table 3.1: Performance (AUC) of different aggregations of visual (VGG Face) and audio (MFCC) representations. Top to bottom: 1. Average pooling of all frames; 2. Clip-level VGG Face features are used to train and test, scores are averaged for player-level inference; 3. Fisher Vector of clip-level VGG Face features; 4. Fisher Vector of clip-level VGG Face features after feature selection procedure; 5. LiarRank of the Fisher Vector of clip-level VGG Face features after feature selection; 6. Fisher Vector of all MFCC features; 7. Fisher Vector of clip-level MFCC features.

Another baseline we explore is to consider every clip-level feature as a point in the dataset, and to assign each clip the label of the player this clip belongs to. To generate player-level predictions, we perform inference for every clip and average clip-level predictions. The highest AUC we achieve using VGG Face is 0.55, which supports the claim that for deceptive behavior detection it is necessary to consider video as a whole.

Fisher Vector (FV) is better than the above baselines, achieving an AUC of 0.584. We attribute this to the fact that FV captures statistical information from the whole video rather than from a short clip.

Finally, LiarRank of Fisher Vector of VGG Face feature obtains the highest 0.663 AUC after feature selection (FS), and this improvement is statistically significant ($p < 0.01$). To verify that improvement comes from the proposed meta-feature and not merely from feature selection procedure, we perform feature selection on Fisher Vector of VGG Face (base feature for LiarRank in our experiments), which achieves the highest AUC of 0.522. This experiment suggests that LiarRank is important for the improvement in accuracy.

Histogram features. As baselines we use mean values of Amazon Rekognition features, Facial Action Units and Eye/Head movement features over all the frames in a video. Although some of these baselines (0.586 AUC for Amazon Rekognition, 0.6 AUC for Facial Action Units and 0.5 AUC for Eye/Head movement) outperform

Amazon Rekognition					
Frame hist.		Clip hist.		Combined	
Disgusted, Surprised	0.630	Smile, Angry, Disgusted	0.634	Smile, Angry, Disgusted	**0.676**
Surprised	0.622	Smile , Angry	0.623	Smile, Disgusted	0.647
Calm	0.622	Smile, Disgusted, Calm	0.618	Angry	0.638
All features	0.557	All features	0.544	All features	0.563
Facial Action Units					
Frame hist.		Clip hist.		Combined	
AU07+AU10+AU12	**0.621**	AU06+AU14	0.609	AU07+AU09+AU10	**0.621**
AU12+AU23+AU25	0.614	AU07+AU09+AU10	0.606	AU07+AU10+AU23	0.617
AU09+AU10+AU12	0.612	AU07+AU14+AU45	0.603	AU12+AU25	0.611
All features	0.592	All features	0.577	All features	0.608
Eye/Head movement					
Frame hist.		Clip hist.		Combined	
3+8	0.632	1+6+8	**0.671**	1+3+4+5+6+8	0.643
3	0.624	1+6	0.642	1+3+5+8	0.627
3+7	0.615	1+3+6+8	0.636	1+3+5+6+8	0.625
All features	0.591	All features	0.560	All features	0.618

Table 3.2: Performance (AUC) of histogram based representations: top three subsets and all features for frame-level histograms, clip-level histograms, and combined histograms. In all cases sets of all features perform worse than proper subsets due to excessive noise introduced by irrelevant features. For Action Units numbers refer to FACS [46]. Movement features encoding is the following: 1–2: horizontal and vertical eyes movements, 3–5: Euler angles of head rotations, 6–8: x, y, z head translations.

VGG Face baselines, they are significantly inferior to histogram-based player-level features based on corresponding frame-level features.

Each aforementioned frame-level representation consists of several features corresponding to individual emotions or facial expressions, not all of which are useful for the task of deception detection. To address this problem, we perform cross-validation with exhaustive search through all possible combinations of features within every representation. So, when computing histogram vectors, we concatenate histograms of a subset of features.

Table 3.2 shows that different ways of producing histograms (from frame-level features and from clip-level features) perform differently not just in terms of classification performance but also in terms of best subset of features. In case of Amazon Rekognition features and Facial Action Units, it is advantageous to use combined histogram features. For Eye/Head movement features, however, clip-level histograms yield the best performance.

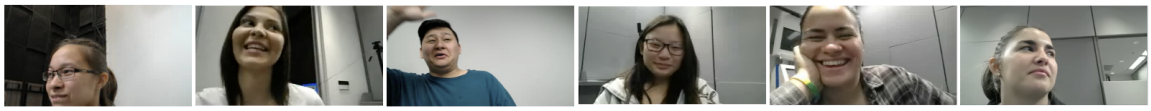


Figure 3.3: Examples of facial expressions (left to right): Angry, Smile, Disgusted as detected by Amazon Rekognition service, AU07 (lid tightened), AU09 (nose wrinkled), AU10 (upper lip raised) as detected using the OpenFace library.

Our experiments show that for Amazon Rekognition based features, the combination of three expressions "Smile", "Angry", and "Disgusted" performs the best and achieves 0.676 AUC. For Facial Action Units, the combination of AU07 (Lid Tightener), AU09 (Nose Wrinkler), and AU10 (Upper Lip Raiser) achieves 0.621 AUC. The combination of horizontal eyes movements and x, z head translations achieves 0.671 AUC. Examples of top-performing facial expressions are shown in Figure 3.3 (except for movement samples as those are dynamic). In all cases representations including all the individual feature histograms ("All features" in Table 3.2) perform worse than some of the subsets.

Classifiers	AUC	F1	FNR	FPR	Precision	Recall
LR+RF+NB+L-SVM+NB	**0.705**	0.466	0.621	0.142	0.666	0.379
LR+L-SVM+NB+L-SVM+NB	**0.705**	0.466	0.610	0.169	0.660	0.390
KNN+RF+NB+RF+NB	0.704	0.403	0.673	0.173	0.622	0.327
NB+L-SVM+NB+L-SVM+NB	0.704	0.406	0.667	0.151	0.624	0.333
LR+KNN+NB+L-SVM+NB	0.704	0.468	0.620	0.143	0.684	0.380

Table 3.3: Performance (AUC) of Top 5 ensemble models. Classifiers in the table are trained on the features in the following order: histograms of AU07, AU09, AU10; Fisher Vectors of MFCC; histograms of Smile, Angry, Disgusted; histograms of horizontal eyes movement, x and z head movement; LiarRank of VGG Face Fisher Vector.

3.2.2 Ensemble Prediction and Feature Importance

For our ensemble classifier, we use five best performing features: histogram features of facial action units (AU07, AU09, AU10), Fisher Vectors of MFCC, histogram features of Amazon Rekognition predictions (Smile, Angry, Disgusted), histogram features of best movement feature combinations in Table 3.2 and LiarRank of VGG Face Fisher Vector. Since for single-feature experiments we use a number of classifiers, we perform exhaustive search through all possible combinations of classifiers for the mentioned

features. Once single-feature classifiers are trained, we perform late fusion using grid search as described in the Section 3.1.3. Table 3.3 shows our Top-5 ensemble prediction results, including what classifiers were used for the corresponding features. Our best predictive models yield an AUC of 0.705.

To assess the importance of features for the ensemble classifier, we repeated the process leaving out one class of features at a time. We show the results of this ablation experiment in Table 3.4. We can see that LiarRank of VGG Face Fisher Vectors and the Emotion (Amazon Rekognition) histogram features are the most important as the performance drops the most when they are absent.

Removed feature	AUC	F1	FNR	FPR	Precision	Recall
MFCC	0.703	0.463	0.610	0.175	0.655	0.390
E/H Movement	0.703	0.508	0.548	0.197	0.599	0.452
FAUs	0.702	0.448	0.598	0.209	0.587	0.402
Amazon Rekognition	**0.688**	0.524	0.485	0.281	0.556	0.516
LiarRank	**0.688**	0.411	0.344	0.721	0.104	0.560

Table 3.4: Classification performance (AUC) when one feature class is left out in ensemble predictions. Features details are in Table 3.3.

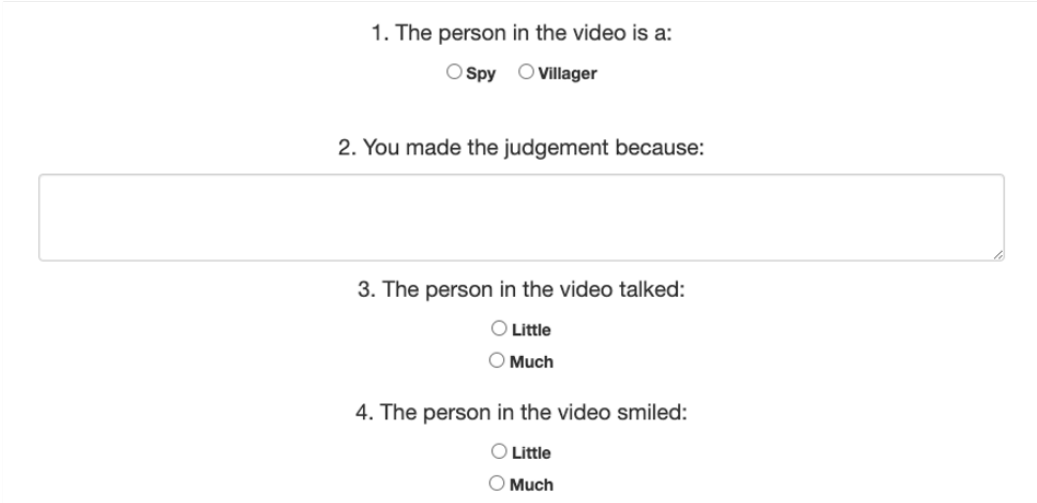
3.2.3 Human Study

To assess the complexity of the task and obtain some objective baseline we conducted a human study using the Amazon Mechanical Turk service[1]. To provide a fair comparison, we presented workers with the same data we are using for testing our model: we stitched 10-second clips together with a 1 second transition between them keeping the sound on. Workers were provided with a brief description of the game they were about to watch as shown in Figure 3.4a and asked to make a decision whether the player in the video was a spy or a member of the resistance. We programmatically ensured that the subjects cannot submit the answers until they finish watching the video. To further verify the quality of annotations, workers were asked to provide written justification for their decision and answer two questions about the content of the video as shown in Figure 3.4b.

We selected 10 games containing 66 videos in total, and got every video annotated by 3 different workers. Correct player's role was guessed by a majority (2–3 workers out of 3) only in 53% of videos. We also used the average vote of turkers as a prediction

[1] https://www.mturk.com/

(a)



(b)

Figure 3.4: Setup of the human study. Instructions explained what subjects should do and the rules of the game subjects were about to watch (a). A set of questions was designed to ensure the subjects watched the entire video carefully (b).

Spies	Villagers
facial expressions, visible emotions, looking nervous	did nothing suspicious
lack of interest, lack of participation, looking bored	admitted to be a villager, denied being a spy
staying quiet, talking little	distrusted others, accused others of being spies
observed others, tried to remain in the background	team player, tried helping out
laughed or smiled nervously, laughed with others	clueless, not interested, confused
acted guilty, acted sucpiciously	casual, laid back, funny, friendly
smiling, laughter	quiet, focused, observing the game
body language	body language
became sad when villagers win, smiled when spies win	happy when team succeeds

Table 3.5: Human study: most popular justifications for decisions sorted by popularity.

score for the video. In this case, the AUC for human prediction is 0.583, while our ensemble predictor gets 0.701 AUC for the same data ($p < 0.01$). This suggests that detecting deception in long videos is a hard task for humans. We also found that in more than 80% of the videos, players were suspected to be spies when the actual ratio of spies in the dataset was 42%. This means that humans, when presented with the fact that a player could be a spy, tend to interpret a player's behavior as suspicious. We also present the most common justifications annotators gave for their judgments in Table 3.5. We can see that the range of clues that humans pay attention to is wide sometimes including mutually contradictory observations: some annotators thought smile and laughter is a sign of a spy, others thought it is a sign of a villager, some turkers attributed quiet behavior to attempts at staying stealthy while others connected quietness with attempts at observing others and figuring out who are the spies.

3.3 Conclusion

We presented an ensemble based automated deception detection framework called LiarOrNot which predicts deception in a group setting by processing long videos. Our framework utilizes appropriate representations at different temporal resolutions for multiple features which capture low and high level information. We also propose a novel class of meta-features called LiarRank which provide a significant boost in overall performance. By evaluating LiarOrNot on a dataset which was collected across different sites, in a rigorous cross-validation based testing protocol which separated identities and games during training and inference, we obtained an AUC greater than 0.7, which was 12% better than average humans.

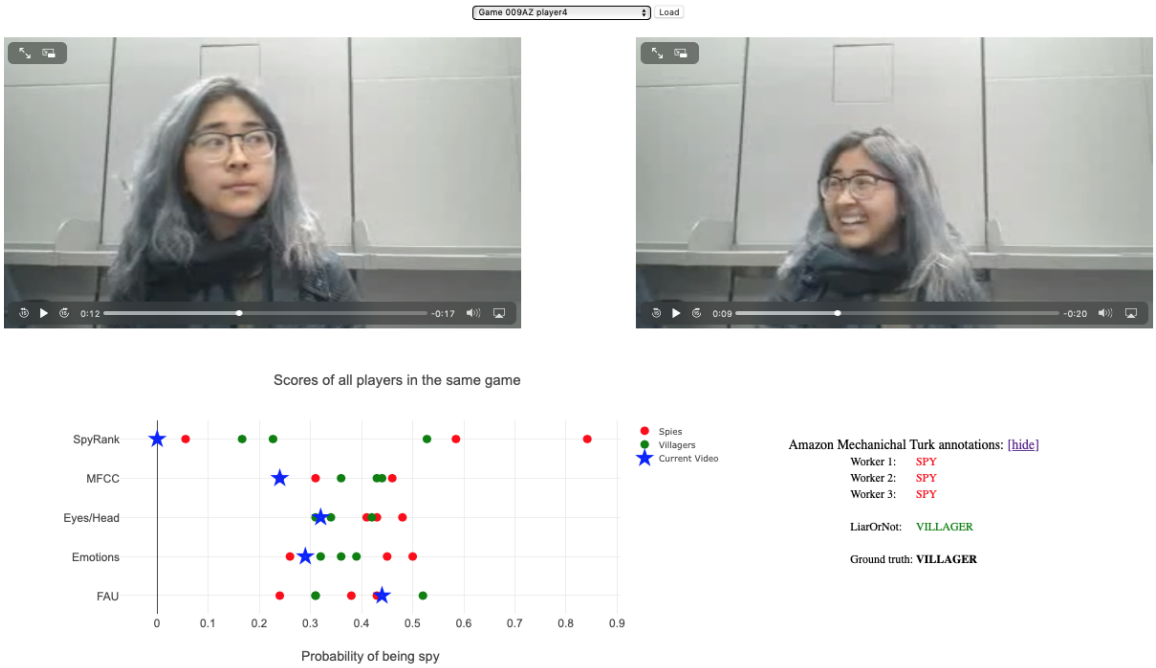


Figure 3.5: Screenshot of LiarOrNot demo. We demonstrate two 30-second clips from the Resistance dataset and show how this player's predicted deception scores compare to other players in the same group.

In addition to that we have built a prototype LiarOrNot system demonstrating the performance of our methods. In our demo[1], we opted to show several players from our dataset. For each player we demonstrate two 30-second samples of video. We visualize the predicted probabilities of the player being a spy from our best five individual predictors (Figure 3.5). We compare how these predictions correspond to prediction for the rest of the players in the same game. Additionally, we show AMT workers' opinions on the player's role, as well as our system's final prediction and the ground truth role.

[1] https://www.cs.dartmouth.edu/~mbolonkin/liar-or-not/demo/

Chapter 4

Political deception detection

We often put trust in the elected officials we vote for. But is this trust warranted? On their campaign trails politicians make a lot of statements carefully crafted to elicit support from their base. However, for a regular person distinguishing a truthful statement from a statement that just looks truthful sometimes is a hard task without proper research. With the spread of misinformation through social and mass media the need in fact-checking resources significantly increased. Such fact-checking enterprises as PolitiFact or FactCheck.org analyze the content of the statements made by politicians and determine whether the statements are true or false.

This, however, begs the question: why not apply automated deception detection methods to this problem? There is a significant number of research papers on deception detection from audio [101, 54], text [67, 96], and video [122, 158, 3, 134]. Therefore, we consider the following task: given a video of a political figure making a public statement predict whether the statement is truthful or deceitful.

In addition to general deception detection challenges, his task has its own:

- First, there is an obvious lack of publicly available multimodal datasets for this task: the sole available dataset [100] contains only text and restricts itself to only US politicians,
- Second, it is obvious that the veracity of the statement may depend on the subject area of that statement. For instance, it is possible that a politician will be honest about some topics but less honest about more controversial topics such as aspects of his or her past private life, his or her medical or financial records, and so forth. Still, it is not immediately clear how to analyze the content in the broader context.

In an attempt to overcome these challenges, we make the following novel contributions:

1. We collect a dataset of public figures from several countries worldwide making truthful and deceptive statements. We call this dataset the Global Political Deception Dataset. It is the first multimodal dataset of its kind with subjects with subjects from a wide array of countries;

2. We propose a novel graph with nodes representing videos of politicians, topics of the messages, and news articles about those politicians: this method allows putting the content of the analyzed statement into a broader context and assess how likely it is to be deceptive;

3. We develop a novel class of features we call Deception Score that brings together intrinsic properties of the video (how likely it is to be deceptive) with the assessment of how likely the message from the video to be deceptive;

4. We show that our proposed Deception Score in conjunction with basic features greatly improve upon basic features alone and a comprehensive set of baselines.

4.1 Global Political Deception Dataset

We aim at building a multimodal dataset of real-life examples of political deception. There is a noticeable lack of high-stakes deception datasets. Apart from the widely used Real-Life Trial dataset [122], text-based LIAR dataset for fake news detection [154], and text-based CT-CWC-18 corpus [100], there is only one relevant dataset containing videos of United States politicians making true or false statements [76], but as of the writing of this dissertation the dataset is not publicly available yet. Moreover, the latter dataset is focused on public figures from a single country. We want to create a collection of examples of political deception across different countries and different languages. We call our new dataset the Global Political Deception Dataset.

Dataset collection. Overview of the dataset creation process is shown in Figure 4.1. The primary task was to search for videos of political figures making statements verified by one or more fact-checking resources as false. Examples of such resources are "Ojo Publico"[1], "Africa Check"[2], "ABC News"[3], and others. We col-

[1] https://ojo-publico.com/
[2] https://africacheck.org/
[3] https://www.abc.net.au/

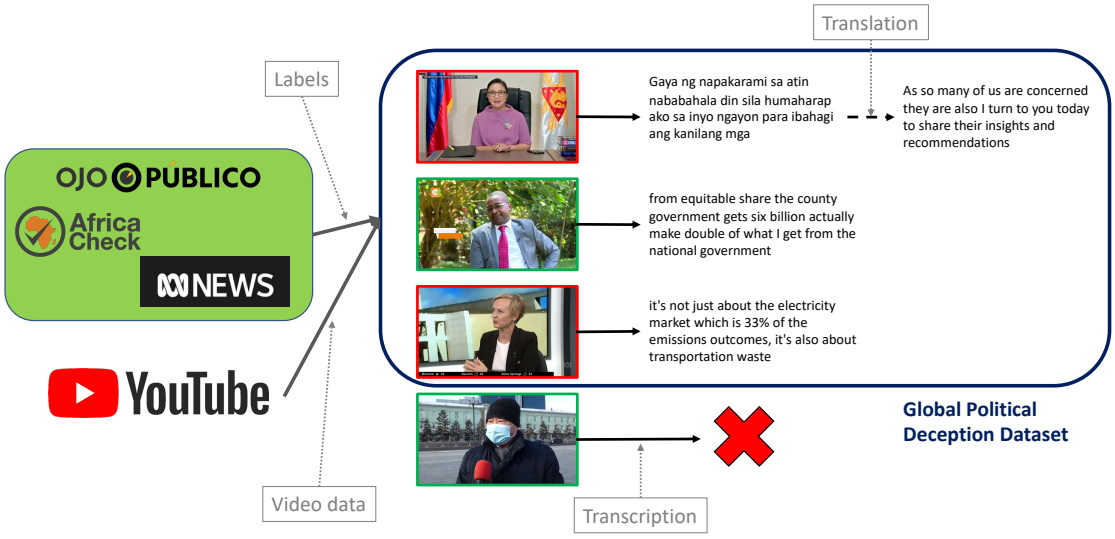


Figure 4.1: Global Political Deception Dataset creation process. We first find videos of politicians making statements fact-checked to be true or false. Then we automatically transcribe the videos. For videos in languages other than English we perform automated translation. Collected videos, transcripts, and translations form our dataset.

lected a total of 104 videos depicting politicians and public figures in various settings such as political debates, interviews, and public speeches, one per person. Then for each person in the dataset, we collected a video with a true statement verified the same way. We primarily collected the videos from YouTube, with some videos coming from social networks such as Facebook or directly from news agencies' websites. We also acknowledge that despite calling the statements truthful and deceitful, there is no evidence to conclude that the public figure making that statement is knowingly lying. Sometimes public figures can make false statements because of the sincere belief in it or because of the lack of knowledge on the topic. We, however, use the fact-checked labels as a proxy to deception similar to other works on political deception [84, 76].

After initial collection, we discard the videos that were very noisy or did not have the face of the politician. We then extracted automatically generated transcripts. Some of the videos were in languages that did not support automated transcription (e.g., Mongolian language). We excluded those videos from the dataset. Overall, we ended up with 148 videos (75 with false statements and 73 with true statements) depicting 76 public figures from 18 countries. If videos and transcripts were in any

Figure 4.2: Examples from the Global Political Deception Dataset. From left to right, top row: Festus Keyamo (Nigeria), Katie Allen (Australia); bottom row: Leni Robredo (Philippines), Mwangi wa Iria (Kenya).

language other than English, we used Google Translate API[1] to translate it into English automatically.

Dataset description. Our dataset contains videos, automated original transcripts, and automated translations of the transcripts into English if the original language is other than English. Videos span lengths from 10 seconds to 1200 seconds with a median length of 30 seconds. Figure 4.2 shows some examples of the videos in our dataset. Table 4.1 shows excerpts of transcripts.

Our dataset contains 76 public figures from 18 countries. Out of them, 14 are female, and the rest are male. This gender bias reflects the general situation in politics across the world. We obtained samples from each region of the world. Retrieval of labeled data and associated textual data for the purposes of this research, however, led to skew in geographic distribution (see Figure 4.3 for the distribution).

[1] https://cloud.google.com/translate

Politician	True statements	False statements
William Ruto (Kenya)	we were told in 2017 that you cannot win an election on the basis over development track record as a government you need some emotional things you need to appeal to emotion you need to appeal to your community you need to appeal to but we said no Kenyans can see for themselves	diaspora is an important component of our development architecture in fact the diaspora is the largest contributor of our foreign exchange I think upwards of maybe *290 billion* last year which is a huge contribution by a community out there that we we haven't reached out sufficiently
Imelda Marcos (Philippines)	you fall in love with the man you marry the minute you were born because she starts again dreaming you're kind of a dream man and suddenly here he comes	well now it's my best defense because when the world went to my closet to email this concept *they did not find skeletons* they find beautiful shoes
Bashar Asad (Syria)	we have fully believed in negotiations and in political actions since the beginning of the crisis however if we negotiate it does not mean that we will stop fighting terrorism	were in the area of the militants we're in the area under the control of the terrorists that were they that were they could accuse first the people or the militant that are responsible of the security of this convoy so *we don't have any idea about what happened*
Veronica Alonso (Uruguay)	what we were saying is just two precandidates but we are I think two candidates with strength with desire and above all we hope to have the most support	we turn on the income of officials let's cut with something look and checking the numbers and doing numbers with the team there are *more than 2500 or more positions of trust* that represents approximately $7 million per month we can not cut it may seem a little for the number but each of those numbers make the final difference

Table 4.1: Excerpts of transcripts of false and true statements. Parts of the false statement that were fact-checked as false are marked in italics. Veronica Alonso's transcript is automatically translated from Spanish.

4.2 Methodology

To approach the problem of deception detection in public statements, we exploit the fact that the statements of interest are usually made about topics of public importance. In other words, there is a significant chance that the same issues were discussed (with or without connection to the public figure that made the analyzed statement) somewhere else, particularly in news articles. This is a novel part of our approach. We discuss these features in more details in the next section followed by an overview of some basic features that we also explored in this work.

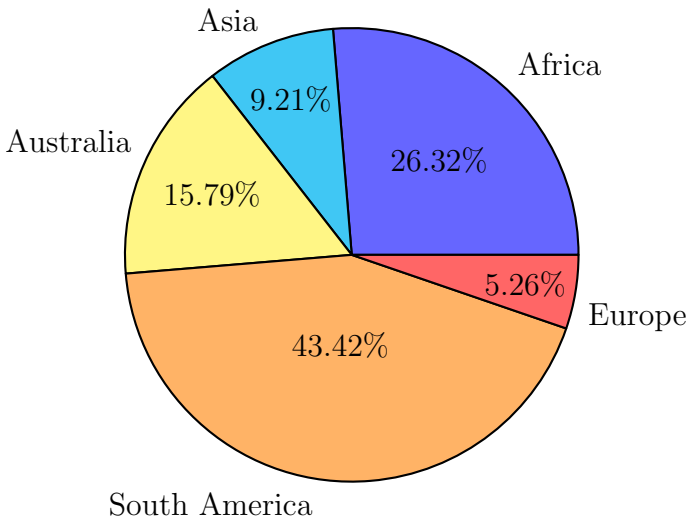


Figure 4.3: Geographical distribution of politicians' countries of origin in the dataset.

4.2.1 Deception scores

To analyze the truthfulness of a statement made by a public figure, we can consider the broad context of the statement. Clearly, some topics may not be worth lying about, and others draw a lot of attention and are prone to misrepresentations or explicit lies. We define the novel concept of a Video-Article-Topic (VAT) graph $G = (V, T, A, E_{VT}, E_{AT})$ to be a tripartite graph. An edge $(v, t) \in E_{VT}$ ($(a, t) \in E_{AT}$ respectively) indicates that a video $v \in V$ (an article $a \in A$ respectively) mentions topic $t \in T$, weighted by $w(v, t)$ ($w(a, t)$ respectively). We define the following node and edge attributes.

Edge anomaly $\delta(a, t)$, $\delta(v, t)$. This value measures how anomalous a given text $a \in A$ is among other texts generated by a given topic $t \in T$ (a transcript of a video $v \in V$ respectively). We define the edge anomaly $\delta(a, t)$ to be the deviation of the weighted sentiment value of the article $s(a)$ from the average weighted sentiment value of all articles generated by the same topic t:

$$\delta(a,t) = \left| w(a,t)s(a) - \frac{\sum_{(a',t)\in E_{AT}} w(a',t)s(a')}{\sum_{(a',t)\in E_{AT}} w(a',t)} \right|. \tag{4.1}$$

In a similar way we define the Video-Topic edge anomaly. These values are meant to represent how different is a particular article or video transcript among those generated by the same topic. If, for example, most of the articles on the given topic express sentiment, then the average sentiment will be positive, and an article with

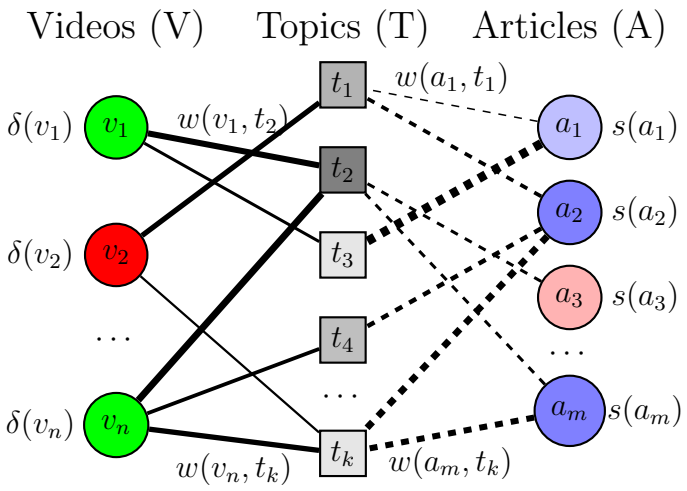


Figure 4.4: Video-Article-Topic (VAT) graph. First, we extract topics (T) from an independent news dataset using Latent Dirichlet Allocation [19]. Then for a set of articles (A), we use obtained LDA model to extract topic assignments. Finally, for each video transcript (V), we estimate the probability of being generated by each of the topics using the same LDA model.

negative sentiment will produce a larger difference in the Equation (4.1). The more different an article in sentiment, the higher value will edge anomaly take.

Topic controversy $c(t)$ of a given topic $t \in T$ is defined as an entropy of normalized sentiment values among all articles generated by this topic:

$$c(t) = H\left(\hat{s}(a,t)\big|_{(a,t)\in E_{AT}}\right), \tag{4.2}$$

where $\hat{s}(a,t) = \frac{w(a,t)s(a)}{\sum_{(a,t)\in E_{AT}} w(a',t)s(a')}$, and $H(\cdot)$ is the entropy function. There is the following intuition behind this definition. A topic with a single shared view on it is hardly controversial. Thus entropy of a set of almost equal values will be low. On the other hand, if the sentiments about the topic are uniformly spread over all possible values, that would mean that there are many different opinions on the topic, and the entropy will be relatively high.

Finally, we define the video deception score $VD(v)$ for each video $v \in V$, topic deception score $TD(t)$ for each topic $t \in T$, and an edge deception score $D(v,t)$ for each edge $(v,t) \in E_{VT}$. The intuition behind these scores is to represent the odds of a given video to deceive on a given topic, as well as odds of the given video being deceptive, and a given topic to have a high chance of generating false statements.

We want the edge deception score $D(v,t)$ to be high if both video and topic have high

chances of being deceptive, and the edge (v, t) itself presents itself as anomalous.

$$D(v,t) = \frac{\delta(v,t) + \gamma_1 VD(v) + \gamma_2 TD(t)}{1 + \gamma_1 + \gamma_2} \tag{4.3}$$

Video v must have high deception score when it has a substantial number of edges with high deception scores:

$$VD(v) = \frac{\delta(v) + \alpha \sum\limits_{(v,t)\in E_{VT}} w(v,t) D(v,t)}{1 + \alpha \sum\limits_{(v,t)\in E_{VT}} w(v,t)}, \tag{4.4}$$

where $\delta(v)$, the anomaly of a video v is the intrinsic property of the video.

Lastly, the deception score of a topic t is related to its controversy $c(t)$, lying scores of incoming edges $D(v, t)$ and anomaly of incoming edges $\delta(a, t)$.

$$TD(t) = \frac{1}{\Phi(t)} \left(c(t) + \beta_1 \sum_{(v,t)\in E_{VT}} w(v,t) D(v,t) + \beta_2 \sum_{(a,t)\in E_{AT}} w(a,t)\delta(a,t) \right), \tag{4.5}$$

where Φ is the normalizing factor:

$$\Phi(t) = 1 + \beta_1 \sum_{(v,t)\in E_{VT}} w(v,t) + \beta_2 \sum_{(a,t)\in E_{AT}} w(a,t).$$

In equations 4.3, 4.4, and 4.5 numbers $\alpha, \beta_1, \beta_2, \gamma_1, \gamma_2$ are the hyper-parameter weights determining importance of corresponding summands in the equations.

In the end, we have a set of recurrent equations that we can solve iteratively or with a closed-form solution. For the closed form solution, we first substitute Equation (4.3) into Equation (4.4) and Equation (4.5). After simplifying, we have a system of equations:

$$\begin{bmatrix} \mathbf{A_{11}} & \mathbf{A_{12}} \\ \mathbf{A_{21}} & \mathbf{A_{22}} \end{bmatrix} \begin{bmatrix} \mathbf{VD} \\ \mathbf{TD} \end{bmatrix} = \begin{bmatrix} \mathbf{B_1} \\ \mathbf{B_2} \end{bmatrix}. \tag{4.6}$$

In this equation

$$\mathbf{VD} = [VD(v_1), VD(v_2), \ldots, VD(v_N)]^T$$

and

$$\mathbf{TD} = [TD(t_1), TD(t_2), \ldots, TD(t_K)]^T$$

are vectors of video and topic deception scores respectively (with N and K being number of videos and number of topics respectively). Matrix $\mathbf{A_{11}}$ is a diagonal matrix

with elements

$$A_{11}(i,i) = \frac{\alpha\gamma_1}{(1+\alpha)(1+\gamma_1+\gamma_2)} \sum_{(v_i,t)\in E_{VT}} w(v_i,t) - 1.$$

Matrix $\mathbf{A_{22}}$ is a diagonal matrix with elements

$$A_{22}(i,i) = \frac{\beta_2\gamma_2}{\Phi(t_i)(1+\gamma_1+\gamma_2)} \sum_{(v,t_i)\in E_{VT}} w(v,t_i) - 1.$$

Matrix $\mathbf{A_{12}}$ has elements

$$A_{12}(i,j) = \begin{cases} \frac{\alpha\gamma_2 w(v_i,t_j)}{(1+\alpha)(1+\gamma_1+\gamma_2)}, & \text{if } (v_i,t_j) \in E_{VT} \\ 0, & \text{otherwise.} \end{cases}$$

Similarly we define matrix $\mathbf{A_{21}}$:

$$A_{21}(i,j) = \begin{cases} \frac{\beta_1\gamma_1 w(v_j,t_i)}{\Phi(t_i)(1+\gamma_1+\gamma_2)}, & \text{if } (v_j,t_i) \in E_{VT} \\ 0, & \text{otherwise.} \end{cases}$$

We can solve the Equation (4.6) using one of the matrix decomposition algorithms. The solution will be the fixed point of Equations (4.3)–(4.5).

Iterative way to find the fixed point is to choose some initial values for $VD(v_i)$ and $TD(t_j)$, substitute into the equations (4.3)–(4.5), find the updated values of these variables, and keep repeating this process until convergence.

Having all values for $D(v,t)$ we can form a feature vector for a given video

$$[D(v,t_1),\ldots,D(v,t_K)]\,,$$

where K is the number of topics.

Deception scores can differ based on hyper-parameters weights or the video anomaly $\delta(v)$ used in the equation (4.4). This anomaly value can be any scalar or vector representing how anomalous the video is on its own. We used the classification scores obtained from several baseline models.

4.2.2 Basic features

Most of the deception researchers agree that an act of lying can have "leaks", cues to the internal state of the deceiving person [25]. Thus, most deception detection methods are trying to detect those cues, whether via physiological parameters (heart rate, sweating levels, electric conductivity, etc.) or from changes in facial expressions or speech (changes in pitch, speed of uttering words, linguistic patterns).

The Global Political Deception Dataset contains videos, audios, and transcripts. We therefore consider a set of features for each of these modalities. These features were also used in one or more previous works on deception detection.

Visual features. To obtain representations of the video content, we first extract the Facial Action Units (FAUs) from the recordings using an off-the-shelf software OpenFace [12]. We then calculate histograms of these intensities with a fixed number of bins as in Chapter 3.

Audio features. There are two types of audio features that we use in this work.

- *Mel-Frequency Cepstral Coefficients* [35] which captures lower-level information about the sound wave.
- Second type is a set of individual features extracted by OpenSMILE software [49]. These features were used in a number of papers and showed their efficiency for speech analysis [84, 76, 106].

Text features. To analyse the transcripts we use the following text and linguistic features:

- *BERT embeddings.* Transformer architectures have shown great success in recent years and became *de facto* the standard tool for text analysis [41, 145]. For all texts we extract `[CLS]` token embedding from a pretrained BERT-Base, Uncased model [41]. Whenever a text is longer than 512 tokens, we use head+tail truncation method as described in [139]: we leave the first 128 and the last 382 tokens.
- *TF-IDF* vector [72] for unigrams, bi-grams, tri-grams, and four-grams.
- *LIWC feature vector.* We used LIWC 2007 and LIWC 2015 software [140] to extract frequencies of word occurrence from 64 and 90 categories respectively.

4.3 Experimental results

We test our method against a set of baseline methods on the Global Political Deception Dataset.

4.3.1 Baselines

We compare our methods to a number of baselines from the relevant literature.

1. Uni-modal features from [158]: Fisher Vectors of Improved Dense Trajectories (iDT), Fisher Vectors of MFCC, Fisher Vector of Glove representations of the transcripts.

2. Facial Action Units histograms and Head/Eye movement histograms from [9]: we use the best frame-based features as in our dataset we do not have long enough videos to justify splitting into 10-second clips. We use the combinations that shown the best results on the Resistance dataset [9], and we also perform an exhaustive search for the best combination as described in the same work.

3. Temporal Convolutional Network (TCN), Base Network trained on the time-series of FAU intensities from [138].

4. Acoustic, Linguistic, and Visual features as well as their combinations used for political deception detection in [76].

5. Textual (LIWC, TF-IDF, and BERT), Acoustic (ComParE) features, and their ensemble used to detect deception in political debates by [84].

4.3.2 Experimental setup

We split the Global Political Deception Dataset into ten approximately equal folds trying to keep the balance of classes. We ensured that videos of the same public figure do not fall into different folds. Our experiments use a battery of standard classifiers: k-Nearest Neighbor, Logistic Regression, Gaussian Naive Bayes, and Random Forest. We use mean AUC and F1-scores over ten folds as the performance metric. We test our proposed feature types with each classifier as mentioned above and report the best metric for any given class of features among all used classifiers.

For baseline features, we use the same classifiers as described in each corresponding paper: decision tree classifier for [76], Logistic Regression for [84], a battery of classifiers as described above for [9, 158]. We implemented the Base Network TCN [138] according to the description from the paper. We used the same training/evaluation protocol as in the original work.

4.3.3 Building VAT graph

To build the VAT graph, we need to mine news articles, extract topics, and find Article-Topic associations.

For that, we collected a set of news articles using Nexis Uni[1] news sources database. For each video in our dataset, we mined news articles that mentioned the public figure from the video and dated from no more than a month before the video release. We also manually checked the articles to avoid duplicates of the same news story. All pieces were in English. We ended up with 6337 articles with a minimum of 1, a maximum of 104, and an average of 43 articles per video. We share these articles as part of the Global Political Deception Dataset.

To extract topics, we chose a generic set of topics over a set specific for our set of articles. In other words, we decided to extract topics specific for news texts in general rather than for our limited set of news pieces. This choice would ensure that both articles and transcripts are associated with at least one topic in the VAT graph. To extract these topics we used a large news dataset "All the news"[2]. This dataset contains about 143,000 articles scraped from various news websites such as the New York Times, CNN, Fox News, Buzzfeed News, Reuters, the Washington Post, and others. The publication date of the news pieces from this dataset falls between July 2016 and July 2017. These texts allow us to extract a sufficiently rich set of topics.

We use Latent Dirichlet Allocation (LDA) [19] to extract topics from the "All the news" dataset. LDA is widely used for topic modeling of large corpora. We produce two topic models: with 50 and 100 topics. Table 4.2 shows some examples of extracted topics: each topic is represented by a distribution of words that generate this topic. Some of the topics are very similar with a possible change in word importance; some topics look drastically different when we allow for more topics to be formed.

[1] https://www.lexisnexis.com/en-us/professional/academic/nexis-uni.page
[2] https://www.kaggle.com/snapcrack/all-the-news

LDA, 50 topics	LDA, 100 topics
water, area, city, people, mile, land, national, road, storm, coast	water, storm, area, coast, mile, land, weather, snow, river, region
drug, health, doctor, patient, medical, disease, hospital, cancer, treatment, virus	drug, doctor, patient, medical, hospital, cancer, treatment, health, disease, death
police, officer, gun, shooting, shot, department, video, law, violence	church, christian, religious, god, faith, catholic, religion, marriage, pope, francis
brazil, athlete, olympic, rio, gold, olympics, world, sport, brazilian, fashion	brazil, cuba, castro, government, cuban, president, country, brazilian, corruption, latin
islamic, syria, state, force, syrian, group, city, iraq, militant	syria, syrian, air, rebel, aleppo, force, war, strike, assad, government

Table 4.2: Examples of topics extracted by LDA (models with 50 and 100 topics). We show up to 10 words that generate the corresponding topic. Words are sorted in the order of decreasing probability for the given topic. Some of the topics produced by the two models look similar with minor differences in word importance, while others look drastically different.

We then use the trained LDA model to estimate the probability of each article and each video transcript to be generated by these topics (which corresponds to the edge weights $w(a,t)$ and $w(v,t)$ respectively).

4.3.4 Deception scores performance

To calculate the edge anomaly values and topic controversy, we need to find the sentiments of the articles and video transcripts. We use XLNet (large-uncased) transformer model [160] pre-trained on a Stanford Sentiment Treebank dataset [137] for the task of sentiment analysis. This dataset has two levels of label granularity: binary labels (positive vs. negative) and 5-class labels from very negative (1) to very positive (5). Once trained, the model outputs the probability of sentiment scores for each sentence in the text (article or transcript). We calculate the final sentiment of a text by averaging weighted mean sentiment over all sentences in the text (weights being the estimated probabilities).

For video anomaly score $\delta(v)$ we use deception probability scores produced by several baseline features: FAU histogram and Head/Eye movement histogram that produced the best result on the Resistance dataset (Chapter 3). These values are calculated from video alone.

Metric	Feature type	LDA-50		LDA-100	
		Sent-2	Sent-5	Sent-2	Sent-5
AUC	DS Vector	0.666	0.579	**0.714**	0.684
	DS Histogram	0.632	0.630	0.642	0.639
F1-score	DS Vector	0.628	0.629	0.654	0.653
	DS Histogram	0.663	**0.700**	0.644	0.667

Table 4.3: Performance of our features. We show results for two metrics (AUC and F1-score) for the best performing features for each of the features types, topic models, and sentiment granularity. The difference between highest values and the rest of the models is statistically significant ($p < 0.01$).

After the deception scores are calculated, we can build a vector representation (DS Vector) for each video $v \in V$ as follows: $[D(v, t_1), \ldots, D(v, t_k)]$ for all topics $t \in T$. Thus, we have features of lengths 50 and 100. We also build feature vectors by calculating histograms (DS Histogram) of video deception scores $VD(v)$ calculated from different hyper-parameter weights $\alpha, \beta_1, \beta_2, \gamma_1, \gamma_2$. We use 100 bins for these histograms.

Table 4.3 shows the performance of the best features for each of the two topic models, two kinds of sentiment granularity, and two ways of building feature vectors.

We can see that Deception Scores calculated from LDA model with 100 topics always outperforms models with 50 topics. Models that used binary sentiment (Sent-2) work better than models with fine-grained sentiment predictions if we use AUC as a metric, but situation is reversed when we use F1-score instead.

4.3.5 Individual features comparison

Table 4.4 shows the performance of our individual features against baseline and basic features.

We can see that although basic TF-IDF feature performs the best according to both metrics, our features are producing the second best result according to F1-score metric and the third best result according to AUC, outperforming all other baselines and staying close behind the textual features from [84].

Another observation from the Table 4.4 is that aggregation methods and classifiers used to train are no less critical for the performance than the features themselves. For

Source	Modality	Feature	F1-score	AUC	Accuracy
Ours		DS Vector, LDA-100, Sent-2	0.636	0.714	0.628
		DS Histogram, LDA-50, Sent-5	0.700	0.540	0.541
Wu et al. [158]	Visual	iDT Fisher Vector	0.513	0.537	0.541
	Acoustic	MFCC Fisher Vector	0.588	0.514	0.554
	Textual	Glove Fisher Vector	0.647	0.660	0.635
Chapter 3	Visual	FAUs Histogram (Resistance)	0.650	0.571	0.534
		Head/Eye movement Histogram (Resistance)	0.568	0.569	0.568
		FAU Histogram (our data)	0.680	0.648	0.628
		Head/Eye movement Histogram (our data)	0.634	0.629	0.601
Stathopoulos et al. [138]	Visual	Base Network TCN	0.463	0.563	0.540
Kamboj et al. [76]	Visual	Emotions	0.427	0.527	0.481
		FAUs	0.471	0.464	0.467
		Gaze	0.513	0.513	0.517
		Pose	0.496	0.492	0.491
		Vis Comb 1	0.494	0.526	0.527
		Vis Comb 2	0.609	0.593	0.588
		Vis Comb 3	0.489	0.513	0.510
	Linguistic	Glove	0.404	0.456	0.462
		LIWC 2015	0.556	0.594	0.595
		Polarity	0.500	0.508	0.512
		POS	0.556	0.549	0.550
		Unigrams	0.599	0.585	0.586
		Ling Comb 1	0.510	0.495	0.502
		Ling Comb 2	0.470	0.482	0.485
		Ling Comb 3	0.500	0.548	0.548
	Acoustic	IS09	0.530	0.533	0.534
		IS13	0.565	0.581	0.581
		IS09+IS13	0.552	0.519	0.520
	Ensemble	Fusion-All	0.531	0.514	0.517
		Fusion1-2M	0.573	0.568	0.566
		Fusion1-3M	0.543	0.532	0.535
		Fusion2-2M	0.491	0.525	0.529
		Fusion2-3M	0.504	0.514	0.516
		Fusion3-2M	0.472	0.504	0.508
		Fusion3-3M	0.546	0.561	0.561
Kopev et al. [84]	Textual	BERT	0.663	0.718	0.635
		LIWC 2007	0.570	0.579	0.581
		TF-IDF	**0.718**	**0.722**	0.624
	Acoustic	ComParE	0.545	0.489	0.482
	Ensemble	Probability Avg.	0.621	0.668	0.613

Table 4.4: Individual features comparison. Refer to the corresponding papers for the relevant description of individual features and combination details. The main metrics of interest are ROC AUC and F1-score; accuracy is presented for reference only. TF-IDF, BERT, and Deception Score features are significantly better than the rest of the features ($p < 0.01$). Differences between these top-3 features are statistically significant ($p < 0.05$).

Our features	Basic Features
DS Vector, LDA-100, Sent-2	TF-IDF [72, 84]
DS Vector, LDA-100, Sent-5	BERT [41, 84]
DS Vector, LDA-50, Sent-2	Probability Avg. [84]
DS Histogram, LDA-100, Sent-2	Glove Fisher Vector [115, 158]
DS Histogram, LDA-100, Sent-5	

Table 4.5: Top-performing features used for ensemble models sorted in decreasing order of performance (AUC).

example, both Kamboj et al. [76] and Wu et al. [158] use Glove word representations. The former simply average the vectors over the text and use Random Forest classifier, the latter use Fisher Vector aggregation and Logistic Regression as classifier. These differences lead to more than 20% improvement.

4.3.6 Ensemble Models

We use the best features produced by our method and top performing basic features to build an ensemble using late fusion (Table 4.5). If S_i is the score returned by a classifier for the ith feature type for $i \in \{1, \ldots, K\}$, where K is the total number of features in the ensemble, then the final score S is obtained by late fusion of named models:

$$S = \sum_{i=1}^{i=K} \alpha_i S_i \ ,$$

where $\sum_{i=1}^{K} \alpha_i = 1$. Late fusion weights α_i are obtained by grid-search and cross-validation. Due to computational reasons we choose K to be no more than 7.

Table 4.6 shows the results of the late fusion experiments. First, we note that the best performing ensemble (consisting of the best two basic features and best three our features) significantly improves upon the best individual feature from Table 4.4. This ensemble yields 5.3% higher AUC than TF-IDF feature [84] (with $p < 0.01$). Second, we can see that the best ensemble that consists only of basic features yields 4.6% lower AUC than the best performing ensemble. Moreover, for the sets of Top-2, Top-3, and Top-4 basic features adding one or more of our proposed features into the ensemble almost universally leads to improved performance. This fact demonstrates that our Deception Score features are complementary to unimodal representations.

	Basic features				
Our features	No features	Top-1	Top-2	Top-3	Top-4
No features	-	0.722	0.729	0.726	0.692
Top-1	0.714	0.712	0.749	0.739	0.761
Top-2	0.671	0.726	0.753	0.756	0.751
Top-3	0.692	0.697	**0.775**	0.754	0.761
Top-4	0.684	0.697	**0.775**	0.754	0.751
Top-5	0.669	0.697	0.773	0.751	—

Table 4.6: Late fusion results (AUC). Each number represents a performance (AUC) of an ensemble of models consisting of certain number of basics features and certain number of our proposed features (cf. Table 4.5).

4.4 Conclusion

In this chapter, we considered the task of predicting deception in videos of political figures making public statements. First, we collected a dataset for this task. This Global Political Deception Dataset contains videos of public figures from 18 countries across the world making true and false statements as verified by fact-checking resources. We then proposed a novel class of features we call Deception Score that aimed at analyzing the content of the analyzed statement by putting it in a broader context of texts about similar topics.

We tested the Deception Score features on our Global Political Deception Dataset and demonstrated that they perform on par with the best baselines features and outperform most of the other baselines. We then used our best features and the best basic features to build ensemble models that even further improve the system's performance. We showed that the best ensemble outperforms the best individual feature by 5.3%. We also showed that adding Deception Score features to basic features consistently improves the ensemble performance. This means that Deception Score features are complementary to the textual and other basic features.

Chapter 5

DIPS: A Dyadic Impression Prediction System for Group Interaction Videos

There are many situations in group settings where we wish to understand the impressions that a person p_i has of another person p_j. For instance, in a diplomatic negotiation, it might be critical for one side to understand the mutual feelings of people on the other side toward one another as this can provide important leverage. A person called in to a business meeting with a group of people she doesn't know might wish to understand the like/dislike relationships between the people she is meeting with.

In this chapter, we consider the problem of *dyadic impression prediction* using nonverbal cues. Specifically, we would like to use nonverbal cues such as facial action units [12], facial emotions [89], gaze relationships [11], and more in order to predict subject p_i's impression about subject p_j's likability. The surveys in the Resistance game data capture capture likability through six survey questions designed to elicit p_i's impression of p_j.

In order to predict these six types of dyadic impressions, we have developed a framework called DIPS (Dyadic Impression Prediction System). DIPS involves the following novel features.

1. *Emotion Ranks.* We develop a novel class of features called *emotion ranks*. It is well-known in social science (e.g. [36]) that the emotions of a subject p_i about a subject p_j may be influenced by the emotions of others toward p_j. We first consider emotions of a dyad (p_i, p_j) simultaneously and define *emotion score* as the intensity of a given emotion on p_i's face times the probability that that emotion is directed from p_i toward p_j. Emotion rank takes these dyadic emotion scores as input and uses gaze networks to account for the fact that user

p_i's impression of p_j might depend not only on his facial emotions, but also that of others, as well as his attitude toward those other individuals. This leads to a complex interplay of emotions and network interactions that we aggregate. *Past work on predicting impressions [98] only considers direct gaze relationships and does not consider such network interactions.*

2. *Social Balance Theory.* Classical social network theory has identified the importance of triangles in friend/enemy networks. Such networks are called *signed networks* [43, 50] in which edges can be positive or negative (whether $\pm$ 1 or with positive or negative weights). Balance theory forms an important part of social network theory going back to the time of Heider in the 1950s [66]. It suggests that for a balanced triad (three individuals in this case), the products of any pair of edge weights must be positive. Important phenomena explained by balance theory include the ideas that "a friend of my friend is my friend" and "an enemy of my enemy is my friend". In this paper, we consider the effect of a third party p_k on the impression that p_i has of p_j via a class of features that measure the degree of *imbalance* capturing p_i's impressions of p_j vis-a-vis such third parties. *To our knowledge, this is the first time that social balance theory has been used for predicting impressions from nonverbal data.*

3. *Emotion, Facial Action Units*, and *Temporal Alignment.* Social science theory [36] posits that p_i's impression of p_j and p_j's impression of p_i are not independent. Of course, we see this in our daily lives - if a person doesn't like you, you may not like them back. We might therefore get clues about p_i's impression of p_j by looking at p_j's facial emotions. We define a novel class of *alignment vectors* that capture the alignment — with possible temporal delays in order to account for subjects' response times — between the facial emotions and action units of subjects p_i and p_j.

4. *Temporal Delayed Network.* We introduce the novel concept of a Temporal Delayed Network which is a *multi-layer* network [37, 83] where each layer represents a particular time point. Within a single layer, nodes correspond to players and edges correspond to different interactions between players (e.g. look at, talk to, listen to). Within a layer, edges are labeled with the probability that the stated interaction occurs. Across layers, edges represent identity information by linking the same individuals in different layers, as well as delayed interaction information. *To our knowledge, this is the first time that multi-layer networks have been used in predicting impressions of subjects.* Using this multi-layer

network as an underlying graph, we build a Graph Convolution Network [82] with an attention mechanisms [148] to learn representations and predict dyadic impressions of p_i toward p_j.

Finally, we have implemented the DIPS framework as well as some baselines encompassing past work [10, 16, 98]. We show that DIPS is able to generate AUCs ranging from 73–77% for the six dependent variables capturing impressions of a person p_i w.r.t. person p_j, improves upon the performance of 8 competing baselines from the literature by 19.9–30.8% in AUC and 12.6–47.2% in F1-score. We further conduct ablation tests to show that the novel features we introduce all contribute to this increase in predictive performance.

5.1 Dataset analysis and task description

We test several baselines as well as our new Temporal Delayed Network approach on the Resistance dataset. In most cases, players do not know each other, and the majority of the players in our experiments never played this game or similar games such as "Mafia" or "Werewolf" before. Prior to the game, players go through warm-up activities to get familiar with each other as well as one practice round. After the game, all players fill out a survey. In this chapter, we are interested in questions related to players' impressions of each other. In particular, we focus on the set of questions in a post-game survey asking about the likeability of other players (see Table 5.1). Players had to rate each other on six variables on a 7-point scale.

5.1.1 Dataset analysis

In this section, we conduct a brief statistical analysis of the data. Overall, the dataset contains video and survey data for 348 players from 48 games with 135 of them playing spy roles. Gender distribution is the following: 44% of players were male and 56% were female. Participants were recruited from college student populations with a median age of 21.

Hypothesis 1. First, we test the hypothesis that the gender of players affects the impression, i.e., that females rate players differently than males, and that players rate female and male players differently based on their own gender. Table 5.2 shows

Question #	Variables in the survey
Q1	Very cold : Very warm
Q2	Very negative : Very positive
Q3	Very unpleasant : Very pleasant
Q4	Very unfriendly : Very friendly
Q5	Very unlikable : Very likable
Q6	Very unsociable : Very sociable

Table 5.1: All players had to fill in a survey for each of the other players p_i rating their perception on a 7-point scale for six questions. All questions had the same form but different variables to rate: "Was Player p_i friendly, likable, and pleasant or cold, negative, and unfriendly?". Note that in the original survey questions 2, 4, and 6 were offered in reversed order (lower rating for more positive perception). For the purpose of our research we reversed the scale of these questions so that all survey answers are aligned in polarity.

Rating player	Rated player	Q1	Q2	Q3	Q4	Q5	Q6
Female	Female	4.94 (1.59)	4.87 (1.58)	5.17 (1.44)	4.87 (1.67)	5.21 (1.45)	4.78 (1.77)
Female	Male	5.05 (1.55)	4.93 (1.56)	5.21 (1.40)	4.99 (1.65)	5.23 (1.41)	4.86 (1.70)
Male	Female	5.01 (1.59)	4.99 (1.49)	5.23 (1.32)	5.10 (1.52)	5.32 (1.34)	4.82 (1.61)
Male	Male	4.95 (1.57)	4.79 (1.54)	4.99 (1.45)	4.83 (1.64)	5.04 (1.43)	4.76 (1.65)

Table 5.2: Gender-based distribution of scores: mean (std). We found no statistically significant difference depending on genders of players.

the means and standard deviations of the variables depending on the gender of rating and rated players. We used Mann-Whitney U-test and found no significant difference between these groups.

Hypothesis 2. Second, we checked a similar hypothesis about impression differences based on players' roles in the game. Table 5.3 shows the corresponding means and standard deviations. We did not find any significant differences in this case either.

Hypothesis 3. Table 5.4 shows that correlations between different ratings of questions are relatively high, which means players who score other players high on one variable tend to score the same players high on the other variables. The highest correlation is for a question about Pleasant–Unpleasant and a question about Likable–Unlikable. The lowest correlation is for questions Warm–Cold and Sociable–Unsociable.

Rating player	Rated player	Q1	Q2	Q3	Q4	Q5	Q6
Villager	Villager	5.12 (1.53)	5.05 (1.53)	5.25 (1.38)	5.08 (1.63)	5.32 (1.37)	5.00 (1.62)
Villager	Spy	4.80 (1.57)	4.78 (1.52)	5.05 (1.40)	4.83 (1.61)	5.10 (1.38)	4.58 (1.74)
Spy	Villager	4.93 (1.64)	4.75 (1.59)	5.10 (1.47)	4.88 (1.65)	5.14 (1.49)	4.74 (1.71)
Spy	Spy	5.14 (1.54)	5.04 (1.54)	5.23 (1.36)	4.95 (1.59)	5.28 (1.43)	4.89 (1.68)

Table 5.3: Role-based distribution of scores: mean (std). We found no statistically significant difference depending on the roles of players.

	Q2	Q3	Q4	Q5	Q6
Q1	0.63	0.71	0.56	0.68	0.49
Q2		0.70	0.68	0.63	0.59
Q3			0.64	0.78	0.56
Q4				0.67	0.64
Q5					0.56

Table 5.4: Mutual Spearman correlation between different variables. All correlations are statistically significant with $p \leq 0.05$.

5.1.2 Problem description.

Given the past social science findings that negative impressions are expressed via facial expressions [56] (as opposed to positive impressions which may be "internalized" and not facially expressed), we study the problem of predicting if player p_i will have a negative impression of player p_j according to each of six variables in the survey (Table 5.1). For this, we define the impression on a given variable to be positive if player p_i rates player p_j as 4 or above on the 7-point scale, and negative for ratings of 3 and below. In the Resistance dataset 11-23% of ratings are negative, which is expected, as by default people tend to have a neutral or positive impression of strangers. We observed this both in our own data and it has also been noted in earlier social science research [22]. Yet interactions and observations of a person over time (i.e. during the game) can change the impression to negative. Therefore, we consider a binary classification problem of predicting negative impressions between people according to each of the six variables (in other words, negative impression rating is the positive class in our problem), with six tasks in total.

5.2 Methodology

Most of the social science and psychological literature acknowledges the role of emotions and expressed behavior in forming an impression of a person [64, 36]. We

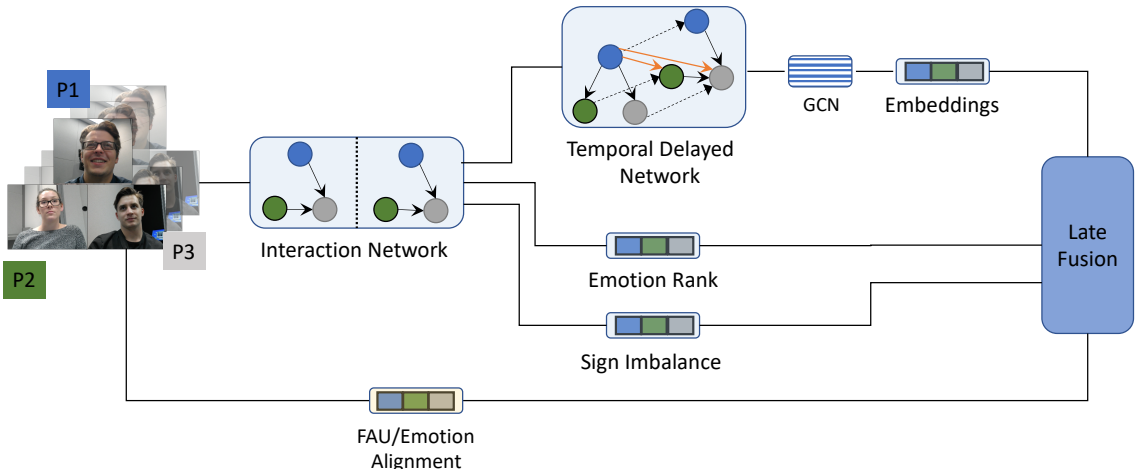


Figure 5.1: DIPS framework. From frontal videos of players we extract: facial expressions and Interaction Networks (look-at, listen-to, talk-to). We use facial expressions to calculate alignment features (Sec. 5.2.3). We use interaction networks and facial expressions to build Emotion Rank (Sec. 5.2.1) and Sign Imbalance (Sec. 5.2.2) features. Furthermore, we use Interaction Networks to build and train our novel Temporal Delayed Network (Sec. 5.2.4) algorithm to produce player embeddings that we use to predict impressions. Each of the feature classes is calculated using all three Interaction Networks (but for the sake of simpilicity, only one is shown in the Figure). Finally, we use individual predictions of all of these methods to build ensembles with late fusion.

therefore use the emotions and facial action units extracted using off-the-shelf tools ([12, 89]) as inputs. After extracting these values for the whole video, we get the vector of values $\mathbf{P}(p_i, e) = [v_1(p_i, e), v_2(p_i, e), \ldots, v_T(p_i, e)]$, where $v_t(p_i, e)$ is either the probability of emotion e [89] for player p_i at time t, or the intensity of a particular facial action unit for OpenFace [12] As negative emotions are causes of negative impressions according to social theories [64], we split emotional expressions into two subsets: positive emotions $\mathcal{E}^+$ (happy) and negative emotions $\mathcal{E}^-$ (angry, disgusted, fearful, sad).

To capture the dynamics of group interactions, we also consider three dynamic interaction networks $G_I = (V_I, E_I)$ derived from [10, 11]: look-at, talk-to, and listen-to networks. Vertices in these networks are participants and edges are interactions among them evolving over time. Formally, a vertex in any of these networks $p_{i,t} \in V_I$ represents player p_i in the game at time t. Each directed edge $(p_{i,t}, p_{j,t}) \in E_I$ has an associated weight representing the probability of a particular interaction between players p_i and p_j at time t: whether player p_i looks at, talks to, or listens to player p_j. The probability of player p_i talking to the player p_j is defined as the product

of probabilities of the player p_i speaking (estimated from facial movements) and the probability of the player p_i looking at the player p_j (estimated using the collective classification approach in [11]). Similarly, the probability of the player p_i listening to player p_j is defined as the product of probability of the player p_i looking at the player p_j and the probability of the player p_j speaking.

Figure 5.1 shows our overall DIPS framework, an ensemble of four novel components. We extract facial expressions and Interaction Networks (look-at, listen-to, talk-to) from the frontal videos of the players. Extraction of the interaction networks also uses the layout of the players in space as described in [11]. We build our novel Emotion Rank (Sec. 5.2.1) and Sign Imbalance (Sec. 5.2.2) features using the interaction networks and facial expressions. We also use facial expressions to calculate our novel alignment features (Sec. 5.2.3). Furthermore, we use Interaction Networks to build and train Temporal Delayed Network (Sec. 5.2.4) to produce player embeddings that we use to predict impressions. Finally, we use individual predictions of all of these methods to build an ensemble with late fusion. The rest of this section describes each of these methods in detail.

5.2.1 Emotion Rank

Building on extensive social science background [36, 133, 56, 64] we propose a way to quantify interpersonal attitude using emotional responses during the game.

First we define the notions of emotion scores and emotion vectors which capture the "amount" of emotions directed from one person to another in a period of time. Given a dynamic interaction network $G_I = (V_I, E_I)$, an emotion $e \in \mathcal{E} = \mathcal{E}^+ \cup \mathcal{E}^-$, participants $p_1, p_2 \in V_I$, a time window τ, a weight function w associated with every edge (probability of the given interaction), we define the *emotion score* (ES) as follows:

$$ES(e, p_i, p_j, G_I, \tau) = \pm \frac{1}{|\tau|} \sum_t v_t(p_i, e) \cdot w(p_{i,t}, p_{i,t}), \quad (p_{i,t}, p_{j,t}) \in E_I, \tag{5.1}$$

where the summation goes over the time window τ with length $|\tau|$, and the sign depends on the emotion e: positive if $e \in \mathcal{E}^+$ and negative otherwise. We further define the *emotion vector* as a vector $EV(p_i, p_j, G_I, \tau) = [ES(e, p_i, p_j, G_I, \tau)]_{e \in \mathcal{E}}$ of emotions scores for all emotions considered.

Second, we discuss how to aggregate the emotion vector into a scalar in order to define the emotion rank. For notational simplicity, we drop the parameters for EV and use

denote $EV^+ = [ES(e, p_i, p_j, G_I, \tau)]_{e \in \mathcal{E}^+}$ and $EV^- = [ES(e, p_i, p_j, G_I, \tau)]_{e \in \mathcal{E}^-}$ respectively to denote the positive and negative subvectors, respectively, of $EV(p_i, p_j, G_I, \tau)$. We define combine the vector $EV(p_i, p_j, G_I, \tau)$ into a single score. This can be done in many possible ways. We experimented with the five aggregation functions shown below:

- $f(EV) = \mathbb{I}_e(EV) = ES(e, p_i, p_j, G_I, \tau)$
- $f(EV) = \max(EV^+) + \min(EV^-)$,
- $f(EV) = \text{avg}(EV^+) + \text{avg}(EV^-)$,
- $f(EV) = sel(max(EV^+), min(EV^-))$,
- $f(EV) = sel(\text{avg}(EV^+), \text{avg}(EV^-))$,

where

$$sel(x_1, x_2) = \begin{cases} x_1, if\ x_1 > -x_2 \\ x_2, if\ x_1 < -x_2 \\ 0, otherwise \end{cases}$$

Note that the *sel* function takes $x_1 (x_1 \geq 0)$ and $x_2 (x_2 \leq 0)$ as inputs, and either returns the one whose absolute value is larger or returns 0 if their absolute values are equal. In our case, the first function selects one of the emotion components, the second and third aggregation functions sum up the attitudes from positive and negative emotions to get an overall attitude, while the last two forms try to select the valence (positive vs. negative) which is more prominent.

We are now ready to recursively define the *Emotion Rank* $ER_f(p_i, p_j, G_I, \tau)$ as

$$\begin{aligned} &ER_f(p_i, p_j, G_I, \tau) = \alpha_0 + \alpha_1 f(EV(p_i, p_j, G_I, \tau)) + \\ &\alpha_2 \sum_{k \neq i,j} \frac{ER_f(p_k, p_j, G_I, \tau) \cdot f(EV(p_k, p_j, G_I, \tau))}{out(p_k)} + \\ &\alpha_3 \sum_{k \neq i,j} \left\{ \frac{ER_f(p_i, p_k, G_I, \tau) \cdot f(EV(p_i, p_k, G_I, \tau))}{out(p_i)} \cdot \frac{ER_f(p_k, p_j, G_I, \tau) \cdot f(EV(p_k, p_j, G_I, \tau))}{out(p_k)} \right\} \end{aligned} \tag{5.2}$$

where f is one of the 5 aggregation functions defined above, $\alpha_i \geq 0$, $\sum_i \alpha_i = 1$, and $out(p_i)$ is an out-degree of the vertex p_i in the graph G_I.

Intuitively, the Emotion Rank from p_i to p_j depends on: (1) the direct edge $(p_i, p_j) \in E_I$; (2) other peoples' Emotion Rank towards p_j; and (3) any path of length 2 $(p_i, p_k, p_j,$ where $(p_i, p_k), (p_k, p_j) \in E_I$. Note that the first summation reflects the

hypothesis that other participants' average attitudes toward p_j may influence the attitude of p_i toward p_j, and the rationale behind the second summation is that the emotional attitude can "pass" along the interactions between people: if p_i is positive towards p_k and p_k is also positive towards p_j, the impression of p_j held by p_i might also shift to the positive side.

As in the case of algorithms such as PageRank [109], we calculate the Emotion Rank values iteratively, starting with a fixed initial value and computing according to the recursive equation (5.2) until convergence with a preset tolerance or until a maximum number of iterations is reached. More details are presented in the Section 5.3.1.

As a result, for a given pair of players, we get the vector of values $[ER_f(p_i, p_j, G_I, \tau)]$ over a varying set of time intervals τ spanning from 1 second to the length of the whole video T. To make predictions for the whole video, we need to aggregate these values into a fixed-length vector to be able to apply standard classifiers. As in [10], we calculate histograms of these values with a fixed number of bins and use these histograms as features for our classification task.

5.2.2 Sign Imbalance

Social scientists have studied balance theory for many years [65, 66, 43]. Intuitively, balance theory looks at triangles in graphs. A triangle is balanced if the number of negative edges is even (i.e. 0 or 2). In the case of weighted graphs, a triangle is balanced if the product of the edge weights is positive, otherwise it is imbalanced. Balance theory has been tested and validated in the study of tribes in India [147], social networks of individuals in towns in Africa [97], how faction-faction relationships change with the Al-Qaeda and ISIS factions ecosystem [14], and much more.

Inspired by the concept of balance in signed networks [66, 87], we define a class of features in the following way. For a given time window τ and a given interaction graph G_I, we build a weighted multi-layer graph (V, E), where V is a set of participants, and for every ordered pair of vertices $p_i, p_j \in V$ there are two edges in E:

- $(p_i, p_j)^+ \in E$ with the associated weight $w^+(p_i, p_j) = max_{e \in \mathcal{E}^+} |ES(e, p_i, p_j, G_I, \tau)|$
- $(p_i, p_j)^- \in E$ with the associated weight $w^-(p_i, p_j) = max_{e \in \mathcal{E}^-} |ES(e, p_i, p_j, G_I, \tau)|$

Note that $w^+(p_i, p_j) \in [0; 1]$ and $w^-(p_i, p_j) \in [0; 1]$ because of the way Emotion Scores are calculated (see Sec. 5.2.1).

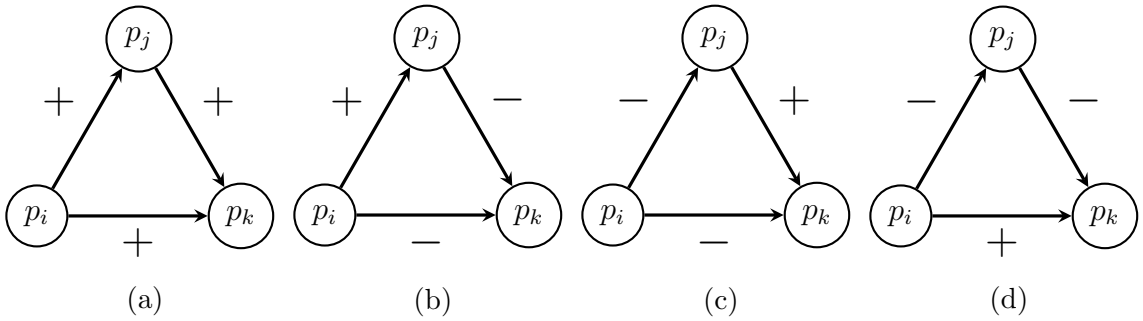


Figure 5.2: Balanced directed signed triads: for any triangle to be balanced the product of signs should be positive. These triads correspond to the following situations: (a) a friend of my friend is my friend, (b) an enemy of my friend is my enemy, (c) a friend of my enemy is my enemy, (d) an enemy of my enemy is my friend.

According to the balance theory in graphs [87], there are four possible balanced relations in any given triangle (Figure 5.2): a friend of my friend is my friend (Fig. 5.2a), an enemy of my friend is my enemy (Fig. 5.2b), a friend of my enemy is my enemy (Fig. 5.2c), and an enemy of my enemy is my friend (Fig. 5.2d). Since in our graph every edge (p_i, p_j) has a weight $w(p_i, p_j) \in [0, 1]$ representing the intensity of emotions of a particular sign aligned with a given interaction, for any triangle to be balanced, balance theory suggests that the following equality will hold:

$$w(p_i, p_j) \cdot w(p_j, p_k) = w(p_i, p_k), \tag{5.3}$$

where w corresponds to w^+ or w^- depending on the sign of the edge of the triangle (Figure 5.2).

We define a *sign imbalance* feature for a participant p_i as the average discrepancy in balance (Eq. 5.3) over all triangles $\{(p_i, p_j), (p_j, p_k), (p_i, p_k)\}$ involving p_i in the graph:

$$SI(p_i, G_I, \tau) = \frac{1}{N} \sum_{p_j, p_k \in V, i \neq j \neq k} |w(p_i, p_j) \cdot w(p_j, p_k) - w(p_i, p_k)|, \tag{5.4}$$

where the summation goes over all possible triangles in the graph containing vertex p_i, N is the number of such triangles, and w corresponds to w^+ or w^- depending on the sign of the edge of the triangle (Figure 5.2).

A variant would use Emotion Rank with the selector aggregation function instead of the Emotion Score values:

- $(p_i, p_j)^+ \in E$ with the associated weight $w^+(p_i, p_j) = max_{e \in \mathcal{E}^+} |ER(e, p_i, p_j, G_I, \tau)|$

- $(p_i, p_j)^- \in E$ with the associated weight $w^-(p_i, p_j) = max_{e \in \mathcal{E}^-} |ER(e, p_i, p_j, G_I, \tau)|$

In this case, we need to normalize the ER values to be in the $[0, 1]$ interval. We obtain a feature vector for each variant of the Emotion Rank ER_f.

Similar to the Emotion Rank features we aggregate the values over all possible time windows τ by calculating histograms with a fixed number of bins. These histograms are used for the classification task at hand.

5.2.3 Emotion and Facial Action Units alignment

The social science literature draws a connection between mutual liking between two people and establishing rapport [141] by synchronizing body language and emotional states. A computational effort [107] also built on this idea and mined the co-occurrence patterns between the features of two people in order to successfully predict personality traits and behaviors.

Since we are considering a task concerning two people, we are interested in how well their emotions and facial expressions are aligned with each other and whether particular emotions expressed by one player cause the same or different emotions in the other player. We use cosine distance $\cos(\mathbf{P}(p_i, e), \mathbf{P}(p_j, e))$ as a measure of alignment between two time series of emotions or facial action units, where $\mathbf{P}(p_i, e)$ is a vector of emotion or facial action unit e intensities that player p_i shows, and $\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^T \mathbf{y}}{||\mathbf{x}|| \cdot ||\mathbf{y}||}$.

As it usually takes time for a person to see another person's emotional state and react to it [124], we also consider the alignment between vector values shifted forward in time $\mathbf{P}_{+\Delta t}(p_i, e) = [v_{\Delta t}(p_i, e), v_{\Delta t+1}(p_i, e), \ldots, v_T(p_i, e)]$. To be able to compute the cosine distance between $\mathbf{P}_{+\Delta t}(p_i, e)$ and $\mathbf{P}(p_j, e))$, we trim the latter to match the length of the shifted vector.

As we also do not know the direction of the effect, i.e. whether player p_i reacts to player p_j or the other way around, we also consider the alignment between vectors shifted backwards in time by a factor Δt as follows:

$$\cos(\mathbf{P}_{-\Delta t}(p_i, e), \mathbf{P}(p_j, e)) = \cos(\mathbf{P}(p_i, e), \mathbf{P}_{+\Delta t}(p_j, e)).$$

Finally, we form a vector $AL(p_i, p_j, e)$ of cosine distances for time shifts varying from

$-\Delta t$ to $+\Delta t$:

$$AL(p_i, p_j, e) = [\cos(\mathbf{P}_{-\Delta t}(p_i, e), \mathbf{P}(p_j, e)), \ldots, \cos(\mathbf{P}(p_i, e), \mathbf{P}(p_j, e)), \ldots, \cos(\mathbf{P}_{+\Delta t}(p_j, e), \mathbf{P}(p_i, e))]$$

We also extend the definition by considering possible pairs of facial expressions for a given pair of players $AL(p_i, p_j, e_l, e_k)$. For the prediction task at hand, for each pair of players we concatenate alignment vectors for different pairs of emotions or facial action units e_l and e_k to form a feature vector

$$AL(p_i, p_j) = [Al(p_i, p_j, e_l, e_k)], \ (l, k) \in [1, N] \times [1, N],$$

where N is the number of facial expression considered.

5.2.4 Temporal Delayed Network

We leverage the concept of multi-layer networks [37, 83] as well as recent advances in non-euclidean learning such as Graph Convolution Networks [82, 148], which have been proved to be powerful for learning in social networks. We propose an approach to building graphs that captures the interaction between players as well as the dynamics of players' behavior. We call this model a Temporal Delayed Network (TDN).

Given a dynamic interaction graph $G_I = (V_I, E_I)$ (for instance, look-at graph), we build a multi-layer network [20] $G = (V, E)$ in the following way (Figure 5.3):

- Vertices $p_{i,t} \in V$ represent player p_i's state at time point t.
- We introduce three types of edges $E = E_1 \cup E_2 \cup E_3$:
    1. *Interaction edges* derived from the interaction graph G_I: $(p_{i,t}, p_{j,t}) \in E_1$ if and only if $(p_{i,t}, p_{j,t}) \in E_I$. Interaction edges carry the same weight as their counterparts in the interaction graph G_I.
    2. *Identification edges* connect the vertices corresponding to the same player at different points in time: $(p_{i,t}, p_{i,t'}) \in E_2$ if $t' - t \leq \Gamma$. Each identification edge has associated weight exponentially decaying with difference in time steps: $c(p_{i,t}, p_{i,t'}) = \gamma^{t'-t}$. This allows propagation of the player's inner state in time but restricts the effect of the past behaviour on the present behaviour.

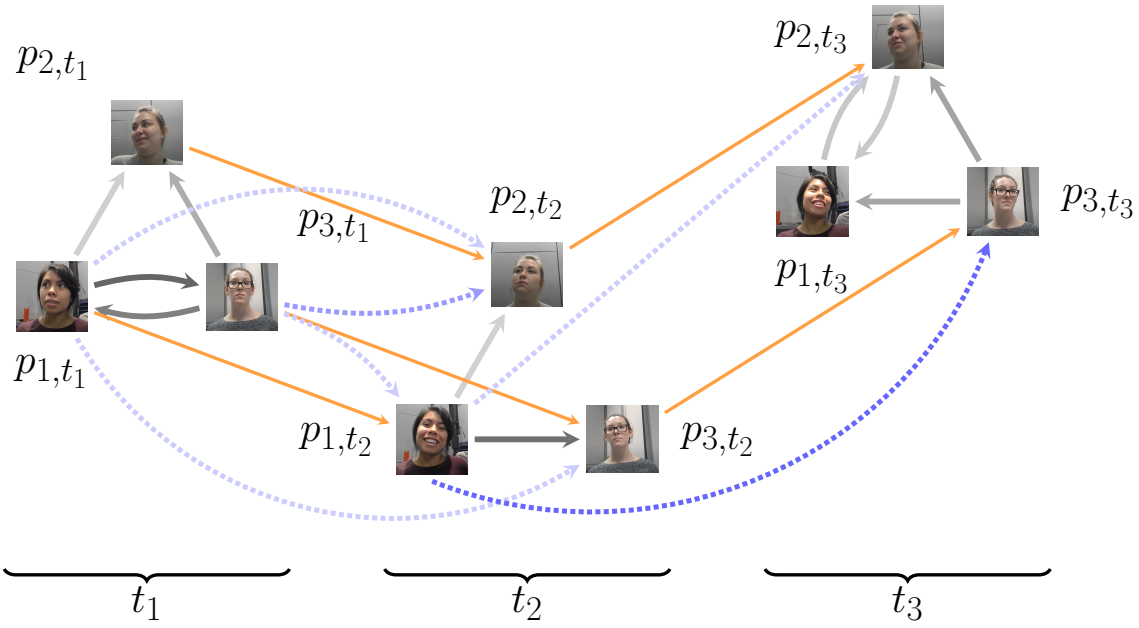


Figure 5.3: Example of a Temporal Delayed Network (TDN) for one of the games in the dataset (best viewed in color). Thick gray edges represent the interaction graph (in this example, look-at graph), thin orange edges represent identification edges (connecting the same player in different layers), blue dotted edges represent the delayed influence edges. Color intensity represents the probability of the given interaction occurring at that time step (in other words edge weights). Here we show a subset of players and a subset of edges for clarity.

3. *Delayed influence edges* build on the idea that interactions can have a delayed effect: for instance, player p_i seeing player's p_j facial expression can affect player p_i's impression only on the next time step. So, $(p_{i,t}, p_{j,t'}) \in E_3$ if and only if $(p_{i,t}, p_{j,t}) \in E_1$ and $(p_{j,t}, p_{j,t'}) \in E_2$. Associated weight is $c(p_{i,t}, p_{j,t'}) = c(p_{i,t}, p_{j,t'}) \cdot c(p_{j,t}, p_{j,t'})$.

For any person p_i at time t, we want to learn an embedding of the corresponding node $p_{i,t}$ which contains the temporal visual information of the person, the influence from the person to others in the group, and conversely from others in the group to the person. These representations are further grouped pairwise to learn the dyadic impression one person has of another.

For the sake of simplicity, we denote vertices of the network with letters u and v in the following discussion. We use $IN_k(v) = \{u|(u,v) \in E_i\}$ and $OUT_k(v) = \{u|(v,u) \in E_k\}$ to denote the incoming and outgoing edge sets of a vertex $v \in V_I$ for edge type E_k, respectively. Inspired by [159] who build spatial temporal Graph Neural Networks to model the temporal dynamics of skeleton joints, we employ the graph convolution

layer in our three sets of directed edges to update the node embedding $x_v \in \mathbf{R}^m$ of a vertex v:

$$\tilde{x}_v = \sum_{k=1}^{3} \left(\sum_{u \in IN_k(v)} c(u,v) w_k(u,v) f_k(x_v) + \sum_{u \in OUT_k(v)} c(v,u) w_k(v,u) f_k(x_v) \right), \quad (5.5)$$

where $f_k(\cdot)$ is a fully connected layer, and $w_k(u,v)$ denotes the learnable weights of the edge $(u,v) \in E_k$. We use the graph attention mechanism [148] to allow the model to attend edge importance from the projected features:

$$w_k(u,v) = attn(f_k(x_u), f_k(x_v)), \quad (5.6)$$

where $attn : \mathbf{R}^n \times \mathbf{R}^n \to \mathbf{R}$ is the asymmetric attention block from [148]:

$$attn(x_1, x_2) = \frac{exp(LeakyReLU(a^T[x_1||x_2]))}{\sum_{x_2} exp(LeakyReLU(a^T[x_1||x_2]))}, \quad (5.7)$$

where $a \in R^{2n}$ is a learnable attention vector, $x_1, x_2 \in R^n$, and $||$ denotes vector concatenation. Note that for any given v, this and the normalization of $c(u,v)$ ensures that $\sum_v c(u,v) = 1$ for all u and $\sum w_k(u,v) = 1$ for all k.

Finally, we update the node embedding x_v using the ReLU function:

$$x_v = ReLU(\tilde{x}_v).$$

After two layers of graph convolutions, we apply temporal average pooling for node embeddings of each person:

$$\bar{x}_{p_i} = \frac{1}{T} \sum_{t=1}^{T} x_{p_{i,t}}. \quad (5.8)$$

To predict whether person p_i has a negative impression of the player p_j, we apply the prediction layer below to output the probability:

$$P(p_i, p_j) = \sigma(o^T[\bar{x}_{p_i}||\bar{x}_{p_j}]), \quad (5.9)$$

where o is the trainable projection vector, σ is the sigmoid function.

Initialization of node embeddings. We use the facial expression embeddings [89] to initialize our node embeddings. Specifically, we remove the last fully con-

	Feature class	Q1	Q2	Q3	Q4	Q5	Q6
Baselines	FAU + hist. [10]	0.612	0.581	0.594	0.567	0.586	0.609
	Emotions + hist. [10]	0.584	0.555	0.600	0.577	0.595	0.579
	Speech features [98]	0.589	0.514	0.563	0.562	0.577	0.608
	Face features [98]	0.589	0.519	0.574	0.535	0.569	0.566
	Face and speech features [98]	0.595	0.523	0.555	0.552	0.569	0.605
	Speaking act [16]	0.574	0.506	0.538	0.533	0.555	0.584
	VFOA [16]	0.501	0.522	0.547	0.511	0.554	0.545
	VFOA-Spk-Act [16]	0.587	0.513	0.506	0.525	0.545	0.614
DIPS (ours)	FAU alignment	**0.676**	0.597	**0.685**	**0.623**	**0.727**	0.627
	EMO alignment	0.597	0.565	0.627	0.598	0.626	0.583
	Emotion Rank: look-at	0.573	0.562	0.588	0.573	0.580	0.583
	Emotion Rank: talk-to	0.572	0.577	0.577	0.570	0.583	0.558
	Emotion Rank: listen-to	0.593	0.571	0.583	0.578	0.578	0.605
	Sign Imbalace: look-at	0.572	0.564	0.595	0.558	0.586	0.561
	Sign Imbalance: talk-to	0.592	0.560	0.598	0.578	0.609	0.563
	Sign Imbalance: listen-to	0.581	0.568	0.576	0.555	0.599	0.588
	TDN: look-at	0.635	0.574	0.606	0.577	0.617	**0.633**
	TDN: talk-to	0.615	**0.638**	0.626	0.610	0.612	0.573
	TDN: listen-to	0.649	0.611	0.605	0.610	0.633	0.630

Table 5.5: Performance (AUC) of individual features on six variables. The top eight rows show results for baseline features. The rest of the table shows the performance of our proposed features derived from different interaction graphs, as well as the performance of TDN built upon these interactions graphs. On all of the variables our proposed methods yield the best performance with statistically significant difference over the best baseline results ($p < 0.05$).

nected layer of their proposed CNN and use the extracted features as our initial node embeddings.

5.3 Experimental results

5.3.1 Experimental setup

General setup. We split the dataset into 10 folds by games. Since each player appears in only one game, we always make predictions about players never seen before. We use four standard classifiers for our predictions: k-Nearest Neighbor, Logistic Regression, Gaussian Naive Bayes, and Random Forest.

As our dataset is highly imbalanced (11–23% of positive samples), we use AUC as the performance metric. We test our proposed feature types with each of the afore-

mentioned classifiers. We report the best AUC for any given class of features among all classifiers.

In the case of Emotion Rank features, we initialize the values in Equation 5.2 to $\frac{1}{n}$, where n is the number of players in a particular game. We then iteratively calculate the values until convergence with tolerance level 10^{-5} for L_∞ distance between values at the end of consecutive iterations or for up to 100 iterations.

For facial expression alignment features, we perform iterative greedy feature selection by first considering one feature $Al(p_i, p_j, e_{l_1}, e_{k_1})$ at a time and then selecting the best performing one. We then construct a concatenation of two features: the first is the best feature from the previous stage and the second is the one discovered in the current iteration which, when added to the feature selected in the previous iteration gives the best result. We repeat this process until adding new features does not improve the performance on a validation fold. We use this process instead of exhaustive search because the number of features increases exponentially with increasing length.

For Emotion Rank and Sign Imbalance features, we consider all possible combinations of features produced by different aggregation functions. We report the performance of the best combination and we further analyze the influence of aggregation functions on the performance.

Baselines. We adopted emotion and FAU histograms as described in Bai et al. [10], speech acts, facial, and multi-modal features (face and speech) as described in Muller et al. [98], speaking acts, the visual focus of attention (VFOA), and combined features as described in Beyan et al. [16]. In all three cases, we used the best performing features that we could calculate on our dataset (for instance, we did not use features related to hand movements because not all videos contain a clear view of hands). Since all of the aforementioned papers deal with predicting values for a single person and our tasks are dyadic, we form dyadic features by concatenating features of a pair of individuals just as we do with our proposed methods. We then applied the same battery of classifiers mentioned earlier.

Temporal Delayed Network (TDN) training. We split each video into 100 clips. For each clip, we sample $\Gamma = 5$ frames (1 frame per second) to build a Temporal Delayed Network. We set the decay rate $\gamma = 0.8$. We use two graph convolution layers with 128 dimensions of node embeddings for both layers. Each layer is followed by

Batch Normalization and ReLU activation. We use the Adam optimizer with learning rate 10^{-4} and weight decay 10^{-4}. The network is trained for 200 epochs with a batch size of 64.

5.3.2 Head to Head Feature Comparisons

First, we compare the individual performance of the proposed feature classes. Table 5.5 shows the performance (AUC) of our proposed methods compared to each other and to the chosen set of baselines. We see that on all of the variables, at least one of our proposed methods outperforms the baselines: TDN models yield the best performance on two variables out of six, and FAU alignment features with greedy feature selection performs best on the other four variables. When it comes to particular classes of features, even though Emotion Rank features and Sign Imbalance features do not always yield the best performance, on all of the variables their results are either on par with baseline features or higher. If we consider F1 as a metric of interest (Table 5.6) rather than AUC, our proposed methods still yield the highest F1-score.

5.3.3 Late fusion

Figure 5.1 shows how several individual classes of features provide predictions for our task. To further improve performance and take advantage of the complementarity of individual approaches, these predictions are then combined using late fusion. Given a predicted probability p_i from the i'th individual predictor, we combine the predictions linearly as $\Sigma_{i=1}^{N} w_i p_i$ (where each $w_i \in [0, 1]$ and $\Sigma_{i=1}^{N} w_i = 1$) to compute an overall probability. We use a grid search over the space of possible values to find the best w_i's value. The best w_i learned on the training and validation sets are used in the predictions on the test set (so in particular, the test set was never used when computing the w's).

Table 5.7 shows the best performing ensembles for each predicted variable as well as the results of an ablation study for those ensembles. The AUC numbers are shown by default and F1 scores are shown in parenthesis. DIPS improves the best baseline models by 19.9%–30.8% for AUC and 12.6%–47.2% when using the F1-score metric. The *improvement* provides by DIPS over a baseline algorithm *Base* w.r.t. a metric μ (e.g. AUC or F1 score) is defined to be $impr = \frac{\mu(\text{DIPS}) - \mu(Base)}{\mu(Base}$.

	Feature class	Q1	Q2	Q3	Q4	Q5	Q6
Baselines	FAU + hist [10]	0.306	0.318	0.239	0.318	0.207	0.350
	Emotions + hist [10]	0.266	0.301	0.225	0.341	0.212	0.365
	Speech features [98]	0.302	0.269	0.217	0.300	0.198	0.369
	Face features [98]	0.310	0.277	0.221	0.285	0.194	0.360
	Face and speech features [98]	0.307	0.269	0.190	0.313	0.184	0.363
	Speaking act [16]	0.291	0.245	0.198	0.291	0.201	0.360
	VFOA [16]	0.246	0.202	0.190	0.262	0.175	0.290
	VFOA-Spk-Act [16]	0.295	0.267	0.179	0.308	0.191	0.388
DIPS (ours)	FAU alignment	0.337	0.243	0.252	0.299	**0.269**	0.372
	EMO alignment	0.298	0.274	0.237	0.280	0.222	0.354
	Emotion Rank: look-at	0.264	0.293	0.212	0.292	0.216	0.358
	Emotion Rank: talk-to	0.277	0.294	0.198	0.330	0.196	0.344
	Emotion Rank: listen-to	0.290	0.304	0.214	0.320	0.185	0.349
	Sign Imbalace: look-at	0.290	0.302	0.203	0.303	0.175	0.343
	Sign Imbalance: talk-to	0.254	0.296	0.198	0.293	0.201	0.325
	Sign Imbalance: listen-to	0.286	0.309	0.212	0.330	0.210	0.365
	TDN: look-at	0.350	0.323	0.243	0.352	0.238	**0.410**
	TDN: talk-to	**0.403**	**0.359**	**0.263**	0.365	0.249	0.403
	TDN: listen-to	0.378	0.357	0.242	**0.368**	0.234	0.358

Table 5.6: Individual features performance (F1-score) on six variables. The top eight rows show results for baseline features. The rest of the table shows the performance of our proposed features derived from different interaction graphs, as well as the performance of TDN built upon these interactions graphs. Our proposed methods show statistically significant improvements (with $p < 0.05$) over the baseline approaches (highlighted in the table).

Thus, if μ is AUC and DIPS and a baseline algorithm yield AUCs of 0.8 and 0.7 respectively, then the improvement ratio would be $\frac{0.8-0.7}{0.1} = 14.29\%$. To assess the importance of each feature in the ensemble, we exclude features one at a time, find the performance of the reduced ensemble, and compare with the performance of the full ensemble. *The Excl. columns show the reduced performance when excluding the specific features from late fusion.* Comparing Table 5.7 with Tables 5.5 and 5.6, we can see that ensembles significantly outperform the individual features. Of all the types of features considered, we observe that FAU alignment features (Sec. 5.2.3) and TDN models (Sec. 5.2.4) are the most important across all predicted variables.

5.3.4 Ablation Study

In this section, we report experiment results from our ablation study.

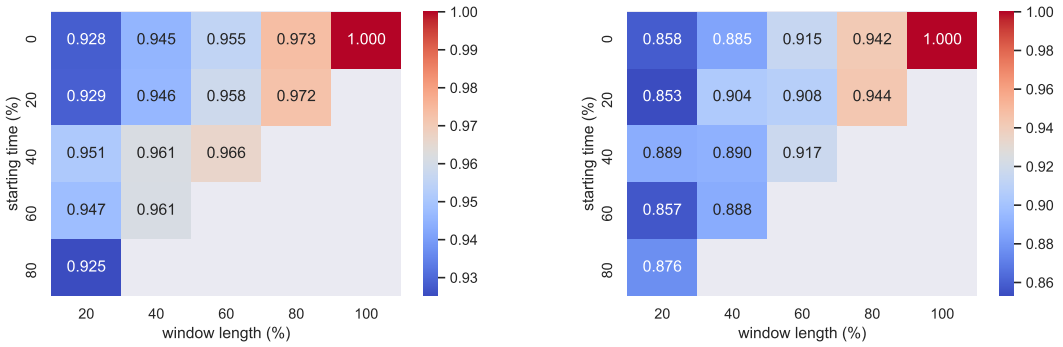
Q1			Q2		
Features	All	Excl.	Features	All	Excl.
ER: look-at	0.744 (0.385)	-0.014	ER: look-at	0.719 (0.378)	-0.015
FAU AL		-0.027	ER: listen-to		-0.006
TDN: look-at		-0.028	ER: talk-to		-0.021
TDN: listen-to		-0.015	Emo Alignment		-0.011
TDN: talk-to		**-0.036**	FAU Alignment		-0.029
			TDN: look-at		-0.010
			TDN: listen-to		**-0.041**
			TDN: talk-to		-0.023
Q3			**Q4**		
Features	All	Excl.	Features	All	Excl.
Emo Alignment	0.767 (0.317)	-0.024	ER: look-at	0.733 (0.384)	-0.026
FAU Alignment		**-0.061**	Emo Alignment		-0.028
TDN: look-at		-0.022	FAU Alignment		-0.020
TDN: listen-to		-0.032	TDN: look-at		-0.027
TDN: talk-to		-0.015	TDN: listen-to		-0.034
			TDN: talk-to		**-0.036**
Q5			**Q6**		
Features	All	Excl.	Features	All	Excl.
ER: listen-to	0.778 (0.312)	-0.022	SI: look-at	0.736 (0.441)	-0.011
Emo Alignment		-0.020	SI: listen-to		-0.009
FAU Alignment		**-0.050**	SI: talk-to		-0.012
TDN: look-at		-0.023	FAU Alignment		-0.015
TDN: listen-to		-0.028	**TDN: look-at**		**-0.042**
TDN: talk-to		-0.028	TDN: listen-to		-0.002
			TDN: talk-to		-0.028

Table 5.7: DIPS results (AUC). For each variable we report the best combination of features, ensemble performance (All) and drop in performance when excluding one of the features in the combination (Excl.) with the most important feature highlighted. We also report the F1 score for the corresponding feature late fusion model (shown in parenthesis). Ensemble performance improvements over the best individual feature performance are statistically significant ($p < 0.01$)

.

Time effect analysis

Our proposed features such as Alignment features, Emotion Rank and Sign Imbalance features use all available video footage. We are interested in identifying which part of the video provides the most important information for the problem of impression prediction. In order to determine this, we ran our experiments on videos restricted to



(a) Emotion Alignment (b) FAU alignment

Figure 5.4: Time effect on the feature performance. Heatmaps show how performance of the corresponding features drops if we restrict available video length and vary the starting time of that video relative to the highest achieved performance (equal to 1 on heatmaps). Numbers in heatmaps are averaged over all six variables.

specific time windows defined by varying the length of the window and starting time of the window.

Figure 5.4 shows the relative performance change we observed for various time windows.

Finding 1 *We found that considering only 20% of the video yields more than 86% of the classification performance achieved on the whole video. The longer the window we consider, the higher the performance we get. To achieve the best result, we need to consider the whole video. Given the same window length, we can achieve slightly better results if we consider the second half of the game rather than the first half, but the starting time is less important than video length.*

Interaction graph effect

To analyze which of the three interaction graphs provides the best performance, we build ensembles from features built using only one of the graphs. For each of the look-at, listen-to, and talk-to graphs we used Emotion Rank, Sign Imbalance, and TDN models. Table 5.8 shows the performance of corresponding ensembles for each of the predicted variables.

Finding 2 *First, we see that different features in the ensemble are complementary to each other, as every ensemble improves upon each individual feature performance (Ta-*

ble 5.5). We also see that look-at is the least important graph when taken individually, as ensembles based on the other two graphs outperform it on each predicted variable. From Table 5.7, we can see that all three graphs contribute to the best performing ensembles on each of the variables, however.

Emotions/FAU effect

To get more insight into which of the facial expressions provide the most information to our models, we look at the combinations of FAU and emotion alignment features that yield the best performance in our models (Table 5.5). Table 5.9 shows the most and the least important facial action units defined by how often they occur among the best performing expression pairs in alignment features (Sec. 5.2.3).

Finding 3 *The most common pair of expressions was (AU05, AU23) suggesting that raising of the upper lid (AU05) and tightening of lips (AU23) are the most important FAUs. These findings are consistent with a similar analysis of Action Unit importance in a different case, namely low rapport detection [98].*

Attention weights of the Temporal Delayed Network

In this experiment, we study the learned attention weights (Equation 5.6) of interaction edges and delayed influence edges defined in Temporal Delayed Network in Section 5.2.4. For any given dependent variable and a given graph, we first get all pairs of people (p_i, p_j) whose impression labels are predicted correctly by the trained TDN model within the training set. Second, we compute the average attention weights of these pairs for the two types of edges separately. The larger the average weight of an edge, the more the trained TDN focuses on the type of edge in order to make

Interaction Graph G_I	Q1	Q2	Q3	Q4	Q5	Q6
look-at	0.659	0.606	0.648	0.614	0.645	0.651
listen-to	0.616	**0.654**	0.643	0.633	**0.677**	0.629
talk-to	**0.667**	0.650	**0.655**	**0.648**	0.622	**0.655**

Table 5.8: Interaction graph importance: we find ensemble performance for features obtained using only one of the interaction graphs. Features used in the ensemble: Emotion Rank (ER), Sign Imbalance (SI) and TDN.

	FAU		Emotions	
	Most often	Least often	Most often	Least often
Scoring player p_i	AU02, AU05, AU15, AU20, AU25	AU07 AU14	Happy	Angry
Scored player p_j	AU23 AU14 AU01 AU17 AU25	AU04 AU20	Happy	Fearful
Overall	AU17 AU05 AU45 AU23 AU15 AU25	AU07 AU12 AU04	Happy	Angry

Table 5.9: Most important facial expressions in the alignment features. FAUs are sorted in the increasing order of their occurrence among the best performing combinations (according to greedy selection process).

	Q1		Q2		Q3		Q4		Q5		Q6	
Graph G_I	E_1	E_3	E_1	E_3	E_1	E_3	E_1	E_3	E_1	E_3	E_1	E_3
look-at	0.342	0.076	0.111	0.071	0.294	0.312	0.104	0.341	0.301	0.294	0.350	0.306
talk-to	0.207	0.353	0.244	0.354	0.335	0.284	0.235	0.336	0.263	0.225	0.135	0.313
listen-to	0.337	0.068	0.293	0.097	0.377	0.259	0.260	0.295	0.333	0.228	0.361	0.276

Table 5.10: Average attention weights of edges for correctly predicted dislike pairs. For each variable, the left number (E_1) shows the average attention weights for interaction edges, while the right (E_3) shows the average attention weights for delayed influence edges.

correct predictions. Therefore, larger numbers indicate higher importance of edges in making predictions. Table 5.10 shows the results.

Finding 4 *Comparing the two types of edges, we observe that the interaction edges (E_1) get more attention from* TDN *models than delayed influence edges (E_3) on average.*

Finding 5 *Among the three types of graphs, we find that the* TDN *focuses more on the interaction edges (E_1) of the listen-to graph, while it focuses more on the delayed influence edges (E_3) of speak-to graphs.*

Variable analysis

We want to use our experimental results to answer the question: *which of the dependent variables is the hardest to predict and which is the easiest?* From the results on the performance of individual features (Table 5.5) and late fusion models (Table 5.7), we see that our models yield the highest performance for Question 5 of the survey (very unlikable to very likable scale). At the same time, our models perform the worst on Question 4 (very unfriendly to very friendly scale) and Question 2 (very negative

to very positive scale). This effect could be partly attributed to higher imbalance in Question 5: only 12% of samples are positive for this variable, compared to 19% and 21% for Question 2 and Question 4 respectively. Another possible explanation is the nature of the questions: it could be easier to answer such questions as to whether a person is likable or unlikable and whether that person is pleasant or unpleasant (Questions 5 and 3) as opposed to more vague questions such as whether a person is positive or negative and friendly or unfriendly (Questions 2 and 4 respectively).

5.4 Conclusion

There are many applications where it is important to understand the like/dislike relationships between people in a group. A particular case is a diplomatic or trade negotiation between countries where it might be useful for country C1 to understand the like/dislike relationships between people in the delegation for country C2.

In this chapter, we provide a framework called DIPS (Dyadic Impression Prediction System). DIPS has three major innovations. First, we develop the novel concept of emotion scores and emotion ranks that combine facial emotions with gaze networks. Second, we use social balance theory for the first time in order to propose sign imbalance features. Third, we develop a novel TDN framework which combines *multi-layer networks* with Graph Convolution Networks (unlike most past work in computer vision that focus on GCNs alone).

We show that the DIPS framework beats out several existing baselines in predicting dyadic impressions by 19.9%–30.8% for AUC and 12.6%–47.2% when using the F1-score metric.

Chapter 6

Discussion and future work

6.1 Conclusion

In this thesis, we studied the following problems:

1. detecting deception from long videos of people interacting in groups,
2. detecting deception from the videos of statements by public figures,
3. predicting dyadic impression from the videos of people participating in group interaction.

We outlined the challenges of these problems among which are:

- lack of large, diverse datasets,
- lack of methods dealing with long videos,
- lack of existing work exploiting natural group structure of the task,
- lack of methods analyzing the content of the speech that is tested for being deceptive.

In an attempt to overcome these challenges, we made the following contributions:

- We collect a dataset of public figures from several countries worldwide making truthful and deceptive statements. We call this the Global Political Deception Dataset. It is the first multimodal dataset of this kind with subjects drawn from a variety of cultures and linguistic backgrounds;
- we introduced a class of histogram-based features that build on well known low-level (eye/head movement, facial action units) and high-level (emotion features from Amazon Rekognition) features to aggregate the information over the whole length of a video,

- we introduced a novel class of "meta-features" called LiarRank that builds on the basic features and allows us to use group-level information to improve deception detection in group videos,

- We proposed a way to build a graph with nodes representing videos of politicians, topics of the messages, and news articles about those politicians: this method allows putting the content of the analyzed statement into a broader context and assess how likely it is to be deceptive;

- We developed a novel class of features we call Deception Score that brings together intrinsic properties of the video (how likely it is to be deceptive) with the assessment of how likely the message from the video to be deceptive;

- We suggested a set of features exploiting the group nature of the interaction to analyze the mutual attitude of participants:

  - We developed a novel class of features called *emotion ranks* that incorporates path in the group graph of lengths up to 3: how one player p_i feels towards player p_j, how other players feel toward player p_j, and finally other players' attitude toward player p_j modulated by how player p_i feels toward those players.

  - We proposed a class of features derived from the balance theory [66] that uses signed networks [43, 50] to find the effect of a third party p_k on the impression that p_i has of p_j via a class of features that measure the degree of *imbalance* capturing p_i's impressions of p_j vis-a-vis such third parties.

  - We also used a novel class of *alignment vectors* that capture the alignment — with possible temporal delays to account for subjects' response times — between the facial emotions and action units of subjects p_i and p_j.

- We introduced the novel concept of a Temporal Delayed Network which is a *multi-layer* network [37, 83] where each layer represents a particular time point. Within a single layer, nodes correspond to players, and edges correspond to different interactions between players (e.g., look at, talk to, listen to). Within a layer, edges are labeled with the probability that the stated interaction occurs. Across layers, edges represent identity information by linking the same individuals in different layers and delayed interaction information. We built a Graph Convolution Network [82] with attention mechanisms [148] on top of this

Figure 6.1: Examples of face obstruction. Top row: in a video of Atiku Abubakar (Nigeria) giving a public speech, a banner obstructs the view of his face. Bottom row: several examples of poor face visibility when players in the Resistance game turn their heads away from the camera or put their hands over their face.

multi-layer network as an underlying graph. This let us learn representations and predict dyadic impressions of p_i toward p_j.

- for all three tasks we considered, we introduced ensemble-based prediction models that significantly improved on existing baselines for those tasks.

6.2 Limitations

Each of the methods we proposed in this dissertation has its own set of limitations, but some limitations are common for all of them:

- *Dependence on video quality.* The datasets we worked with have either HD quality (the Resistance dataset) or TV quality (Global Political Deception Dataset). Most cameras today can record with decent resolution, but this would not be the case if we wanted to analyze some old video footage. This limitation can be partially alleviated by improving the quality of underlying models for facial analysis [12, 11] (for example, by training the deep models on a set of old videos) or by applying super-resolution techniques. There is a limit to how much we can degrade the video until we render facial tracking or video analysis models useless, however. For our proposed models, more research is needed to analyze how the quality of predictions depends on the quality of the video.

- *Dependence on face visibility.* The same facial analysis models highly depend on good face visibility in the analyzed videos: the entire face should be clearly

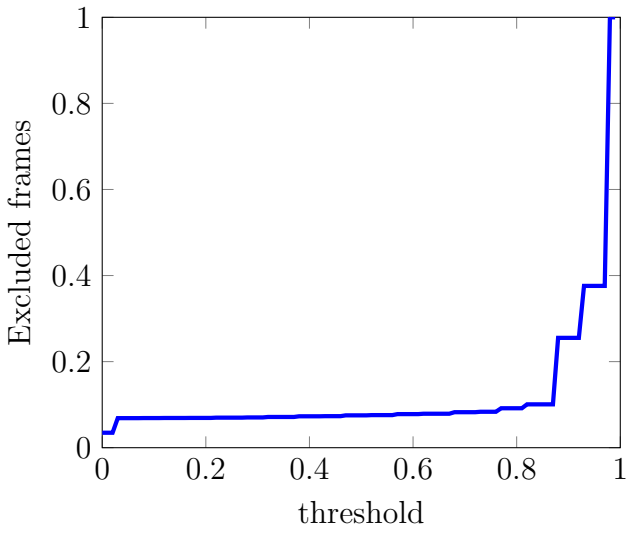


Figure 6.2: Excluded frames rate vs. confidence scores threshold for the Global Political Deception Dataset. We chose to use threshold of 0.75 since it allows us to keep on average 92% of frames.

visible, looking directly at the camera or with as little rotation as possible. This ideal situation is not always the case, as shown in Figure 6.1: people tend to touch their faces, turn their heads, or some objects can appear between the face and the camera. All of this makes it harder to track facial features and decreases the quality of predictions.

To deal with the potential issues of undetected faces we rely on OpenFace library providing confidence scores for detecting faces. We want to exclude frames with low confidence scores. Figure 6.2 shows how the average rate of excluded frames depends on the threshold. We chose the threshold of 0.75 that allows us to keep on average 92% of all frames. We have not assessed, however, how much performance deteriorates when this threshold is changed.

- *Dependence on the language models.* In the Resistance dataset, all participants spoke English. In the Global Political Deception Dataset, we use automated translation to translate non-English transcripts into English. Although current Natural Language Processing tools cover more and more languages, some relatively rare languages do not have good models for transcribing and translating. Acoustic and prosodic features do not depend on the language, but textual and linguistic features use properties of spoken language to make useful representations.

- *Lack of end-to-end training.* We use off-the-shelf tools to extract some facial

features [12] and hand-designed features inspired by psychological and social studies. The rise of Deep Learning techniques, however, has shown the powerful capabilities of deep neural networks to learn good representations directly from data. We use end-to-end trainable models only with Temporal Delayed Network in Chapter 5). This limitation is primarily due to lack of annotated data.

6.3 Future work

The work presented in this dissertation can be extended in several ways. We will discuss some of the potential directions of future research.

Improving performance of existing models. One desirable direction is to advance the models proposed in this thesis to achieve better performance. This can be done in few ways:

- Building end-to-end trainable models. This would be hard to do with the currently available datasets due to their small size. Still, this obstacle can be avoided by using common deep learning techniques: pretraining on larger datasets for related tasks (e.g., emotion recognition), using transfer learning approaches, or using self-supervised learning to exploit massive amounts of unlabeled data available online. In fact, one work on deception detection [42] already tried applying adversarial learning to train a deep model on the small Real-Life Trial dataset.

- Models in this thesis, as well as most other works (some examples: [106, 10, 134]) either do not use temporal information in the videos or use it on a relatively short time scale. In our Temporal Delayed Network we use five second window with one frame per second, Stathopoulos et al. [138] use time-series associated with the video stream but analyze at most 180 consecutive frames which amounts to 6 seconds of video. But temporal information is an essential aspect of the data: transition from happy to angry could be very different from the transition from angry to happy. It could probably reveal some vital information about the subject's inner state. And these transitions happen on multiple scales: from momentary frown to slow change in the mood over minutes and hours. Nevertheless, the problem of analyzing long videos is still open even in the broader computer vision context.

Figure 6.3: Example of AI generated news anchor: a digital version of a regular Xinhua news anchor named Qiu Hao. This version can deliver any news provided in text.

Developing real-time models. Most of the works in deception detection and social behavior analysis, including the methods proposed in this work, rely on the whole video to make an inference. This could be useful if we want to analyze the data retrospectively or when we have the luxury of waiting for the analysis to be done. But in some use cases, promptly predicting in real-time can provide significant advantages. For example, suppose we are participating in business negotiations. In that case, we want to know whether our prospective partners are honest before making any executive decisions. Usually, we have no time to analyze the whole conversation before rendering the decision.

This goal could be achieved by developing models that require a shorter time window to make a sufficiently reliable prediction. The use cases mentioned above would also require making the models lightweight to be implemented on devices like cameras or in video-conferencing software.

Generative tasks. There have been several prominent papers in recent years achieving photo-realistic quality of generated images of human faces [79, 80]. Following the success of generative modeling, several research groups tried generating videos of human faces with different properties [144, 169, 58, 121]. China's Xinhua News Agency announced in 2018 the first AI-generated news anchor[1] shown in Figure 6.3.

[1]Image credit: https://www.theguardian.com/world/2018/nov/09/worlds-first-ai-news-anchor-unveiled-in-china

One way to extend the social behavior analysis is to apply the discriminative models in the generative setting: learning how to generate human faces that look very realistic and behave like real humans. This behavior can include not just the ability to show human emotions but to show the emotions in the proper context. Humans are generally good at reading some of other people's emotions, so we want to create virtual faces that show happiness when the context suggests that or express sadness and empathetical behavior when the conversation prompts that. Going beyond that, it might be useful to create faces that show typical signs of deception, various personality traits, and other high-level psychological activity. One step further would be generating groups of faces that would demonstrate realistic group behavior. Solving this problem can be useful in several ways. First, for creating human-like virtual assistants to help and provide companionship. Second, the generated data and generative models can be used to improve the quality of discriminative models further.

Analyzing the context of the interaction. One of the reasons why polygraph test results are used with caution is that the cues used to detect deception are essentially the same as cues for being nervous or anxious. And when a subject is interrogated under a high-stake accusation, it is not surprising for the subject to become agitated, which subsequently can be mistaken for being deceptive [73]. One of the keys for improved models for social interaction analysis is to better understand the interaction context. This may include analyzing the speech: performing automated speech recognition [8, 132] followed by conversation modeling [125, 171].

Another way to incorporate the context into the model is to build a game-theoretic model of the interaction. For example, the Resistance game can be modeled as an asymmetric, cooperative, zero-sum sequential game. It is possible to make probabilistic inference about player roles based on the following observations only: who went on each "mission", the outcome of the mission (just success/fail outcome or a detailed number of success and fail votes). From this, we can build a classifier with an AUC of about 0.65 after the first round, which increases up to 0.8 by round 4 (under some assumptions about how likely a spy to vote for the mission failure). Humans, however, are not particularly good at probabilistic and Bayesian thinking [75]. And that shows in the Resistance dataset: if we use averaged guess of the villagers about other players' roles, human performance after the first two rounds is as low as 0.63 AUC and only reaches 0.74 AUC by the end of the game. Accuracy of individual guesses changing from 0.58 after the first two rounds to 0.64 after the last.

Finally, we can take an active approach to the problem. In the previous deception detection datasets (e.g., Columbia X-Cultural Deception Corpus [88]), subjects were participating in an interview with the script fixed by experiment designers. Interviewers, however, could ask unexpected follow-up questions to try catching a lie. This feature of the conversations could be improved and automated by building a conversational agent that would generate relevant questions designed to detect potential inconsistencies. Using conversational dialog models [21, 85] in conjunction with possible worlds approach from formal logic can be useful for this approach.

Customizing the models for cultural and individual features. One advantage of both the Resistance dataset and Global Political Deception Dataset is their cross-cultural nature. Subjects in both datasets come from a variety of cultural and socio-economic backgrounds across the world. This fact can be used to further improve the quality of predictions by analyzing differences between cultures and tailoring models for each subset of data. One downside of this approach is that not very large datasets become even smaller when divided into culture-specific subsets. We can try to overcome this problem by employing transfer learning techniques, for example adopting approaches from cross-lingual learning in Natural Language Processing field [119].

To go even further, we can try tailoring models for each individual. For example, when a polygraph test is administered, the protocol requires the test to start with a set of neutral questions designed to calibrate the subject's physiological response [1]. Almost all research on deception detection, starting with founding psychological works [46], operates under the assumption that all humans exhibit the same behavior when lying. While this is shown to be true to some extent (specifically, that some of the deception cues are common for all humans across all cultures [46]), it is natural to expect peculiar deceiving behavior from each person. In everyday life, we are usually much better at spotting changes in our friends and relatives than strangers. This is because we know how our close ones behave generally, and we can see if something is different about them. This observation can be applied in the Machine Learning approach by building a model specific for each subject. To be able to employ machine learning algorithms, we need to have a significant number of samples from both positive and negative classes. This, however, poses a challenge since without setting up a massive human study, it is hard to obtain enough deceitful statements from a single person. It is usually much easier to get truthful videos from a person (for example, in the context

of political deception, we can mine hundreds of hours of video with any public figure where there is no reason to suspect lies, e.g., personal interviews). Since the number of samples where the subject lies is still extremely limited, we cannot use a supervised learning approach in this setting. Instead, the automated deception detection can be posed as the anomaly detection problem [153, 110] or few-shot learning problem [44]. In this way, we don't need deceitful samples at the training stage. In addition to that, the act of lying can be considered anomalous with respect to everyday behavior.

Exploring other personality traits and interpersonal communication features. The other side of the deception coin is trust. One question that could be approached is: can we predict if a person A would trust person B after some interaction history. This problem could be posed in a variety of ways in terms of what is the input data for the algorithm: (1) can we predict trust by looking at person B's appearance (in other words, how trustworthy that person looks), (2) can we predict trust by looking at person A's appearance (in other words, how skeptical that person looks), (3) can we improve the prediction quality by looking at both participants and other group members.

The Resistance dataset contains ratings for trust between players. Additional relevant data can be mined from various TV game shows, where players have to trust each other to achieve common goal: "Golden Balls"[1] (UK), "Shafted"[2] (UK), "Friend or Foe?"[3] (USA), "Take It All"[4] (USA), "The Bank Job"[5] (UK). In these games, players have to participate in a generalized Prisoners Dilemma to maximize their winnings. This fact invites several research questions: given a video of the game (1) will player A trust player B, (2) will player B trust player A, (3) will they betray each others' trust?

Group Deception prediction In this thesis (Chapter 3) we predict whether a given player p_i is a spy in the game. We know, however, that there are several spies in the game. Suppose there are N players and K of them are spies, and we have predicted probabilities for each player $P(p_i)$ to be a spy. Then for a given group of

[1] https://www.imdb.com/title/tt1186336/
[2] https://www.imdb.com/title/tt0302191/
[3] https://www.imdb.com/title/tt0320860/
[4] https://www.imdb.com/title/tt2486556/
[5] https://www.imdb.com/title/tt2513534/

K people, the probability that all of them are spies is equal to

$$P(\bigcup_{j=1}^{K}\{p_{i_j}\}) = \prod_{j=1}^{K} P(p_{i_j}). \tag{6.1}$$

Thus, knowing K we can predict a group of spies to be the subset of K players with the highest joint probability.

Kumar et al. [86] observed that in the Resistance-game-like situation where there are two groups of adversarial agents, behavioral patterns differ for participants from different groups. For instance, spies tend to look at each other significantly less than at other players. On the other hand, villagers do not discriminate in attention between spies and other villagers (because they don't know other players' roles). This brings us to a hypothesis that knowing the number of spies in the game we can exploit the interaction dynamics within the group to predict on a subset level rather than on an individual level so that the probability of the group to be spies is higher than the simple joint probability (Eq. 6.1). This would not be the first attempt to do so, as Yu et al. [162] employed a similar attitudinal analysis to identify a cluster of adversarial players in a similar game, but this will be the first application of this approach to video data and the first one that takes temporal aspect into account.

This could be done, for example, by building a multi-layer network, where layers will represent a variety of automatically inferred social interactions such as gaze and its derivative (addressed speech, auditory attention) [11], relative expressed dominance [10], expressed relative nervousness, and dyadic impressions (Chapter 5).

Human decision anticipation Humans naturally desire to be able to predict other humans' actions and decisions beforehand. Some attempts at automatic prediction of humans' decisions we explored in a variety of settings such as social dilemma [68] and negotiations [111]. One more situation where such predictions could be of significant interest is gambling. One recent study [149] explored a possibility of automated predicting what action a poker player will make several seconds ahead of the action: fold, call, or raise. For this study the authors conducted an experiment where human subjects were playing a simplified version of the poker game: in each round, each player received a single card, then after a series of actions (raise, call, fold) the player with the highest card won the pot. The authors use automatically detected facial action units and random forest models to predict whether a player will fold or raise/call in his next action.

We hypothesise that model performance on this task can be greatly improved if group nature and the context of the interaction is taken into account: the way players looks at each other, the way players react to other players decisions can help us anticipate what each player will do even before that action happens. We can also enhance the predictive ability of the model by incorporating domain knowledge and observable information such as player's bank, game history including opened cards and player bets.

Interpretability of predictions. With the rise of deep learning, researchers started worrying about the lack of interpretability in deep neural networks, especially in high stakes applications such as medicine [161]. Interpretable models, however, come with various trade-offs with regards to performance or security [166]. There has been some efforts at providing *post hoc* interpretability [62]. Since detecting deception is usually crucial in high-stakes situations (in the court of law, in important negotiations, etc.), it is just as important to explain why the model predicts someone to be deceptive and provide levels of uncertainty in the prediction.

One more way to achieve interpretability in deception detection systems is to improve fine-grained deception analysis. Our proposed work (Chapter 3) makes predictions about deceptive behavior about the whole video. Instead, it would be advantageous to pinpoint specific statements or actions that are predicted to be false. The Box-of-Lies dataset [52] is developed for this kind of task: it contains dense annotations for deception in conversations.

6.4 Social and ethical implications.

Finally, we want to discuss some ethical and social implications of computational social science research. No technological advances stay isolated in laboratories or conference papers. They usually get implemented and applied in real life wherever it is economically feasible and prove useful: increasing income or reducing expenses. But more often than not, the straightforward applications of these advances disregard the social impact and ethical considerations. For example, facial recognition technologies are known to work significantly better on Caucasian people, which leads to racial discrimination [7]. Figure 6.4 shows how image super-resolution makes a Caucasian

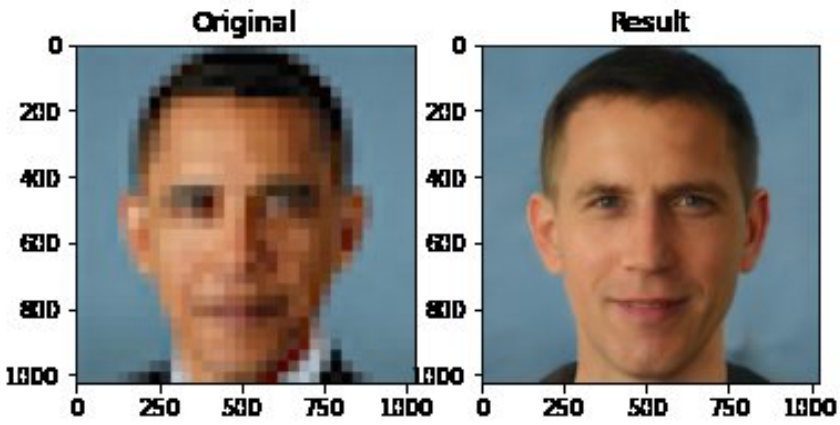


Figure 6.4: One of the examples of biased models: this image of unpixelated Barack Obama face was produced using models by Menon et al. [94] and is clearly showing a Caucasian man.

male from the pixelated photo of Barack Obama[1]. This problem also affects any technology that relies on facial analysis: detecting and tracking faces and detecting facial expressions and emotions. Audio analysis is not immune from this problem either, as well as textual analysis, which could be affected by a variety of variables, for example, if English is not a native language for a subject.

Ford [57] discusses legal use of the deception detection. Currently, in the U.S., 27 states and D.C. do not allow polygraph examination as evidence in criminal trials, while 22 states allow it. While Supreme Court decisions ruled denying lie detector evidence unconstitutional, the bar for the acceptance is very high as most legal scholars agree on the lack of reliability of polygraphs. One major flaw that is attributed to this type of lie detectors is that they were mostly tested on white males [6] and do not take into account a lot of confounding factors that can affect the predictions[2]. The same problem, obviously, concerns systems performing automated deception detection from videos. Current suggested models are trained on very small datasets (by modern Deep Learning standards). They lack the necessary diversity to be trustworthy in situations where human freedom or even life is at stake.

Even outside of the legal system, deception detection technology can be used rather recklessly. For example, in the recent TV show "Roswell: The Final Verdict"[3] producers used deception detection tools to analyze the video recordings of witness testimonies about alleged UFO crashes in Roswell, New Mexico in 1947. Some of the

[1] Image credit: https://twitter.com/Chicken3gg/status/1274314622447820801
[2] *State v. Anthony*, 100 N.M. at 738, 676 P.2d at 265
[3] https://www.imdb.com/title/tt14821454/

stories explored in the series were in the form "Person X told me there was [he/she saw] an object Y". And when the deception detection system produced a negative result for lying, producers of the show presented the finding as verification that there was indeed an object Y. Correct interpretation, however, would be at best that there are no signs of deception in that statement (which does not automatically make the statement true). At worst, that would mean Person X indeed told the witness about the object Y. This finding still does not rule out many possibilities from Person X being delusional to him plain lying to the witness.

This kind of irresponsible use of automated human behavior analysis tools (not just lie detectors but also personality traits detectors and others) can harm subjects. Publishing an unreliable detection of lies in some politician's speech can doom that politician's career. Unreliable results of personality traits prediction or mistakes in anxiety detection in children can result in biased attitudes from peers and teachers. But this can also bring harm to the field of research by discrediting the technology and scientific findings because of several high-profile wrong predictions.

These concerns raise the question: what to do? We see at least two measures that should be taken to avoid negative consequences. These two measures answer questions "how to develop models?" and "how to use models?"

How to develop models? First and foremost, we need to create more diverse datasets containing enough samples for each confounding variable: race, gender, culture, age, and others. The Resistancedataset and our Global Political Deception Dataset try to achieve that goal. This, however, is not enough. When one of Deep Learning most prominent researchers, Yann LeCun phrased the problem as "ML systems are biased when data is biased"[1], a lot of researchers replied that some Machine Learning models have inductive biases and making perfectly unbiased datasets is not just impossible but would not even be enough [135] (see also discussion under Yann LeCun's tweet). Building unbiased models, however, is a whole major research area and is outside of the scope of this dissertation.

How to use models? When using statistical models, it is vital to understand the limitations of technologies at hand (cf. Section 6.2) to make sure the model is used in a proper context: it is used for the same task the model was built for, it is used on data that has a similar distribution to the data the model was trained on, assumptions

[1] https://twitter.com/ylecun/status/1274782757907030016

that were put in the model are met. Another critical point is the interpretation of the results: if the model predicts that a subject is deceptive with a probability 0.6, that should not warrant immediate accusation in lying. Conversely, if the model shows 40% chance of lying, that does not necessarily mean the subject is telling the truth. Finally, it is essential not to conflate truthful statements about facts with the veracity of said facts.

Bibliography

[1] *The polygraph and lie detection*. National Academies Press, 2003.

[2] Mohamed Abouelenien, Veronica Pérez-Rosas, Rada Mihalcea, and Mihai Burzo. Deception Detection Using a Multimodal Approach. In *Proceedings of the 16th International Conference on Multimodal Interaction*, ICMI '14, pages 58–65, New York, NY, USA, 2014. ACM.

[3] Mohamed Abouelenien, Verónica Pérez-Rosas, Rada Mihalcea, and Mihai Burzo. Detecting deceptive behavior via integration of discriminative features from multiple modalities. *IEEE Transactions on Information Forensics and Security*, 12(5):1042–1055, 2017.

[4] Fabio Anselmi, Nicoletta Noceti, Lorenzo Rosasco, and Robert Ward. Genuine personality recognition from highly constrained face images. In Elisa Ricci, Samuel Rota Bulò, Cees Snoek, Oswald Lanz, Stefano Messelodi, and Nicu Sebe, editors, *Image Analysis and Processing – ICIAP 2019*, pages 421–431, Cham, 2019. Springer International Publishing.

[5] Oya Aran and Daniel Gatica-Perez. One of a kind: Inferring personality impressions in meetings. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, ICMI '13, pages 11–18, New York, NY, USA, 2013.

[6] American Medical Association. Polygraph. Council on Scientific Affairs. *JAMA*, 256(9):1172–1175, Sep 1986.

[7] Fabio Bacchini and Ludovica Lorusso. Race, again: how face recognition technology reinforces racial discrimination. *Journal of Information, Communication and Ethics in Society*, 17(3):321–335, August 2019.

[8] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations, 2020.

[9] Chongyang Bai, Maksim Bolonkin, Judee Burgoon, Chao Chen, Norah Dunbar, Bharat Singh, V. S. Subrahmanian, and Zhe Wu. Automatic long-term

deception detection in group interaction videos. In *Proceedings - 2019 IEEE International Conference on Multimedia and Expo, ICME 2019*, Proceedings - IEEE International Conference on Multimedia and Expo, pages 1600–1605. IEEE Computer Society, July 2019.

[10] Chongyang Bai, Maksim Bolonkin, Srijan Kumar, Jure Leskovec, Judee Burgoon, Norah Dunbar, and V. S. Subrahmanian. Predicting dominance in multi-person videos. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4643–4650. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[11] Chongyang Bai, Srijan Kumar, Jure Leskovec, Miriam Metzger, Jay F. Nunamaker, and V. S. Subrahmanian. Predicting the visual focus of attention in multi-person discussion videos. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4504–4510. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[12] Tadas Baltrusaitis, Amirali Bagher Zadeh, Yao Chong Lim, and Louis-Philippe Morency. Openface 2.0: Facial behavior analysis toolkit. In *13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 59–66, 2018.

[13] Tadas Baltrušaitis, Marwa Mahmoud, and Peter Robinson. Cross-dataset learning and person-specific normalisation for automatic action unit detection. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 06, pages 1–6, 2015.

[14] Daniele Bellutta, Youdinghuan Chen, Daveed Gartenstein-Ross, Chiara Pulice, Anja Subasic, and VS Subrahmanian. Understanding shifting triadic relationships in the al-qaeda/isis faction ecosystem. *IEEE Transactions on Computational Social Systems*, 2020.

[15] Cigdem Beyan, Francesca Capozzi, Cristina Becchio, and Vittorio Murino. Prediction of the leadership style of an emergent leader using audio and visual nonverbal features. *IEEE Transactions on Multimedia*, 20(2):441–456, 2018.

[16] Cigdem Beyan, Muhammad Shahid, and Vittorio Murino. Investigation of small group social interactions using deep visual activity-based nonverbal features. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18,

page 311–319, New York, NY, USA, 2018. Association for Computing Machinery.

[17] Cigdem Beyan, Andrea Zunino, Muhammad Shahid, and Vittorio Murino. Personality traits classification using deep visual activity-based nonverbal features of key-dynamic images. *IEEE Transactions on Affective Computing*, pages 1–1, 2019.

[18] Joan-Isaac Biel and Daniel Gatica-Perez. The youtube lens: Crowdsourced personality impressions and audiovisual analysis of vlogs. *IEEE Transactions on Multimedia*, 15(1):41–55, 2012.

[19] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022, March 2003.

[20] Stefano Boccaletti, Ginestra Bianconi, Regino Criado, Charo I Del Genio, Jesús Gómez-Gardenes, Miguel Romance, Irene Sendina-Nadal, Zhen Wang, and Massimiliano Zanin. The structure and dynamics of multilayer networks. *Physics reports*, 544(1):1–122, 2014.

[21] Antoine Bordes, Y-Lan Boureau, and Jason Weston. Learning end-to-end goal-oriented dialog. In *ICLR*. OpenReview.net, 2017.

[22] A. Jerry Bruce and Brian G. McDonald. Face recognition as a function of judgments of likability/unlikability. *The Journal of General Psychology*, 120(4):451–462, 1993. PMID: 8189210.

[23] Pradeep Buddharaju, Jonathan Dowdall, Panagiotis Tsiamyrtzis, Dvijesh Shastri, Ioannis Pavlidis, and Mark G. Frank. Automatic thermal monitoring system (ATHEMOS) for deception detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 1179 vol. 2–, June 2005.

[24] Peter Bull. Detecting lies and deceit: the psychology of lying and the implications for professional practice. Aldert Vrij. (2000) Wiley: Chichester. xv + 254 pp. ISBN 0-471-85316-X. *Journal of Community & Applied Social Psychology*, 16(2):166–167.

[25] David B. Buller and Judee K. Burgoon. Interpersonal deception theory. *Communication Theory*, 6(3):203–242, August 1996.

[26] Judee K. Burgoon, J. P. Blair, Tiantian Qin, and Jay F. Nunamaker. Detecting deception through linguistic analysis. In Hsinchun Chen, Richard Miranda, Daniel D. Zeng, Chris Demchak, Jenny Schroeder, and Therani Madhusudan, editors, *Intelligence and Security Informatics*, pages 91–101, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[27] Judee K. Burgoon, Lesa A. Stern, and Leesa Dillman. *Interpersonal Adaptation: Dyadic Interaction Patterns*. 01 1995.

[28] David M. Buss. *Evolutionary Psychology*. Routledge, February 2019.

[29] Alessia Celeghin, Matteo Diano, Arianna Bagnis, Marco Viola, and Marco Tamietto. Basic emotions in human neuroscience: Neuroimaging and beyond. *Frontiers in Psychology*, 8:1432, 2017.

[30] Gilberto Chávez-Martínez, Salvador Ruiz-Correa, and Daniel Gatica-Perez. Happy and agreeable? multi-label classification of impressions in social video. In *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia*, MUM '15, page 109–120, New York, NY, USA, 2015. Association for Computing Machinery.

[31] Gokul Chittaranjan and Hayley Hung. Are you a Werewolf? detecting deceptive roles and outcomes in a conversational role-playing game. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5334–5337, March 2010.

[32] David E. Clementson. Deceptively dodging questions: A theoretical note on issues of perception and detection. *Discourse & Communication*, 12(5):478–496, 2018.

[33] David E. Clementson. Effects of dodging questions: How politicians escape deception detection and how they get caught. *Journal of Language and Social Psychology*, 37(1):93–113, 2018.

[34] David E. Clementson. Truth bias and partisan bias in political deception detection. *Journal of Language and Social Psychology*, 37(4):407–430, 2018.

[35] Steven B. Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, August 1980.

[36] Mariya Davydenko, John M. Zelenski, Ana Gonzalez, and Deanna Whelan. Does acting extraverted evoke positive social feedback? *Personality and Individual Differences*, 159:109883, 2020.

[37] Manlio De Domenico, Albert Solé-Ribalta, Emanuele Cozzo, Mikko Kivelä, Yamir Moreno, Mason A Porter, Sergio Gómez, and Alex Arenas. Mathematical formulation of multilayer networks. *Physical Review X*, 3(4):041022, 2013.

[38] Bob de Ruiter and George Kachergis. The Mafiascum dataset: A large text corpus for deception detection. *arXiv preprint*, abs/1811.07851, 2018.

[39] Sergey Demyanov, James Bailey, Kotagiri Ramamohanarao, and Christopher Leckie. Detection of Deception in the Mafia Party Game. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, pages 335–342, New York, NY, USA, 2015.

[40] Bella M. Depaulo, James J. Lindsay, Brian E. Malone, Laura Muhlenbruck, Kelly Charlton, and Harris Cooper. Cues to deception. *PSYCHOLOGICAL BULLETIN*, 129(1):74–118, 2003.

[41] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[42] Mingyu Ding, An Zhao, Zhiwu Lu, Tao Xiang, and Ji-Rong Wen. Face-focused cross-stream network for deception detection in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[43] Patrick Doreian and Andrej Mrvar. Partitioning signed social networks. *Social Networks*, 31(1):1–11, 2009.

[44] Keval Doshi and Yasin Yilmaz. Any-shot sequential anomaly detection in surveillance videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[45] Norah E. Dunbar, Bradley Dorn, Mohemmad Hansia, Becky Ford, Matt Giles, Miriam Metzger, Judee K. Burgoon, Jay F. Nunamaker, and V. S. Subrahmanian. *Dominance in Groups: How Dyadic Power Theory Can Apply to Group Discussions*, pages 75–97. Springer International Publishing, Cham, 2021.

[46] Paul Ekman and Wallace Friesen. *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Consulting Psychologists Press, Palo Alto, 1978.

[47] Lucca Eloy, Angela E.B. Stewart, Mary Jean Amon, Caroline Reinhardt, Amanda Michaels, Chen Sun, Valerie Shute, Nicholas D. Duran, and Sidney D'Mello. Modeling team-level multimodal dynamics during multiparty collaboration. In *2019 International Conference on Multimodal Interaction*, ICMI '19, page 244–258, New York, NY, USA, 2019. Association for Computing Machinery.

[48] Hugo Jair Escalante, Heysem Kaya, Albert Ali Salah, Sergio Escalera, Yagmur Gucluturk, Umut Güçlü, Xavier Baró, Isabelle Guyon, Julio Jacques Junior, Meysam Madadi, Stéphane Ayache, Evelyne Viegas, Furkan Gurpinar, Achmadnoer Sukma Wicaksana, Cynthia C. S. Liem, Marcel A. J. van Gerven, and Rob van Lier. Explaining First Impressions: Modeling, Recognizing, and Explaining Apparent Personality from Videos. *IEEE Transactions on Affective Computing*, 2020. Accepted at Transaction on Affective Computing.

[49] Florian Eyben, Martin Wöllmer, and Björn Schuller. Opensmile: The munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, page 1459–1462, New York, NY, USA, 2010. Association for Computing Machinery.

[50] Giuseppe Facchetti, Giovanni Iacono, and Claudio Altafini. Computing global structural balance in large-scale signed social networks. *Proceedings of the National Academy of Sciences*, 108(52):20953–20958, 2011.

[51] Sheng Fang, Catherine Achard, and Séverine Dubuisson. Personality classification and behaviour interpretation: An approach based on feature categories. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, ICMI '16, page 225–232, New York, NY, USA, 2016. Association for Computing Machinery.

[52] Rada Mihalcea Felix Soldner, Verónica Pérez-Rosas. Box of lies: Multimodal deception detection in dialogues. *2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2019.

[53] Kexin Feng and Theodora Chaspari. A review of generalizable transfer learning in automatic emotion recognition. *Frontiers in Computer Science*, 2:9, 2020.

[54] Sinead V. Fernandes and Muhammad S. Ullah. Use of machine learning for deception detection from spectral and cepstral features of speech signals. *IEEE Access*, 9:78925–78935, 2021.

[55] Corneliu Florea, Laura Florea, Mihai Badea, Constantin Vertan, and Andrei Racoviteanu. Annealed label transfer for face expression recognition. 2018.

[56] Kory Floyd and Judee K. Burgoon. Reacting to nonverbal expressions of liking: A test of interaction adaptation theory. *Communication Monographs*, 66(3):219–239, 1999.

[57] Elizabeth B. Ford. Lie detection: Historical, neuropsychiatric and legal dimensions. *International Journal of Law and Psychiatry*, 29(3):159–177, 2006.

[58] Ohad Fried, Ayush Tewari, Michael Zollhöfer, Adam Finkelstein, Eli Shechtman, Dan B Goldman, Kyle Genova, Zeyu Jin, Christian Theobalt, and Maneesh Agrawala. Text-based editing of talking-head video. *ACM Trans. Graph.*, 38(4):68:1–68:14, July 2019.

[59] Giorgos Giannakakis, Matthew Pediaditis, Dimitris Manousos, Eleni Kazantzaki, Franco Chiarugi, Panagiotis G Simos, Kostas Marias, and Manolis Tsiknakis. Stress and anxiety detection using facial cues from videos. *Biomedical Signal Processing and Control*, 31:89–101, 2017.

[60] Mandar Gogate, Ahsan Adeel, and Amir Hussain. Deep learning driven multimodal fusion for automated deception detection. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–6, 2017.

[61] Lewis R Goldberg. The development of markers for the big-five factor structure. *Psychological assessment*, 4(1):26, 1992.

[62] Sachin Grover, Chiara Pulice, Gerardo I. Simari, and V. S. Subrahmanian. Beef: Balanced english explanations of forecasts. *IEEE Transactions on Computational Social Systems*, 6(2):350–364, 2019.

[63] Viresh Gupta, Mohit Agarwal, Manik Arora, Tanmoy Chakraborty, Richa Singh, and Mayank Vatsa. Bag-of-lies: A multimodal dataset for deception detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 83–90, 2019.

[64] Shlomo Hareli and Ursula Hess. What emotional reactions can tell us about the nature of others: An appraisal perspective on person perception. *Cognition and Emotion*, 24(1):128–140, 2010.

[65] Fritz Heider. Attitudes and cognitive organization. *Journal of Psychology*, 21:107–112, 1946.

[66] Fritz Heider. *The Psychology of Interpersonal Relations*. John Wiley & Sons, 1958.

[67] Julia Hirschberg, Stefan Benus, Jason Brenier, Frank Enos, Sarah Hoffman, Sarah Gilman, Cynthia Girand, Martin Graciarena, Andreas Kathol, Laura Michaelis, Bryan Pellom, Elizabeth Shriberg, and Andreas Stolcke. Distinguishing deceptive from non-deceptive speech. pages 1833–1836, 01 2005.

[68] Rens Hoegen, Giota Stratou, and Jonathan Gratch. Incorporating emotion perception into opponent modeling for social dilemmas. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '17, page 801–809, Richland, SC, 2017. International Foundation for Autonomous Agents and Multiagent Systems.

[69] Mohammed (Ehsan) Hoque, Matthieu Courgeon, Jean-Claude Martin, Bilge Mutlu, and Rosalind W. Picard. Mach: My automated conversation coach. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13, page 697–706, New York, NY, USA, 2013. Association for Computing Machinery.

[70] Martha J. Farah, J Hutchinson, Elizabeth A. Phelps, and Anthony Wagner. Functional MRI-Based Lie Detection: Scientific and Societal Challenges. *Nature Reviews Neuroscience*, 15:278–278, 2014.

[71] Uday Jain, Bozhao Tan, and Qi Li. Concealed knowledge identification using facial thermal imaging. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1677–1680, March 2012.

[72] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

[73] Roland M. Jones. Psychology and law: An empirical perspective, edited by neil brewer and kipling d. williams. guilford press, new york ny10012, usa, 2005. isbn 1-59385-122-7. *Criminal Behaviour and Mental Health*, 17(3):191–192, 2007.

[74] Jyoti Joshi, Hatice Gunes, and Roland Goecke. Automatic prediction of perceived traits using visual cues under varied situational context. In *2014 22nd International Conference on Pattern Recognition*, pages 2855–2860, 2014.

[75] Daniel Kahneman. *Thinking, fast and slow*. Macmillan, 2011.

[76] Manvi Kamboj, Christian Hessler, Priyanka Asnani, Kais Riani, and Mohamed Abouelenien. Multimodal political deception detection. *IEEE MultiMedia*, 28(1):94–102, 2021.

[77] Onno Kampman, Elham J. Barezi, Dario Bertero, and Pascale Fung. Investigating audio, video, and text fusion methods for end-to-end automatic personality prediction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 606–611, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[78] Hamid Karimi, Jiliang Tang, and Yanen Li. Toward end-to-end deception detection in videos. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 1278–1283, 2018.

[79] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

[80] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[81] Ahmet Alp Kindiroglu, Lale Akarun, and Oya Aran. Multi-domain and multi-task prediction of extraversion and leadership from meeting videos. *EURASIP Journal on Image and Video Processing*, 2017(1):77, Nov 2017.

[82] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[83] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P Gleeson, Yamir Moreno, and Mason A Porter. Multilayer networks. *Journal of complex networks*, 2(3):203–271, 2014.

[84] Daniel Kopev, Ahmed Ali, Ivan Koychev, and Preslav Nakov. Detecting deception in political debates using acoustic and textual features. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 652–659, 2019.

[85] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1378–1387. JMLR.org, 2016.

[86] Srijan Kumar, Chongyang Bai, V. S. Subrahmanian, and Jure Leskovec. Deception detection in group video conversations using dynamic interaction networks. In Ceren Budak, Meeyoung Cha, Daniele Quercia, and Lexing Xie, editors, *Proceedings of the Fifteenth International AAAI Conference on Web and Social Media, ICWSM 2021, held virtually, June 7-10, 2021*, pages 339–350. AAAI Press, 2021.

[87] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Signed networks in social media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, page 1361–1370, New York, NY, USA, 2010. Association for Computing Machinery.

[88] Sarah I. Levitan, Guzhen An, Mandi Wang, Gideon Mendels, Julia Hirschberg, Michelle Levine, and Andrew Rosenberg. Cross-cultural production and detection of deception from speech. In *Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection*, WMDD '15, page 1–8, New York, NY, USA, 2015. Association for Computing Machinery.

[89] Shan Li, Weihong Deng, and JunPing Du. Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2584–2593. IEEE, 2017.

[90] Yun-Shao Lin and Chi-Chun Lee. Predicting performance outcome with a conversational graph convolutional network for small group interactions. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8044–8048, 2020.

[91] Jon K. Maner and Andrew J. Menzel. *Evolutionary Social Psychology*, chapter 23. American Cancer Society, 2012.

[92] Candy Olivia Mawalim, Shogo Okada, Yukiko I. Nakano, and Masashi Unoki. Multimodal bigfive personality trait analysis using communication skill indices and multiple discussion types dataset. In Gabriele Meiselwitz, editor, *Social Computing and Social Media. Design, Human Behavior and Analytics*, pages 370–383, Cham, 2019. Springer International Publishing.

[93] Gary McKeown, Michel Valstar, Roddy Cowie, Maja Pantic, and Marc Schroder. The semaine database: Annotated multimodal records of emotionally colored conversations between a person and a limited agent. *IEEE Transactions on Affective Computing*, 3(1):5–17, 2012.

[94] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[95] Nicholas Michael, Mark Dilsizian, Dimitris Metaxas, and Judee K. Burgoon. Motion profiles for deception detection using visual cues. In *Proceedings of the 11th European Conference on Computer Vision: Part VI*, ECCV'10, pages 462–475, Berlin, Heidelberg, 2010.

[96] Rada Mihalcea and Carlo Strapparava. The lie detector: Explorations in the automatic recognition of deceptive language. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, page 309–312, USA, 2009. Association for Computational Linguistics.

[97] James Clyde Mitchell. *Social networks in urban situations: analyses of personal relationships in Central African towns*. Manchester University Press, 1969.

[98] Philipp Müller, Michael Xuelin Huang, and Andreas Bulling. Detecting low rapport during natural interactions in small groups from non-verbal behaviour. In *23rd International Conference on Intelligent User Interfaces*, pages 153–164, 2018.

[99] Aiko Murata, Hisamichi Saito, Joanna Schug, Kenji Ogawa, and Tatsuya Kameda. Spontaneous facial mimicry is enhanced by the goal of inferring

emotional states: evidence for moderation of "automatic" mimicry by higher cognitive processes. *PloS one*, 11(4):e0153128, 2016.

[100] Preslav Nakov, Alberto Barrón-Cedeño, Tamer Elsayed, Reem Suwaileh, Lluís Màrquez, Wajdi Zaghouani, Pepa Atanasova, Spas Kyuchukov, and Giovanni Da San Martino. Overview of the clef-2018 checkthat! lab on automatic identification and verification of political claims. In Patrice Bellot, Chiraz Trabelsi, Josiane Mothe, Fionn Murtagh, Jian Yun Nie, Laure Soulier, Eric SanJuan, Linda Cappellato, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 372–387, Cham, 2018. Springer International Publishing.

[101] Hanen Nasri, Wael Ouarda, and Adel M. Alimi. Relidss: Novel lie detection system from speech signal. In *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*, pages 1–8, 2016.

[102] Laurent Nguyen, Denise Frauendorfer, Marianne Mast, and Daniel Gatica-Perez. Hire me: Computational inference of hirability in employment interviews based on nonverbal behavior. *Multimedia, IEEE Transactions on*, 16:1018–1031, 06 2014.

[103] Fumio Nihei, Yukiko I. Nakano, Yuki Hayashi, Hung-Hsuan Hung, and Shogo Okada. Predicting influential statements in group discussions using speech and head motion information. In *Proceedings of the 16th International Conference on Multimodal Interaction*, ICMI '14, page 136–143, New York, NY, USA, 2014. Association for Computing Machinery.

[104] Richard E. Nisbett and Michael Smith. Predicting interpersonal attraction from small samples: A reanalysis of newcomb's acquaintance study. *Social Cognition*, 7(1):67–73, 03 1989. Copyright - © 1989 Guilford Publications Inc; Last updated - 2018-10-15; CODEN - SOCOEE.

[105] Shogo Okada, Oya Aran, and Daniel Gatica-Perez. Personality trait classification via co-occurrent multiparty multimodal event discovery. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, pages 15–22, New York, NY, USA, 2015.

[106] Shogo Okada, Laurent Son Nguyen, Oya Aran, and Daniel Gatica-Perez. Modeling dyadic and group impressions with intermodal and interperson features.

ACM Transactions on Multimedia Computing, Communications, and Applications, 15(1s):13:1–13:30, January 2019.

[107] Shogo Okada, Laurent Son Nguyen, Oya Aran, and Daniel Gatica-Perez. Modeling dyadic and group impressions with intermodal and interperson features. *ACM Trans. Multimedia Comput. Commun. Appl.*, 15(1s), January 2019.

[108] Shogo Okada, Yoshihiko Ohtake, Yukiko I. Nakano, Yuki Hayashi, Hung-Hsuan Huang, Yutaka Takase, and Katsumi Nitta. Estimating communication skills using dialogue acts and nonverbal features in multiple discussion datasets. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, ICMI '16, page 169–176, New York, NY, USA, 2016. Association for Computing Machinery.

[109] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[110] Guansong Pang, Cheng Yan, Chunhua Shen, Anton van den Hengel, and Xiao Bai. Self-trained deep ordinal regression for end-to-end video anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[111] Sunghyun Park, Jonathan Gratch, and Louis-Philippe Morency. I already know your answer: Using nonverbal behaviors to predict immediate outcomes in a dyadic negotiation. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction*, ICMI '12, page 19–22, New York, NY, USA, 2012. Association for Computing Machinery.

[112] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 41.1–41.12. BMVA Press, September 2015.

[113] Ioannis Pavlidis, Norman Eberhardt, and James Levine. Human behaviour: Seeing through the face of deception. *Nature*, 415:35, 2002.

[114] Matthew Pediaditis, Giorgos A. Giannakakis, Franco Chiarugi, Dimitris Manousos, Anastasia Pampouchidou, Eirini Christinaki, Galateia Iatraki, Eleni Kazantzaki, Panagiotis G. Simos, Kostas Marias, and Manolis Tsiknakis. Extraction of facial features as indicators of stress and anxiety. *2015 37th Annual*

International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2015.

[115] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[116] Verónica Pérez-Rosas and Rada Mihalcea. Cross-cultural deception detection. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 440–445, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[117] Verónica Pérez-Rosas and Rada Mihalcea. Experiments in open domain deception detection. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1120–1125, Lisbon, Portugal, September 2015.

[118] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the Fisher kernel for large-scale image classification. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 143–156, Berlin, Heidelberg, 2010.

[119] Matúš Pikuliak, Marián Šimko, and Mária Bieliková. Cross-lingual learning for text processing: A survey. *Expert Systems with Applications*, 165:113765, 2021.

[120] Víctor Ponce-López, Baiyu Chen, Marc Oliu, Ciprian Corneanu, Albert Clapés, Isabelle Guyon, Xavier Baró, Hugo Jair Escalante, and Sergio Escalera. Chalearn lap 2016: First round challenge on first impressions - dataset and results. In Gang Hua and Hervé Jégou, editors, *Computer Vision – ECCV 2016 Workshops*, pages 400–418, Cham, 2016. Springer International Publishing.

[121] Albert Pumarola, Antonio Agudo, Aleix M. Martinez, Alberto Sanfeliu, and Francesc Moreno-Noguer. GANimation: One-shot anatomically consistent facial animation. *International Journal of Computer Vision*, 128(3):698–713, August 2019.

[122] Verónica Pérez-Rosas, Mohamed Abouelenien, Rada Mihalcea, and Mihai Burzo. Deception Detection Using Real-life Trial Data. In *Proceedings of the*

2015 ACM on International Conference on Multimodal Interaction, ICMI '15, pages 59–66, New York, NY, USA, 2015. ACM. event-place: Seattle, Washington, USA.

[123] Bashar Rajoub and Reyer Zwiggelaar. Thermal Facial Analysis for Deception Detection. *Information Forensics and Security, IEEE Transactions on*, 9:1015–1023, 2014.

[124] Keith Rayner. Eye movements and attention in reading, scene perception, and visual search. *The Quarterly Journal of Experimental Psychology*, 62(8):1457–1506, 2009.

[125] Pengjie Ren, Zhumin Chen, Christof Monz, Jun Ma, and Maarten de Rijke. Thinking globally, acting locally: Distantly supervised global-to-local knowledge selection for background based conversation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8697–8704, Apr. 2020.

[126] Stephen Reysen. Construction of a new scale: The reysen likability scale. *Social Behavior and Personality*, 33(2):201–208, 2005.

[127] Stephen Reysen. A new predictor of likeability: Laughter. *North American Journal of Psychology*, 8(2):373–382, Jun 2006. Copyright - Copyright North American Journal of Psychology Jun/Jul 2006; Last updated - 2011-06-17.

[128] Rodrigo Rill-García, Hugo Jair Escalante, Luis Villaseñor-Pineda, and Verónica Reyes-Meza. High-level features for multimodal deception detection in videos. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1565–1573, 2019.

[129] Selwin Gabriel Samuel, Tanushree Chatterjee, Himadri Thapliyal, and Priyanka Kacker. Facial psychophysiology in forensic investigation: A novel idea for deception detection. *Journal of forensic dental sciences*, 11(2):90–94, 2019. 32082044[pmid].

[130] Dairazalia Sanchez-Cortes, Oya Aran, Marianne Schmid Mast, and Daniel Gatica-Perez. A nonverbal behavior approach to identify emergent leaders in small groups. *IEEE Transactions on Multimedia*, 14(3):816–832, 2011.

[131] Dairazalia Sanchez-Cortes, Oya Aran, Marianne Schmid Mast, and Daniel Gatica-Perez. A nonverbal behavior approach to identify emergent leaders in small groups. *IEEE Transactions on Multimedia*, 14(3):816–832, June 2012.

[132] George Saon, Gakuto Kurata, Tom Sercu, Kartik Audhkhasi, Samuel Thomas, Dimitrios Dimitriadis, Xiaodong Cui, Bhuvana Ramabhadran, Michael Picheny, Lynn-Li Lim, Bergul Roomi, and Phil Hall. English conversational telephone speech recognition by humans and machines. In *Proc. Interspeech 2017*, pages 132–136, 2017.

[133] John S. Seiter, Harry Weger Jr., Harold J. Kinzer, and Andrea Sandry Jensen. Impression management in televised debates: The effect of background nonverbal behavior on audience perceptions of debaters' likeability. *Communication Research Reports*, 26(1):1–11, 2009.

[134] Umut Mehmet Sen, Veronica Perez-Rosas, Berrin Yanikoglu, Mohamed Abouelenien, Mihai Burzo, and Rada Mihalcea. Multimodal deception detection using real-life trial data. *IEEE Transactions on Affective Computing*, pages 1–1, 2020.

[135] Deven Santosh Shah, H. Andrew Schwartz, and Dirk Hovy. Predictive biases in natural language processing models: A conceptual framework and overview. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5248–5264, Online, July 2020. Association for Computational Linguistics.

[136] Shan Lu, G. Tsechpenakis, D. N. Metaxas, M. L. Jensen, and J. Kruse. Blob analysis of the head and hands: A method for deception detection. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, pages 20c–20c, 2005.

[137] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[138] Anastasis Stathopoulos, Ligong Han, Norah Dunbar, Judee K. Burgoon, and Dimitris Metaxas. Deception detection in videos using robust facial features. In Kohei Arai, Supriya Kapoor, and Rahul Bhatia, editors, *Proceedings of the Future Technologies Conference, FTC 2020, Volume 3*, Advances in Intelligent Systems and Computing, pages 668–682, Germany, 2021. Springer Science and Business Media Deutschland GmbH.

[139] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In Maosong Sun, Xuanjing Huang, Heng Ji, Zhiyuan Liu, and Yang Liu, editors, *Chinese Computational Linguistics*, pages 194–206, Cham, 2019. Springer International Publishing.

[140] Yla R. Tausczik and James W. Pennebaker. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of Language and Social Psychology*, 29(1):24–54, 2010.

[141] Linda Tickle-Degnen and Robert Rosenthal. The nature of rapport and its nonverbal correlates. *Psychological Inquiry*, 1(4):285–293, 1990.

[142] Jessica L. Tracy and Daniel Randles. Four models of basic emotions: A review of ekman and cordaro, izard, levenson, and panksepp and watt. *Emotion Review*, 3(4):397–405, 2011.

[143] Gabriel Tsechpenakis, Dimitris Metaxas., Mark Adkins, John Kruse, Judee K. Burgoon, Matthew L. Jensen, Thomas O. Meservy, Douglas P. Twitchell, Amit Deokar, and Jay F. Nunamaker. HMM-Based Deception Recognition from Visual Cues. In *2005 IEEE International Conference on Multimedia and Expo*, pages 824–827, July 2005.

[144] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. MoCoGAN: Decomposing motion and content for video generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1526–1535, 2018.

[145] Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962v2*, 2019.

[146] Arjon Turnip, M Faizal Amri, Hanif Fakrurroja, Artha Ivonita Simbolon, M. Agung Suhendra, and Dwi Esti Kusumandari. Deception detection of eeg-p300 component classified by svm method. In *Proceedings of the 6th International Conference on Software and Computer Applications*, ICSCA '17, page 299–303, New York, NY, USA, 2017. Association for Computing Machinery.

[147] Reeve Vanneman, James Noon, Mitali Sen, Sonalde Desai, and Abusaleh Shariff. Social networks in india: Caste, tribe, and religious variations. In *Proceedings of the Annual Meeting of the Population Association of America*, pages 112–145. University of Maryland College Park, 2006.

[148] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018.

[149] Doratha Vinkemeier, Michel Valstar, and Jonathan Gratch. Predicting folds in poker using action unit detectors and decision trees. In *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, pages 504–511, 2018.

[150] Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

[151] Aldert Vrij. *Detecting lies and deceit: pitfalls and opportunities*. Wiley Series in the Psychology of Crime, Policing and Law. Wiley, 2008.

[152] Lezi Wang, Chongyang Bai, Maksim Bolonkin, Judee K Burgoon, Norah E Dunbar, VS Subrahmanian, and Dimitris Metaxas. Attention-based facial behavior analytics in social communication. In *Detecting Trust and Deception in Group Interaction*, pages 123–137. Springer, 2021.

[153] Lin Wang, Fuqiang Zhou, Zuoxin Li, Wangxia Zuo, and Haishu Tan. Abnormal event detection in videos using hybrid spatio-temporal autoencoder. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2276–2280, 2018.

[154] William Yang Wang. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, 2017.

[155] Yanbang Wang, Pan Li, Chongyang Bai, and Jure Leskovec. TEDIC: neural modeling of behavioral patterns in dynamic social interaction networks. In Jure Leskovec, Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia, editors, *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 693–705. ACM / IW3C2, 2021.

[156] Lara Warmelink, Aldert Vrij, Samantha Mann, Sharon Leal, Dave Forrester, and Ronald P. Fisher. Thermal Imaging as a Lie Detection Tool at Airports. *Law and Human Behavior*, 35(1):40–48, 2011.

[157] Leah Windsor, Alistair Windsor, Miriam Van Mersbergen, George Deitz, Allison Sulkowski, and Lily Walljasper. A multimodal approach to analyzing deception in politics. 11 2019.

[158] Zhe Wu, Bharat Singh, Larry S. Davis, and V. S. Subrahmanian. Deception detection in videos. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1695–1702. AAAI Press, 2018.

[159] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[160] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[161] Chang Ho Yoon, Robert Torrance, and Naomi Scheinerman. Machine learning in medicine: should the pursuit of enhanced interpretability be abandoned? *Journal of Medical Ethics*, 2021.

[162] Dian Yu, Yulia Tyshchuk, Heng Ji, and William Wallace. Detecting deceptive groups using conversations and network analysis. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 857–866, July 2015.

[163] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *European conference on computer vision*, pages 151–158. Springer, 1994.

[164] Le Zhang, Songyou Peng, and Stefan Winkler. Persemon: A deep network for joint analysis of apparent personality, emotion and their relationship. *IEEE Transactions on Affective Computing*, pages 1–1, 2019.

[165] Lingyu Zhang, Indrani Bhattacharya, Mallory Morgan, Michael Foley, Christoph Riedl, Brooke Welles, and Richard Radke. Multiparty visual co-occurrences for estimating personality traits in group meetings. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

[166] Xinyang Zhang, Ningfei Wang, Hua Shen, Shouling Ji, Xiapu Luo, and Ting Wang. Interpretable deep learning under fire. In *Proceedings of the 29th USENIX Security Symposium*, Proceedings of the 29th USENIX Security Symposium, pages 1659–1676. USENIX Association, 2020.

[167] Yanxia Zhang, Jeffrey Olenick, Chu-Hsiang Chang, Steve W. J. Kozlowski, and Hayley Hung. Teamsense: Assessing personal affect and group cohesion in small teams through dyadic interaction and behavior analysis with wearable sensors. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(3), September 2018.

[168] Zhi Zhang, Vartika Singh, Thomas E. Slowe, Sergey Tulyakov, and Venugopal Govindaraju. Real-time automatic deceit detection from involuntary facial expressions. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, June 2007.

[169] Hang Zhou, Yu Liu, Ziwei Liu, Ping Luo, and Xiaogang Wang. Talking face generation by adversarially disentangled audio-visual representation. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[170] Lina Zhou, Judee K. Burgoon, and Douglas P. Twitchell. A longitudinal analysis of language behavior of deception in e-mail. In Hsinchun Chen, Richard Miranda, Daniel D. Zeng, Chris Demchak, Jenny Schroeder, and Therani Madhusudan, editors, *Intelligence and Security Informatics*, pages 102–110, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[171] Henghui Zhu, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. Who did they respond to? conversation structure modeling using masked hierarchical transformer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9741–9748, Apr. 2020.

[172] Oya Çeliktutan and Hatice Gunes. Continuous prediction of perceived traits and social dimensions in space and time. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 4196–4200, 2014.

[173] Oya Çeliktutan and Hatice Gunes. Automatic prediction of impressions in time and across varying context: Personality, attractiveness and likeability. *IEEE Transactions on Affective Computing*, 8(1):29–42, 2017.