University of Padua

Doctorate Degree in Brain, Mind and Computer Science

Curriculum in Computer Science

Human Interactions in Cybersecurity: Threats and Opportunities

Candidate

Matteo Cardaioli

Supervisor

Prof. Mauro Conti

University of Padova, Italy

Co-supervisor

Prof. Giuseppe Sartori

University of Padova, Italy

Acknowledgments

I want to express my sincere gratitude to my advisor Prof. Mauro Conti for his valuable support, guidance, encouragement, and enthusiasm in guiding me in my research career. I hope to continue our collaboration in the future and learn even more from him: keep pushing hard!. I sincerely thank my co-supervisors, Prof. Giuseppe Sartori. His experience, curiosity, and motivation allowed me to work with many interdisciplinary groups and apply my knowledge to different research fields. I truly thank Prof. Gene Tsudik for remote-hosting me at the University of California, Irvine and giving me insight, continuous guidance, and new ideas to carry on my research.

A special thanks to all my GFT colleagues that supported me in managing my Ph.D. from the very beginning. In particular, Maurizio Ferraris and Luca Moroni have always been available, enthusiastic, proactive and have allowed me to undertake this journey. A big thank also to Alessandro Gnata and Marco Zambon, who have supported me in managing work and research with patience and professionalism.

Many thanks to my parents, Claudio and Patrizia, who encouraged and helped me follow my ambitions.

Special thanks to my future wife, Alice, for her continuous support, patience, and motivation to follow what I like most. I am joyful to share all our achievements.

A great thank to my officemates and SPRITZ group colleagues (Alessio, Ankit, Denis, Eleonora, Federico, Gabriele, Luca PJ, prof. Pasa, Pallavi, Pier Paolo, and Stefano) for their kind help, useful remarks towards my research and valuable aperitives.

Matteo Cardaioli
Padova, March 30, 2022

You know what? I can.
B.S.

Abstract

Over the years, many cybersecurity breaches have been attributed to human error, considering human factors as one of the weakest links in the security chain. In fact, human factors are exploited by cybercriminals, causing significant losses of money and reputation to organizations. According to Verizon's 2021 Data Breach Investigations, 85% of breaches involved a human element, while 61% involved stolen or compromised credentials, causing an average breach cost of more than $3 million. To prevent cyberattacks, organizations focus on training employees and developing new policies, while also trying to maintain a balance between the complexity of security systems and their usability. However, the unpredictability of human behavior, the fast evolution of the digital world, and the increasing availability of technological resources for cybercriminals pose new and evolving cybersecurity challenges in anticipating both cyber threats in new environments and the rise of new threats in systems considered secure to date. On the other hand, the complexity and uniqueness of human behavior give new opportunities for designing new solutions to mitigate threats, improving the security of organizations and users.

In this thesis, we investigate human interactions and cybersecurity, focusing on two main aspects: (i) developing new attacks, based on human interaction, against existing and consolidated authentication methods (i.e., PIN pads), and (ii) proposing new methods leveraging human behavior in multiple contexts to enhance the security of users and organizations. The first part of this thesis demonstrates the effectiveness of three attacks against the security of PIN-based authentication systems, focusing on Automated Teller Machines (ATMs) PIN pads. ATMs have become an indispensable part of the banking ecosystem such that according to the European Central Bank, in 2019 only in Europe, more than 11 billion withdrawal and deposit transactions were made. In particular, we show how ATM PIN pads are exposed to security threats related to human factors even if users have policy-compliant behaviors. We analyze different attack scenarios depending on the sources of information available to the attacker (e.g., video, audio, thermal, typing style). The results show that in the worst-case scenario for the victim, our attacks can reconstruct up to 94% of the 5-digit PINs typed within three attempts.

In the second part of this thesis, we show how the variability and unpredictability of human behavior can be exploited to increase the security of systems and users. We develop new human-based approaches focusing on three different contexts: (i) new methods for bot detection in social networks (i.e., Twitter) relying on the stylistic consistency of posts over time, (ii) a new framework for identifying fake and genuine expressions from videos, and (iii) a new de-authentication method based on the detection of physically blurred faces. Results demonstrate the efficacy of the proposed approaches, achieving an F1-score up to 98% in human-bot detection, an accuracy up to 90% in fake sadness detection, and accuracy in de-authenticating users up to 100% under 3 seconds of grace period.

This thesis highlights the need for more effort in designing security solutions that focus on human factors, showing the direction for further investigation in analyzing human interactions in cybersecurity.

Contents

Chapter 1

Introduction

With the expansion of digital technologies, society have experienced a dramatic increase in opportunities for criminal activity. Almost every aspect of human behavior is managed, recorded, and tracked in the digital realm: e.g., communications, movements, payments. In this context, information security is playing an increasingly critical role by balancing the protection of confidentiality, integrity, and availability of data while maintaining effective use of organizations' systems and their usability. However, organizations invest more in security technology (e.g., firewalls, encryption, secure access devices, hard passwords) than they do in considering the human factor for preventing cyberattacks [189].

Cybersecurity breaches have often been linked to human error over the years, considering human factors as one of the weakest points in the security chain. Human error can be defined as actions taken unintentionally (or lack of action) by users that lead to, spread, or allow a security breach. A variety of actions contribute to the risk, from downloading malware-infected attachments to forgetting to use strong passwords. Also work environments can force users to take shortcuts that expose them to cyber threats. A well known example is the reuse of credentials across multiple applications in order to avoid remembering (or worse transcribing) passwords [98]. Addae et al. [3] suggested that increasing the usability of cybersecurity mechanisms can greatly encourage users to adopt better security controls and behaviors. However, human error is still a critical factor. Verizon's 2021 Data Breach Investigations Report[1], highlighted that 85% of breaches involved a human

[1] https://www.verizon.com/business/resources/reports/dbir/2021

element, and 61% were due to stolen or compromised user credentials. Further, the average cost of such breaches was estimated at $3.33 million.

Although human error remains a primary issue in mitigating cyber incidents, with organizations focusing on employee training and policies, human factors represent an evolving cyber security challenge. In particular, to anticipate possible threats, it is necessary to predict potential security flaws even when behaviors are compliant with standards. An example of these threats is given by the so-called side-channel attacks, where the criminal exploits the indirect effects of the system to collect information. On the other side, human behavior can also be exploited to make existing systems more secure or implement ad hoc solutions to mitigate the emergence of new threats.

In this context, the rapid evolution of technology makes, on the one hand, difficult to predict how human interactions with the digital world will evolve (e.g., the metaverse) and, on the other hand, provides attackers with increasingly powerful tools to perpetrate new attacks. This poses several challenges in both anticipating cyber threats in new environments and the rise of new threats in systems considered secure to date.

1.1 Research Motivation and Contribution

This thesis aims to investigate human interactions and cybersecurity, focusing on two main aspects: (i) showing the feasibility of new attacks, based on human interaction, against existing and consolidated authentication methods (i.e., PIN pads), and (ii) developing new methods that leverage human behavior in multiple contexts to enhance the security of users and organizations.

In this thesis, some passages have been quoted verbatim, and some figures have been reused from the works [15, 41, 42, 43, 44], all coauthored by the author of this thesis.

1.1.1 In-Security Through Human Interaction Analysis: the PIN Pad Case

Financial institutions represent one of the most profitable targets for cybercriminals. The International Monetary Fund estimated that growing cyber-threats represent a serious issue to financial institutions' profits, ranging from 9% up to 50% in worst-case scenarios [35]. Although the spread of FinTech solutions has significantly transformed the banking ecosystem in recent years (e.g., online banking, new payment systems, cryptocurrencies), some elements have had smaller impacts, effectively resisting this new technological wave. Among the most notable exponents of this category are PINs. The combination of security, a very simple user experience, the limited technological resources required, and their ubiquity have resulted in PINs (and passwords) remaining the most popular authentication factor to date.

The security of PIN-based authentication systems is essentially based on the difficulty of an attacker to succeed in guessing the correct sequence of numbers that constitute the secret. Intuitively, longer PINs are harder to attack, but they are also harder to remember. This trade off is an early point where PIN-based security comes into contact with human factors, posing the first challenges. What is the best balance between PIN length and PIN usability? Is it safer to provide randomly generated PINs, with the risk of them being written down somewhere, or to allow users to choose their own secret? To answer the first question, according to ISO 9564-1 [133] (the standard for PIN management in financial services), the issuer can assign a PIN up to twelve digits. Still, for usability reasons is recommended not to exceed six digits in length. Regarding the dualism between random PINs and user-chosen PINs, several studies have shown that the latter suffer from significant bias, which compromises their security. Textual passwords have been analyzed for bias starting with Morris and Thompson [191], and confirmed in many studies since [241]. Similarly, Bonneau et al. [33] demonstrated that, in the absence of denied PIN lists, a lost or stolen wallet will be vulnerable to theft up to 8.9% of the time, with birthday-based guessing the most effective technique.

Other concrete threats to PIN security related to victim behavior are so-called shoulder-surfing attacks, in which the attacker tries to spy on the victim while typing the PIN to steal their secret. Financial institutions and standards (e.g., ISO 9564-1 [133]) provide rules of conduct and directions to mitigate shoulder-surfing attacks: hiding the PIN while typing, making sure no one watches the screen, PIN digits must not be displayed, and the duration and type of feedback sound emitted by the pressed keys must be the same for each key.

Assuming that users and institutions follow the best standards to ensure PIN security (e.g., random PIN, cover the PIN pad), in this part of the thesis, we want to study whether other factors related to human behavior can affect the security of PIN-based authentication devices.

1.1.1.1 Your PIN Sounds Good: Interkeystroke-timing Based Attacks on PINs

Personal Identification Numbers are widely used as the primary authentication method for Automated Teller Machines (ATMs) and Point of Sale (PoS). ATM and PoS typically mitigate attacks, including shoulder-surfing, by displaying dots on their screen rather than PIN digits and by obstructing the view of the keypad. Further, ISO 9564-1 [133] requires that PIN digits should not be identified by different sound characteristics or durations. Indeed, PIN entry systems are developed to provide the same audio feedback (e.g., same tone, same duration) for all keys to achieve a balance between security and usability.

Contributions: In Chapter 2, we explore several sources of information leakage from common ATM and PoS installations that the adversary can leverage to reduce the number of attempts necessary to guess a PIN. Specifically, we evaluate how the adversary can leverage audio feedback generated by a standard ATM keypad to infer accurate inter-keystroke timing information, and how these timings can be used to improve attacks based on the observation of the user's typing behavior, partial PIN information, and attacks based on thermal cameras. Our results show that inter-keystroke timings can be extracted from audio feedback far more accurately than from previously explored sources (e.g., videos). In our experiments, this increase in accuracy translated to a meaningful increase in guessing performance. Further, various combinations of these sources of information allowed us to guess between 44% and 89% of the 4-digit PINs within 5 attempts. Finally, we observed that based on the type of information available to the adversary, and contrary to common knowledge, uniform PIN selection is not necessarily the best strategy. We consider these results relevant and important, as they highlight a real threat to any authentication system that relies on PINs.

1.1.1.2 Hand me your PIN: Inferring PINs from Videos of Users Typing with a Covered Hand

Automated Teller Machines (ATMs) represent the most used system for withdrawing cash. The European Central Bank reported more than 11 billion cash withdrawals and loading/unloading transactions on the European ATMs in 2019. Although ATMs have undergone various technological evolutions, PINs are still the most common authentication method for these devices. Unfortunately, the PIN mechanism is vulnerable to shoulder-surfing attacks performed via hidden cameras installed near the ATM to catch the PIN pad. To overcome this problem, people get used to covering the typing hand

with the other hand. While such users probably believe this behavior is safe enough to protect against mentioned attacks, there is no clear assessment of this countermeasure in the scientific literature.
Contributions: In Chapter 3, we propose a novel attack to reconstruct PINs entered by victims covering the typing hand with the other hand. We consider the setting where the attacker can access an ATM PIN pad of the same brand/model as the target one. Afterward, the attacker uses that model to infer the digits pressed by the victim while entering the PIN. Our attack owes its success to a carefully selected deep learning architecture that can infer the PIN from the typing hand position and movements. We run a detailed experimental analysis including 58 users. With our approach, we can guess 30% of the 5-digit PINs within three attempts – the ones usually allowed by ATM before blocking the card. We also conducted a survey with 78 users that managed to reach an accuracy of only 7.92% on average for the same setting. Finally, we evaluate a shielding countermeasure that proved to be rather inefficient unless the whole keypad is shielded.

1.1.1.3 *$\mathcal{P}$inDrop*: Acoustic Side-Channel Attacks on ATM PIN Pads

Attacks that exploit video recordings of PIN pads have become more widespread over time due to their great simplicity of use. However, this kind of attack requires placing a camera directly on-site, limiting its applicability in many real-world contexts. Moreover, the use of protective shields on the PIN pad or complete coverage by the user are disincentives that further restrict the effectiveness of video-based attacks. Although these elements protect PINs from visual attacks, acoustic emanations from the PIN pad itself open the door for another attack type.
Contributions: In Chapter 4, we show the feasibility of an acoustic side-channel attack (called *$\mathcal{P}$inDrop*) to reconstruct PINs by profiling acoustic signatures of individual keys of a PIN pad. We demonstrate the practicality of *$\mathcal{P}$inDrop* via two sets of data collection experiments involving two commercially available metal PIN pad models and 58 participants who entered a total of 5,800 5-digit PINs. We simulated two realistic attack scenarios: (1) a microphone placed near the ATM (0.3 meters away) and (2) a real-time attacker (with a microphone) standing in the queue at a common courtesy distance of 2 meters. In the former case, we show that *$\mathcal{P}$inDrop* recovers 96% of 4-digit, and up to 94% of 5-digits, PINs. Whereas, at 2 meters away, it recovers up to 57% of 4-digit, and up to 39% of 5-digit PINs in three attempts. We believe that these results are both significant and worrisome.

1.1.2 Securing Securing Computer-Human Interaction

Cybersecurity research is increasingly focusing on behavioral aspects [161]. The unpredictability of behavior and actions makes humans a key element in designing secure systems. As discussed before, making users aware of risks and promoting policies and standards is not always sufficient. However, the advancement of technology, which brings ever-increasing computing capabilities and the rise of new ecosystems (e.g., social networks) and habits, combined with the uniqueness of human behavior, may open new challenges and opportunities in cybersecurity. The uniqueness of human behavior, for example, is a feature that has enabled the development of new security layers in authentication systems (e.g., behavioral biometrics).

This part of the thesis shows how human factors can be leveraged to develop more secure systems with holistic and non-intrusive solutions. In particular, we focus on three application fields where the human factor is significant but still under-exploited in a security context: bot detection in social networks, fake emotion detection, and de-authentication.

1.1.2.1 It's a Matter of Style: Detecting Social Bots through Writing Style Consistency

Social bots are computer algorithms able to produce content and interact with other users on social media autonomously, trying to emulate and possibly influence humans' behavior. Indeed, bots are largely employed for malicious purposes, like spreading disinformation and conditioning electoral campaigns. Nowadays, bots' capability of emulating human behaviors has become increasingly sophisticated, making their detection harder.
Contributions: In Chapter 5, we aim at recognizing bot-driven accounts by evaluating the consistency of users' writing style over time. In particular, we leverage the intuition that while bots compose posts according to fairly deterministic processes, humans are influenced by subjective factors (e.g., emotions) that can alter their writing style. To verify this assumption, by using stylistic consistency indicators, we characterize the writing style of more than 12,000 among bot-driven and human-operated Twitter accounts and find that statistically significant differences can be observed between the different types of users. Thus, we evaluate the effectiveness of different machine learning (ML) algorithms based on stylistic consistency features in discerning between human-operated and bot-driven Twitter accounts and show that the experimented ML algorithms can achieve high performance (i.e., F-measure values up to 98%) in social bot detection tasks.

1.1.2.2 Face the Truth: Detection of Spontaneous and Posed Emotional Facial Expressions

As facial expressions communicate what we are thinking, intending and feeling, the face is considered the most complex and reliable indicator of emotional states. Indeed, the face includes 43 muscles and is capable of making more than ten thousand combinations of facial expressions [91, 135]. Manifesting unfelt, unauthentic emotions has adaptive value. For this reason, this is an ability individuals manifest since infancy. For instance, an unauthentic cry enables a successful communication with the caregiver, when the infant is still not able to talk [194]. In adulthood, individuals become very skilled in simulate or dissimulate emotional expressions to receive personal and social advantages [75, 90]. To date, some attempts have been made to discriminate spontaneous (i.e., genuine) and posed (i.e., fake) emotions automatically. Unfortunately, the results obtained so far revealed significant variability and inconsistency in state of the art. The great inter-individual variability in the facial displays makes impossible the detection of universal deceptive cues in the emotional expressions.

Contributions: In Chapter 6, we developed a framework for the automatic detection of spontaneous and posed emotional facial expressions from clips. We applied the framework in two scenarios to classify the genuineness of emotional expressions ad hoc for each user (i.e., user-dependent scenario) and investigate the relevancy of inter-individual variability in the emotional lie detection (i.e., user-dependent vs user-independent scenario). Results revealed that Machine Learning models achieved high accuracies in genuineness discrimination (84.4% accuracy on average) when capitalized for a single user specifically. Contrarily, the same approach obtained an average accuracy of 67.0% if deployed on all the users generically. Finally, the implications and applications of the results are discussed in light of the state of the art of lie detection, psychology of emotions, and the AI field.

1.1.2.3 BLUFADE: Blurred Face Detection

Ideally, secure user sessions should start and end with authentication and de-authentication phases, respectively. While the user must pass the former to start a secure session, the latter's importance is often ignored or underestimated. Dangling or unattended sessions expose users to well-known *Lunchtime Attacks*. To mitigate this threat, the research community focused on automated de-authentication systems. Unfortunately, no single approach offers security, privacy, and usability. For instance, although facial recognition-based methods might be a good fit for security and usability, they violate user privacy by constantly recording the user and the surrounding environment.

Contributions: In Chapter 7, we propose `BLUFADE`, a fast, secure, and transparent de-authentication system that takes advantage of blurred faces to preserve user privacy. We obfuscate a webcam with a physical blur layer and use deep learning algorithms to perform face detection continuously. To assess `BLUFADE`'s practicality, we collected two datasets formed by 30 recruited subjects (users) and thousands of physically blurred celebrity photos. The former was used to train and evaluate the de-authentication system performances, the latter to assess the privacy and to increase variance in training data. We show that our approach outperforms state-of-the-art methods in detecting blurred faces, achieving up to 95% accuracy. Furthermore, we demonstrate that `BLUFADE` effectively de-authenticates users up to 100% accuracy in under 3 seconds, while satisfying security, privacy, and usability requirements.

1.2 Publications

An overview of the manuscript produced during my Ph.D.period and published or currently submitted in peer-reviewed journals, conferences, and workshops are listed below. All manuscripts are listed in chronological order of acceptance/submission.

1.2.1 Journal Publication

- Balagani, K., Cardaioli, M., Conti, M., Gasti, P., Georgiev, M., Gurtler, T., ... & Wu, L. (2019). Pilot: Password and pin information leakage from obfuscated typing videos. *Journal of Computer Security*, 27(4), 405-425. **(JCR IF 2018: 1.071)**

- Miolla, A., Cardaioli, M., & Scarpazza C.(2022). Padova Emotional Dataset of Facial Expressions (PEDFE): a unique dataset of genuine and posed emotional facial expressions. *Journal of Behavior Research Methods.* **(JCR IF 2020: 6.242)** Submitted

- Cardaioli, M., Miolla, A., Conti, M., Sartori, G., Monaro, M., Navarin, N., & Scarpazza, C. (2022). Inter-individual variability in the detection of spontaneous and posed emotional facial expressions. *PloS one.* **(JCR IF 2020: 3.24)** Submitted

1.2.2 Conference and Workshop Publications

- Cardaioli, M., Monaro, M., Sartori, G., & Conti, M. (2020, July). Detecting Identity Deception in Online Context: A Practical Approach Based on Keystroke Dynamics. *In International Conference on Applied Human Factors and Ergonomics* (pp. 41-48). Springer, Cham.

- Cardaioli, M., Cecconello, S., Conti, M., Pajola, L., & Turrin, F. (2020, September). Fake News Spreaders Profiling Through Behavioural Analysis. *In CLEF (Working Notes).*

- Cardaioli, M., Conti, M., Balagani, K., & Gasti, P. (2020, September). Your pin sounds good! augmentation of pin guessing strategies via audio leakage. *In European Symposium on Research in Computer Security ESORICS (pp. 720-735). Springer, Cham.* **(CORE: A, LiveSHINE: A+, MA: A; Acceptance rate: 19.67%)**

- Cardaioli, M., Kaliyar, P., Capuozzo, P., Conti, M., Sartori, G., & Monaro, M. (2020, December). Predicting Twitter users' political orientation: an application to the italian political scenario. *In 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (pp. 159-165). IEEE. (**LiveSHINE:B, MA:B; Acceptance rate: 17.8%)**

- Cardaioli, M., Conti, M., Di Sorbo, A., Fabrizio, E., Laudanna, S., & Visaggio, C. A. (2021, July). It'sa Matter of Style: Detecting Social Bots through Writing Style Consistency. *In 2021 International Conference on Computer Communications and Networks (ICCCN) (pp. 1-9). IEEE.* (**CORE: B; LiveSHINE:A-, MA:B; Acceptance rate: 25%)**

- Cardaioli, M., Cecconello, S., Monaro, M., Sartori, G., Conti, M., & Orrù, G. (2021, December). Malingering Scraper: A Novel Framework to Reconstruct Honest Profiles from Malingerer Psychopathological Tests. I*n International Conference on Neural Information Processing (pp. 433-440). Springer, Cham..* **(CORE:B; MA:B)**

- Cardaioli, M., Cecconello, S., Conti, M., Milani, S., Picek, S., & Saraci, E. (2022). Hand Me Your PIN! Inferring ATM PINs of Users Typing with a Covered Hand. *In Proceedings of the 31st USENIX Security Symposium (UNESIX Security 2022), in press, Boston, MA, August 10-12, 2022.* **(CORE: A++; LiveSHINE: A++; MA: A++; Acceptance rate: 19%)**

- Cardaioli, M., Conti, M., Tricomi, P., & Tsudik, G. (2022). Privacy-Friendly De-authentication with BLUFADE: Blurred Face Detection. *In Proceedings of the 20th International Conference on Pervasive Computing and Communications (PerCom 2022), in press, Pisa, IT, March 21-25, 2022.* **(CORE:A++; LiveSHINE:A+; MA:A; Acceptance rate: 15.3%)**

- Cardaioli, M., Miolla, A., Conti, M., Sartori, G., Monaro, M., Navarin, N., & Scarpazza, C. (2022). Face the Truth: *Interpretable Emotion Genuineness. IEEE World Congress on Computational Intelligence (WCCI2022).* Submitted

- Balagani, K., Cardaioli, M., Cecconello, S., Conti, M., & Tsudik, G., (2022). We Can Hear Your PIN Drop. *In European Symposium on Research in Computer Security ESORICS. Springer, Cham.* Submitted

- Cardaioli, M., Conti, M., & Ravindranath, A., (2022). For your Voice Only: Exploiting Side Channels in Voice Messaging for Environment Detection. *In European Symposium on Research in Computer Security ESORICS. Springer, Cham.* Submitted

Part I

In-Security Through Human Interaction Analysis: the PIN Pad Case

Chapter 2

Your PIN Sounds Good: Inter-keystroke Timing Based Attacks on PINs

Authentication via Personal Identification Numbers (PINs) dates back to the mid-sixties [25]. The first devices to use PINs were automatic dispensers and control systems at gas stations, while the first applications in the banking sector appeared in 1967 with cash machines [33]. PINs have found widespread use over the years in devices with numeric keypads rather than full keyboards [255].

In the context of financial services, ISO 9564-1 [133] specifies basic security principles for PINs and PIN entry devices (e.g., PIN pads). For instance, to mitigate shoulder surfing attacks [158, 160, 218], ISO 9564-1 indicates that PIN digits must not be displayed on a screen, or identified using different sounds or sound duration for each key.

As a compromise between security and usability, PIN entry systems display a fixed symbol (e.g., a dot) to represent a key being pressed, and provide the same audio feedback (i.e., same tone, same duration) for all keys. While previous work has demonstrated that observing the dots as they appear on screen as a result of a key press reduces the search space for a PIN [16], to our knowledge no work has targeted the use of audio feedback to recover PINs.

In this chapter, we evaluate how the adversary can reduce PIN search space using audio feedback, with (and without) using observable information such as PIN typing behavior (one- or two-handed), knowledge of one digit of the PIN, and knowledge of which keys have been pressed. We compare our attacks with an attack based on the knowledge of PIN distribution.

Exploiting audio feedback has several advantages compared to observing the user or the screen during PIN entry. First, sound is typically easier to collect. The adversary might not be able to observe the ATM's screen directly, and might risk being exposed when video-recording an ATM in a public space. In contrast, it is easy to record audio *covertly*, e.g., by casually holding a smartphone while pretending to stand in a line behind other ATM users. The sound emitted by ATMs is quite distinctive and can be easily isolated even in noisy environments. Second, sound enables higher time resolution compared to video. Conventional video cameras and smartphones record video between 24 and 120 frames per second. In contrast, audio can be recorded with a sampling rate between 44.1 kHz and 192 kHz, thus potentially allowing at least two orders of magnitude higher resolution.

Contributions. In this chapter, we analyze several novel side channels associated with PIN entry. In particular:

- We show that it is possible to retrieve accurate inter-keystroke timing information from audio feedback. In our experiments, we were able to correctly detect 98% of the keystroke feedback sounds with an average error of 1.8ms. Furthermore, 75% of inter-keystroke timings extracted by the software had absolute error under 15 ms. Our experiments also demonstrate that inter-keystroke timings extracted from audio can be more accurate than the same extracted from video recordings of PIN entry as done in [15, 16].
- We analyze how the behavior of the user affects the adversary's ability to guess PINs. Our results show that users who type PINs with one finger are more vulnerable to PIN guessing from inter-keystroke timings compared to users that enter their PIN using at least two fingers. In particular, the combining inter-keystroke timing with the knowledge that the user is a single-finger typist leads to 34-fold improvement over random guessing when the adversary is allowed to perform up to 5 guessing attempts.
- We combine inter-keystroke timing information with knowledge of one key in the PIN (i.e., the adversary was able to see either the first or the last key pressed by the user), and with knowledge of *which* keys have been pressed by the user. The latter information is available, as shown in this chapter as well as in recent work [2, 144, 193, 271] when the adversary is able to capture a thermal image of the PIN pad after the user has typed her PIN. Our experiments show that inter-

keystroke timing significantly improves performance for both attacks. For example, by combining inter-keystroke timing with a thermal attack, we were able to guess 15% of the PINs at the first attempt, reaching a four-fold improvement in performance. By combining multiple attacks, we were also able to drastically reduce the number of attempts required to guess a PIN. Specifically, we were able to guess 72% of the PINs within the first 3 attempts.

- Finally, we show that uniform PIN selection might not be the best strategy against an adversary with access to one or more of the side-channel information discussed in this chapter.

2.1 Related Work

Non-acoustic Side-channels. Vuagnoux and Pasini [252] demonstrated that it is possible to recover keystrokes by analyzing electromagnetic emanations from electronic components in wired and wireless keyboards. Marquardt et al. [178] showed that it is possible to recover key presses by recoding vibrations generated by a keyboard using an accelerometer. Other attacks focus on keystroke inference via motion detection from embedded sensors on wearable devices. For example, Sarkisyan et al. [225] and Wang et al. [254] infer smartphone PINs using movement data recorded by a smartwatch.

Those attacks require that the adversary is able to monitor the user's activity while the user is typing. However, there are attacks that allow the adversary to exploit information available several seconds after the user has typed her password. For instance, one such attack is based the observation that when a user presses a key, the heat from her finger is transferred to the keypad, and can be later be measured using a thermal camera [271]. Depending on the material of the keyboard, thermal residues have different dissipation rates [193], thus affecting the time window in which the attacks are effective. Abdelrahman et al. [2] evaluated how different PINs and unlock patterns on smartphones on can influence thermal attack performance. Kaczmarek et al. [144] demonstrated how a thermal attack can recover precise information about a password up to 30 seconds after it was typed, and partial information within 60 seconds.

Acoustic Side-channels. Asonov and Agrawal showed that each key on a keyboard emits a characteristic sound, and that this sound can be used to infer individual keys [10]. Subsequent work further demonstrated the effectiveness of sound emanation for text reconstruction. Berger et al. [26] combined keyboard acoustic emanation with a dictionary attack to reconstruct words, while Halevi and Saxena [116] analyzed keyboard acoustic emanations to eavesdrop over random password. Because ISO 9564-1 [133] specifications require that each key emits the same sound, those attacks do not apply to common keypads, including those on ATMs.

Another type of acoustic attack is based on time difference of arrivals (TDoA) [166, 256, 277]. These attacks rely on multiple microphones to triangulate the position of the keys pressed. Although this attacks typically result in good accuracies, they are difficult to instantiate in realistic environments.

Song et al. [239] presented an attack based on latency between key presses measured by snooping encrypted SSH traffic. Their experiments show that information about inter-keystroke timing can be used to narrow the password search space substantially. A similar approach was used by Balagani et al. [16], who reconstructed inter-keystroke timing from the time of appearance of the masking symbols (e.g., "dots") while a user types her password. Similarly, Balagani et al. [15] demonstrated that precise inter-keystroke timing information recovered from videos drastically reduces the number of attempts required to guess a PIN. The main limitation of [15, 16] is that they require the adversary to video-record the ATM screen while the user is typing her PIN. Depending on the location and the ATM, this might not be feasible. Further, this reduces the set of vulnerable ATMs and payment systems to those that display on-screen feedback.

To our knowledge, this is the first work to combine inter-keystroke timing information deduced from sound recording with observable information from other sources, and thereby drastically reduce the attempts to guess a PIN compared to prior work. Our attacks are applicable to a multitude of realistic scenarios. This poses an immediate and severe threat to current ATMs or PoS.

2.2 Adversary Model

In this section we evaluate four classes of information that the adversary can exploit to infer PINs. These classes are: (1) Key-stroke timing information extracted from audio recordings; (2) Knowledge of whether the user is a single- or multi-finger typist; (3) Information about the first or the last digit of the PIN; and (4) Information about which keys have been pressed, but not their order. Next, we briefly review how each of these classes of information can be collected by the adversary.

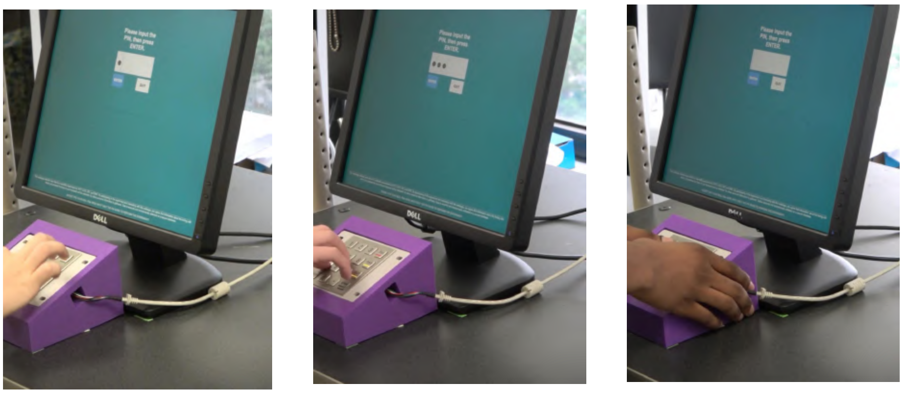


Figure 2.1: Different typing strategies. Left: one finger; center: multiple fingers of one hand; right: multiple fingers of two hands.

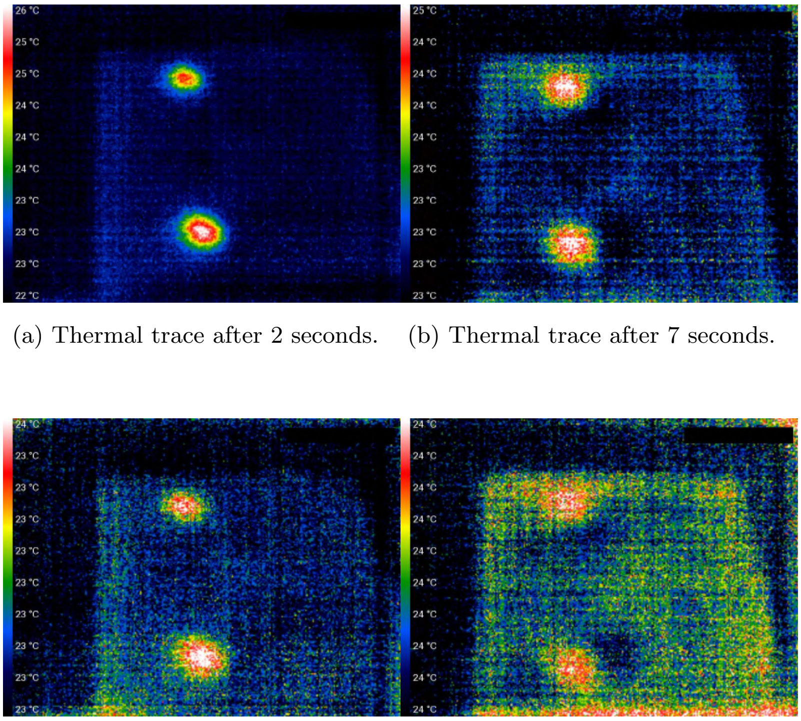
Class 1: Keystroke Timing. Keystroke timing measures the distance between consecutive keystroke events (e.g., the time between two key presses, or between the release of the key and the subsequent keypress). Collecting keystroke timing by compromising the software of an ATM located in a public space, or physically tampering with the ATM (e.g., by modifying the ATM's keyboard) is not practical in most cases. However, as shown in [15], the adversary can infer keystroke timings without tampering with the ATM by using video recordings of the "dots" that appear on the screens when the user types her PIN. In this chapter, we leverage audio signals to infer precise inter-keystroke timings.

Class 2: Single- or Multi-finger Typists. The adversary can typically directly observe whether the user is typing with one or more fingers. While the number of fingers used to enter a PIN does not reveal information about the PIN itself, it might be a useful constraint when evaluating other sources of information leakage. Figure 2.1 shows users typing using a different number of fingers.

Class 3: Information about the first or the last digit of the PIN. As users move their hands while typing their PIN, the adversary might briefly have visibility of the keypad, and might be able to see one of the keys as it is pressed (see Figure 2.1). We model this information by disclosing either the first or the last digit of the PIN to the adversary.

Class 4: Which Keys Have Been Pressed. This information can be collected using various techniques. For instance, the adversary can use a thermal camera to determine which keys are warmer, thus learning which digits compose the PIN (see, e.g., Figure). As an alternative, the adversary can place UV-sensitive powder on the keys before the user enters her PIN, and then check which keys had the powder removed by the users using a UV light.

While these attacks do not reveal the order in which the keys were pressed (except when the PIN is composed of one repeated digit), they significantly restrict the search space. Although this attack can be typically performed covertly, it requires specialized equipment.



(a) Thermal trace after 2 seconds. (b) Thermal trace after 7 seconds.

(c) Thermal trace after 10 seconds. (d) Thermal trace after 15 seconds.

Figure 2.2: Thermal image of a metallic PIN pad after applying a transparent plastic cover for PIN 2200.

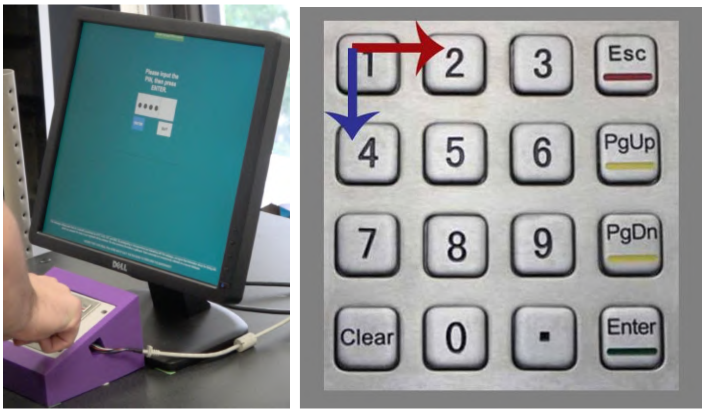


Figure 2.3: Left: user typing a PIN using the ATM simulator. Right: close up view of the ATM simulator's keypad.

2.3 Experiment Results

We extracted keystroke sounds using the dataset from [16]. This dataset was collected from 22 subjects, who typed several 4-digit PINS on a simulated ATM (see Figure 2.3). Nineteen subjects completed three data collection sessions, while three subjects completed only one session.

In each session, subjects entered a total of 180 PINs as follows: each subject was shown a 4-digit PIN. The PIN remained on the screen for 10 seconds, during which the subject was encouraged to type the PIN multiple times. After 10 seconds, the PIN disappeared from the screen. At this point, the subject was asked to type the PIN 4 times from memory. In case of incorrect entry, the PIN was briefly displayed again on the screen, and the subject was allowed to re-enter it. This procedure was repeated in three batches of 15 PINs. As a result, each PIN was typed 12 times per session.

Each time a subject pressed a key, the ATM simulator emitted an audio feedback and logged the corresponding timestamp with millisecond resolution. Users were asked to type 44 different 4-digit PINs which represented all the Euclidean distances between keys on the keypad. Sessions were recorded in a relatively noisy indoor public space (SNR -15 dB) using a Sony FDR-AX53 camera located approximately 1.5 m away from the PIN pad. The audio signal was recorded with a sampling frequency of 48 kHz.

2.3.1 Extraction of Keystroke Timings from Keypad Sound

To evaluate the accuracy of timing extraction from keystroke sounds, we first linearly normalized the audio recording amplitude in the interval $[-1, 1]$. We applied a 16-order Butterworth band-pass filter [39] centered at 5.6 kHz to isolate the characteristic frequency window of the keypad feedback sound. Finally, to isolate the signal from room noise, we processed the audio recording to "mute" all samples with an amplitude below a set threshold (0.01 in our experiments).

We then calculated the maximum amplitude across nearby values in a sliding window of 1,200 samples (consecutive windows had 1199 overlapping samples), corresponding to 25 milliseconds of audio recording. We determined the length of the window by evaluating the distance between consecutive timestamps logged by the ATM simulator (ground truth), which was at least 100 ms for 99.9% of the keypairs. Figure 2.4 shows the result of this process.

We then extracted the timestamps of the peaks of the processed signal and compared them to the ground truth. Our results show that this algorithm can accurately estimate inter-keystroke timing information. We were able to correctly detect 98% of feedback sound with a mean error of 1.8 ms.

Extracting timings from audio led to more accurate time estimation than using video [16]. With the latter, 75% of the extracted keystroke timings had errors of up to 37 ms. In contrast, using audio we were able to extract 75% of the keystroke with errors below 15 ms. Similarly, using video, 50% of the estimated keystrokes timings had errors of up to 22 ms, compared to less than 7 ms with audio. Figure 2.5 shows the errors distribution for timings extracted from video and audio recordings.

2.3.2 PIN Inference from Keystroke Timing (Class 1)

This attack ranks PINs based on the estimated Euclidean distance between subsequent keys in each PIN. In particular, we calculated an inter-key Euclidean distance vector from a sequence of inter-keystroke timings inferred from audio feedback. As an example, the distance vector associated with PIN `5566` is $[0, 1, 0]$, where the first '0' is the distance between keys `5` and `5`, '1' between keys `5` and `6`, and '0' between `6` and `6`. Any four-digit PIN is associated with one distance vector of size three. Each element of the distance vector can be 0, 1, 2, 3, diagonal distance 1 (e.g., `1-3`), diagonal distance 2 (e.g., `3-7`), short diagonal distance (e.g., `2-9`), or long diagonal distance (e.g., `3-0`). Different PINs might be associated with the same distance vector (e.g., `1234` and `4567`). The goal of this attack is to reduce the search space by considering only PINs that match the estimated distance vector.

For evaluation, we split our keystroke dataset into two sets. The first (training set) consists of 5195 PINs, typed by 11 subjects. The second (test set) consists of 5135 PINs, typed by a separate set of 11 subjects. This models the lack of knowledge of the adversary of the specific typing patterns of the victim user.

To estimate the Euclidean distances between consequent keys, we modeled a set of gamma function on the inter-keystroke timing distribution, one for each distance. We then applied the algorithm from [15] to infer PINs from estimated distances. With this strategy, we were able to guess 4% of PINs within 20 attempts—a 20-fold improvement compared to random guessing.

Figure 2.6 shows how timings extracted from audio and video feedback affect the number of PIN guessed by the algorithm compared to ground truth. Timings extracted from audio feedback exhibit a smaller decrease in guessing performance compared to timings extracted from video.

2.3.3 PIN Inference from Keystroke Timing and Typing Behavior (Class 2)

This attack improves on the keystroke timing attack by leveraging knowledge of whether the user is a single- or multi-finger typist. This additional information allows the adversary to better contextualize the timings between consecutive keys. For single-finger typists, the Euclidean distance between keys `1` and `0` is the largest (see Fig 2.3), and therefore we expect the

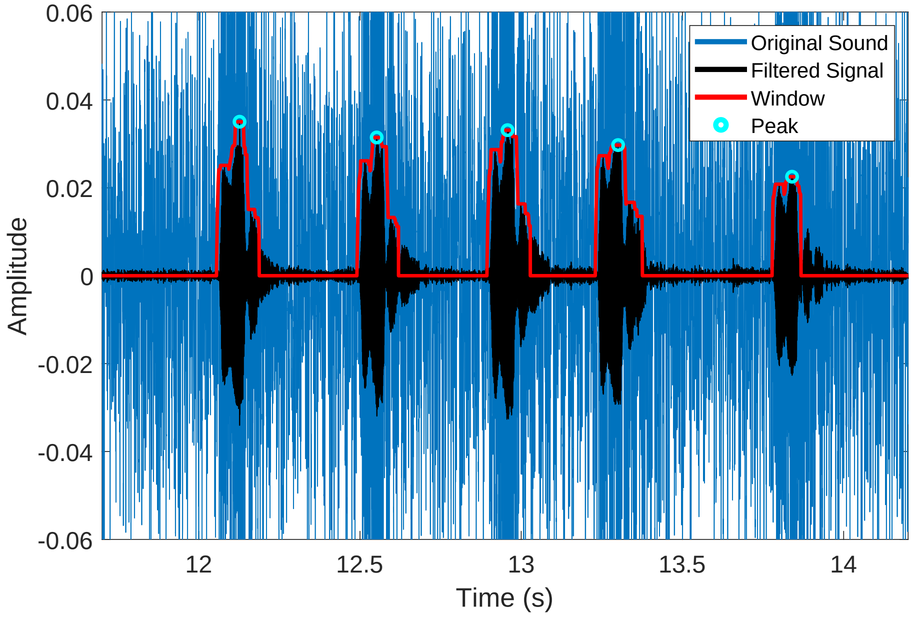
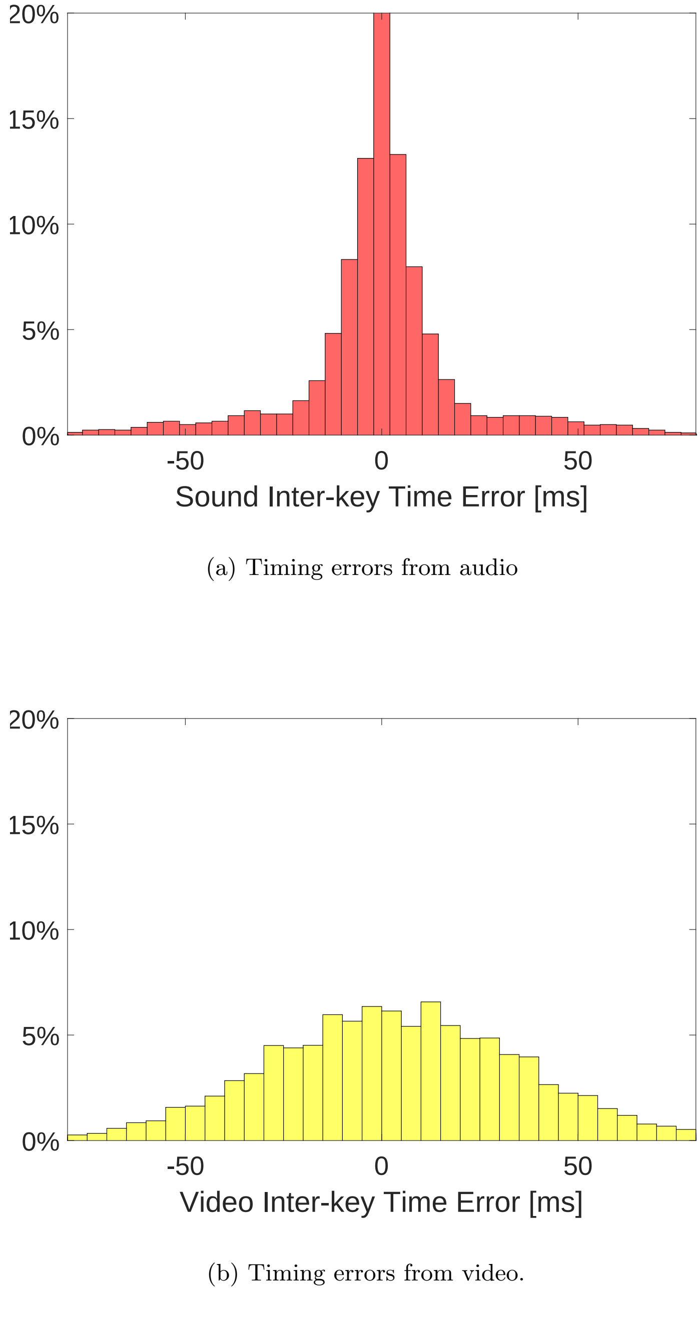


Figure 2.4: Comparison between the original sound signal, filtered sound signal, windowed signal, and extracted peaks.



(a) Timing errors from audio

(b) Timing errors from video.

Figure 2.5: Error distribution of estimated inter-keystroke timings.

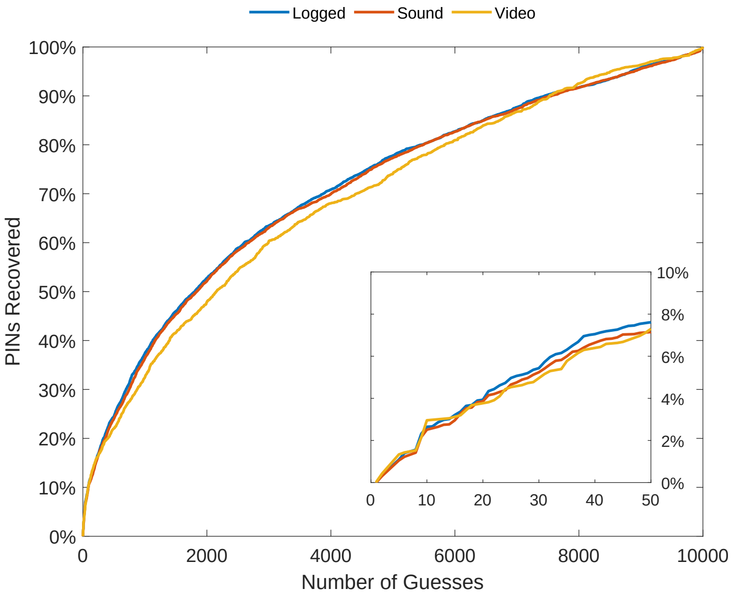


Figure 2.6: CDF showing the percentage of PINs recovered using keystroke timing information derived from the ground truth (logged), sound feedback, and video.

corresponding inter-keystroke timing to be the largest. However, if the user is a two-finger typist, then 1 might be typed with the right hand index finger, and 0 with the left hand index finger. As a result, the inter-keystroke time might not be representative of the Euclidean distance between the two keys.

To systematically study typing behavior, we analyzed 61 videos from the 22 subjects. 70% of the subject were single-finger typists; 92% of them entered PINs using the index finger, and 8% with the thumb. We divided multi-finger typists into three subclasses: (1) PINs entered using fingers from two hands (38% of the PINs typed with more than one finger); (2) PINs entered with at least two fingers of the same hand (34% of the PINs typed with more than one finger); and (3) PINs that we were not able to classify with certainty due obfuscation of the PIN pad in the video recording (28% of the PINs typed with more than one finger).

In our experiments, subjects' typing behavior was quite consistent across PINs and sessions. Users that were predominantly single-finger typists entered 11% of their PINs using more than one finger, while multi-finger typists entered 23% of the PINs using one finger.

We evaluated guessing performance of timing information inferred from audio feedback on single-finger PINs and multi-finger PINs separately. We were able to guess a substantially higher number of PINs for each number of attempts for users single-finger typists (see Figure 2.7) compared to multi-finger typists. In particular, the percentage of PINs recovered within 5 attempts was twice as high for PINs entered with one finger compared to PINs entered with multiple fingers. Further, the guessing rate within the first 5 attempts was 34 times higher compared to random guessing when using timing information on single-finger PINs. However, our ability to guess multi-finger PINs using timing information was only slightly better than random. This strongly suggests that the correlation between inter-keystroke timing and Euclidean distance identified in [16] holds only quite strongly for PINs entered using a single finger, and only marginally for PINs entered with two or more fingers.

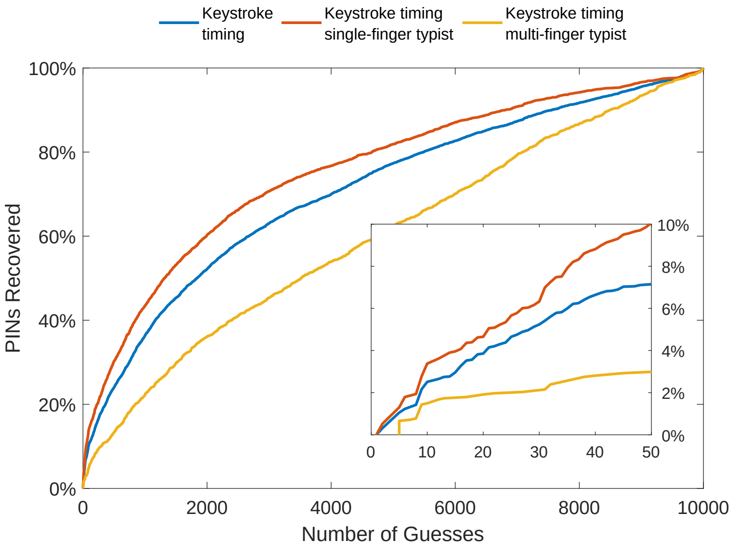


Figure 2.7: CDF showing the percentage of PINs recovered using only keystroke timing information from audio feedback, compared to timing information for single- or multi-finger typists.

2.3.4 Knowledge of the First or the Last Digit of the PIN (Class 3)

In this section, we examine how information on the first or last digit of the PIN reduces the search space when combined with keystroke timings.

Knowledge of one digit alone reduces the search space by a factor of 10 regardless of the digit's position, because the adversary needs to guess only the remaining three digits. (As a result, the expected number of attempts to guess a random PIN provided no additional information is 500.)

To determine how knowledge of the first or the last digit impacts PIN guessing based on keystroke timing, we applied the same procedure described in Section 2.3.2: for each PIN in the testing set, we associated a list of triplets of distances sorted by probability. We then pruned the set of PINs associated with those distance triplets to match the knowledge of the first or last PIN. For instance, given only the estimated distances 3, 0, and $\sqrt{2}$, the associated PINs are 0007, 0009, 2224, and 2226. If we know that the first digit of the correct PIN is 2, then our guesses are reduced to 2224 and 2226.

Information about the first or last digit of the PIN boosted the guessing performance of the keystroke-timing attack substantially, as shown in Figure 2.8. In particular, guessing accuracy increased by 15-19 times within 3 attempts (4.36% guessing rate when the first digit was known, and 5.57% when the last digit was known), 7 times within 5 attempts, and about 4 times within 10 attempts, compared to timing information alone. In all three cases, timing information substantially outperformed knowledge of one of the digits in terms of guessing rate.

2.3.5 Knowledge of Which Keys Have Been Pressed (Class 4)

In this section, we evaluate how knowledge of *which digits* compose a PIN, but not *their order*, restricts the PIN search space, in conjunction with information about keystroke timings. The adversary can acquire this knowledge, for instance, by observing the keypad using a thermal camera shortly after the user has typed her PIN [144], or by placing UV-sensitive powder on the keys before the user enters her PIN, and then checking which keys were touched using a UV light.

Information on which digits compose a PIN can be divided as follows:

1. The user pressed only one key. In this case, the user must have entered the same digit 4 times. No additional information is required to recover the PIN.

2. The user pressed two distinct keys, and therefore each digit of the PIN might be repeated between one and three times, and might be in any position of the PIN. In this case, the number of possible PINs is $2^4 - 2 = 14$, i.e., the number of combinations of two values in four

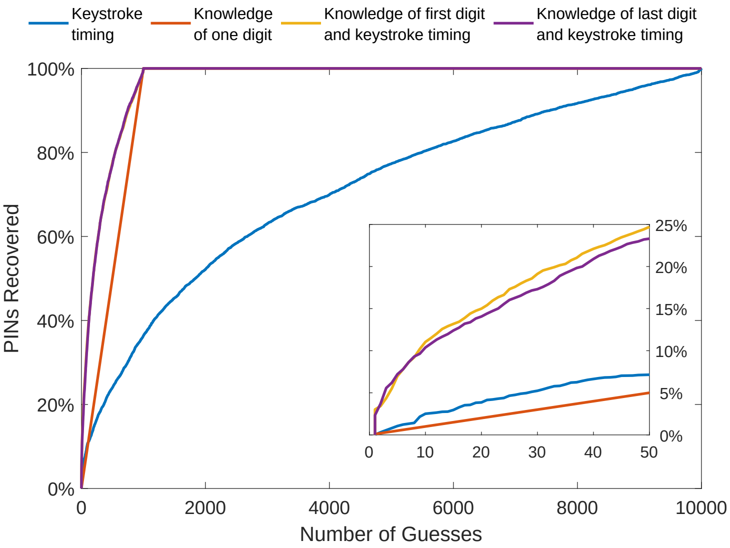


Figure 2.8: CDF showing the percentage of PINs recovered using keystroke timings inferred from audio, random guessing over 3 digits of the PIN, and using inferred keystroke timings and knowledge of the first or the last digit of the PIN.

position, except for the combinations where only one of the two digits appears.

3. The user pressed three distinct keys. The number of possible PINs is equal to the combinations of three digits in four positions, i.e., $4 \cdot 3 \cdot 3 = 36$

4. The user pressed four distinct keys. The number of possible PINs is $4! = 24$.

We evaluated how many PINs the adversary could recover given keystroke timings and the set of keys pressed by the user while entering the PIN. Our results, presented in Figure 2.9, show that combining these two sources of information leads to a high PIN recovery rate. Specifically, within the first three attempts, knowing only which keys were pressed led to the recovery of about 11% of the PINs. Adding timing information increased this value to over 33%.

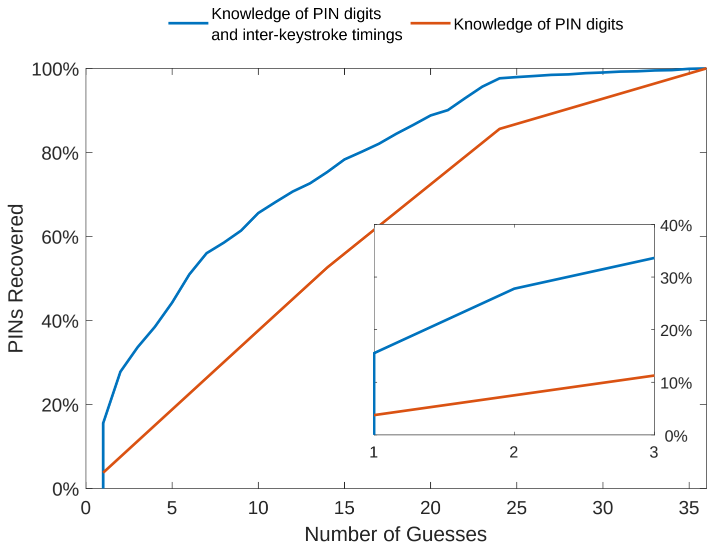


Figure 2.9: CDF showing the percentage of PINs recovered with the knowledge of which keys have been pressed with and without inter-keystroke timing information.

2.3.6 Combining Multiple Classes of Information

In this section we examine how combining multiple classes of information leads to an improvement in the probability of correctly guessing a PIN.

First, we investigated how guessing probability increases when the adversary knows one of the digits of the PIN (first or last), the typing behavior (single-finger typist), and is able to infer inter-keystroke timing information from audio feedback. We used 3461 PINs typed by 11 subjects containing only single-finger PINs. In our experiments, we were able to guess 8.73% of the PINs within 5 attempts, compared to 6.97% with timing information and knowledge of one digit.

We then considered knowledge of the values composing the PIN, typing behavior, and inferred timing information. In this case, we successfully guessed 50.74% of the PINs within 5 attempts, and 71.39% within 10 attempts.

Finally, when we considered the values composing the PIN, one of the PIN's digits, and inferred timing information, we were able to guess 86.76% of the PINs in 5 attempts, and effectively all of them (98.99%) within 10 attempts. All our results are summarized in Table 2.1.

Table 2.1: Results from all combinations of attacks considered in this chapter, sorting by guessing rate after 5 attempts. Because single finger reduces the PIN search space only in conjunction with inter-keystroke timings, we do not present results for single finger alone.

Information				PINs Guessed Within Attempt				
Keystroke Timing	Single Finger	First Digit	PIN Digits	1	2	3	5	10
				0.01%	0.02%	0.03%	0.05%	0.10%
		o		0.10%	0.20%	0.30%	0.50%	1.00%
o				0.02%	0.31%	0.70%	1.05%	2.51%
o	o			0.03%	0.52%	0.91%	1.30%	3.38%
o		o		3.02%	3.72%	4.36%	6.97%	11.04%
o	o	o		3.73%	4.13%	5.43%	8.73%	14.01%
			o	3.76%	7.52%	11.28%	18.80%	37.60%
o			o	15.54%	27.79%	33.63%	44.25%	65.57%
o	o		o	19.04%	34.01%	40.60%	50.74%	71.31%
		o	o	13.27%	26.62%	39.88%	66.40%	92.80%
o		o	o	35.27%	53.46%	66.84%	86.76%	98.99%
o	o	o	o	40.86%	60.24%	71.77%	89.19%	99.28%

2.4 PINs and Their Guessing Probability Distribution

In this section, we evaluate whether the classes of information identified in this work make some of the PINs easier to guess than others, and thus intrinsically less secure. With respect to estimated inter-keystroke timings, different timing vectors identify a different number of PINs. For instance, vector [0,0,0] corresponds to 10 distinct PINs (`0000`, `1111`, ...), while vector [1,1,1] corresponds to 216 PINs (`0258`, `4569`, ...). This indicates that, against adversaries who are able to infer inter-keystroke timing information, choosing PINs uniformly at random from the entire PIN space is not the best strategy.

The adversary's knowledge of which digits compose the PIN has a similar effect of the guessing probability of individual PINs. In this case, PINs composed of three different digits are the hardest to guess, with a probability of 1/36, compared to PINs composed of a single digit, which can always be guessed at the first attempt.

The adversary's knowledge of one digit of the PIN and/or the typing behavior do not affect the guessing probability of individual PINs.

2.5 Summary

In this chapter, we showed that inter-keystroke timing inferred from audio feedback emitted by a PIN pad compliant with ISO 9564-1 [133] can be effectively used to reduce the attempts to guess a PIN. Compared to prior sources of keystroke timing information, audio feedback is easier to collect, and leads to more accurate timing estimates (in our experiments, the average reconstruction error was 1.8 ms). Due to this increase in accuracy, we were able to reduce the number of attempts needed to guess a PIN compared to timing information extracted from videos.

We then analyzed how using inter-keystroke timing increases guessing performance of other sources of information readily available to the adversary. When the adversary was able to observe the first or the last digit of a PIN, inter-keystroke timings further increased the number of PINs guessed within 5 attempts by 14 times. If the adversaries was capable of observing which keys were pressed to enter a PIN (e.g., using a thermal camera), adding inter-keystroke timing information allowed the adversary to guess 15% of the PINs with a single attempt. This corresponds to a 4 times reduction in the number of attempts compared to knowing only which keys were pressed.

We evaluated how typing behavior affects guessing probabilities. Our results show that there is a strong correlation between Euclidean distance between keys and inter-keystroke timings when the user enters her PIN using one finger. However, this correlation was substantially weaker when users typed with more than one finger.

We then showed that the combination of multiple attacks can dramatically reduce attempts to guess the PIN. In particular, we were able to guess 72% of the PINs within the first 3 attempts, and about 90% of the PINs within 5 attempts, by combining all the sources of information considered in this chapter.

Finally, we observed that different adversaries require different PIN selection strategies. While normally PINs should be selected uniformly at random from the entire PIN space, this is not true when the adversary has access to inter-keystroke timings or thermal images. In this case, some classes of PINs (e.g., those composed of a single digit) are substantially easier to guess than other classes (e.g., those composed of three different digits). As a result, uniform selection from appropriate *subsets* of the entire PIN space leads to harder-to-guess PINs against those adversaries.

We believe that our results highlight a real threat to PIN authentication systems. The feasibility of these attacks and their immediate applicability in real scenarios poses a considerable security threat for ATMs, PoS-s, and similar devices.

Chapter 3

Hand me your PIN: Inferring PINs from Videos of Users Typing with a Covered Hand

The wide deployment of various Cyber-Physical Systems (CPS) has a significant impact on our daily lives. Unfortunately, the increased use of CPS also brings more threats to users. This is especially pronounced considering new attack vectors that use machine learning approaches. As such, threats become a global issue, and the need to design secure and robust systems increases. One common security mechanism in devices like Automated Teller Machines (ATMs) and Point of Sale (PoS) depends on the security provided by the Personal Identification Numbers (PINs). While ATMs and PoS devices are widely used [1], many people do not consider security risks and defenses beyond those commonly mentioned [2]: i) hide the PIN while typing, and i) make sure no one watches the screen (shoulder-surfing attack). In the context of financial services, ISO 9564-1 [133] specifies the basic security principles for PINs and PIN entry devices (e.g., PIN pads). For example, to mitigate the shoulder surfing attacks [30, 82], the standard indicates that i) PIN digits must not be displayed on a screen, and ii) the duration and type of feedback sound emitted must be the same for each key. Consequently, as a compromise between security and usability, PIN entry systems display a fixed symbol (e.g., a dot) to represent a digit being pressed and provide the same audio feedback (i.e., same tone, same duration) for all keys. Thus, the combination

[1] https://sdw.ecb.europa.eu/reports.do?node=1000001407

[2] https://www.hsbc.com.hk/help/cybersecurity-and-fraud/atm-scams/

of security mechanisms enforced by standards and the common precaution measures taken by users should provide sufficient protection. Unfortunately, the attackers also improve their approaches over time and consider more sophisticated attacks.

The security of ATM and PoS devices is of great concern as millions of such devices are used [73]. Resourceful attackers that succeed in attacking even a small percentage of those devices can cause significant damage considering costs and public perception. This problem is especially pronounced as last years brought significant developments in the attack techniques [15, 42, 167]. At the same time, attacking ATM or PoS devices is not easy, especially if considering realistic settings. Most of the state-of-the-art attacks can be defeated by a careful user covering the PIN that is entered. Recent results that consider thermal cameras are also difficult to succeed, depending on the keypad type and the time users spend operating the device. The attacker can also use timing or acoustic attacks to infer information about the entered digits, but they are not as effective as the state-of-the-art attacks since they require additional information such as thermal residues [42], making it challenging to apply realistically such attacks.

This work proposes a novel attack aiming to reconstruct PINs entered by victims that cover the typing hand by the other hand. More precisely, we leverage the advances in the deep learning domain to develop an attack predicting what PIN is entered based on the position of the user's hand and the movements while pressing the keys. Our attack gives high accuracy rates even in the cases when the user perfectly covers the typing hand. What is more, our attack reaches higher accuracy values than previous works that needed to consider several sources of the information at the same time (timing, sound, and thermal signatures) [42].

Our attack considers a profiling setting where the attacker has access to a PIN pad that is identical (or at least similar) to the one used by the victim. Then, we build a profiling model that can predict what digit is entered on the target device. This is the first attack on PIN mechanisms that works even when the PIN is covered while being entered to the best of our knowledge. Our attack demonstrates that the ATM and PoS security mechanisms are insufficient, and we must provide novel defenses to mitigate attackers. We made our code and datasets publicly available at https://spritz.math.unipd.it/projects/HandMeYourPIN.

Contributions. The main contributions of this chapter are:

- We propose a novel attack to infer PINs from videos of users covering the typing hand with their non-typing hand.

- We demonstrate that our attack can reconstruct 30% of 5-digit PINs and 41% of 4-digit PINs within three attempts, showing that hiding the PIN while typing is insufficient to ensure proper protection.
- We evaluate our attack via extensive experiments, collecting videos of 5 800 5-digit PINs entered in a simulated ATM by 58 participants. We conduct a study to assess humans' accuracy in inferring covered PINs from videos. We show that our attack outperforms humans, achieving a four-fold improvement on reconstructing 5-digits PINs within three attempts.
- We pre-process our dataset, and we make it publicly available to the research community. We hope this is beneficial to understand the problem better and propose possible solutions.
- We discuss several countermeasures that would make the attack more difficult to conduct. We perform an analysis on the attack performance when covering the PIN pad (coverage 25%, 50%, 75%, and 100%) and show that attacks are possible even when using this countermeasure.

3.1 Related Work

Side-channel attacks specifically target the information gained by the implementation of a system [176]. Most of the time, these attacks exploit channels like sound [105], timing [153], power consumption [150], and electromagnetic emanations [29] to learn the system's secrets in use. In [153], the authors managed to crack RSA keys by carefully timing the operations performed by the key-generating algorithm. Another example of a timing attack is reported in [239], where the authors measured the timing between keystrokes in interactive SSH sessions in an attempt to retrieve the typed passwords.

Human behavior can also be defined as a side-channel of a system, especially if the analyzed behavior directly results from the system's requirements. In [18], the authors analyzed the hand movements of people typing on a keyboard and, by using basic computer vision techniques, they tried to reconstruct the text being typed. In [234], the authors again analyzed the finger motion during the PIN-entry process on smartphones. They showed that 50% of the 4-digit PINs could be retrieved in just one attempt. Different from our work, where the target of the attack is a physical PIN pad, in [234], the attackers could also exploit more information. In particular, the users typed the PIN using only one finger, and the attacker knew the finger the users are typing. The different contexts and assumptions make the works substantially different. In [242], the authors presented a side-channel attack on tablets, consisting of analyzing the backside movements of the tablet itself to infer what is being typed by the victim. To do so, they selected some peculiar features of the backside of the tablets (e.g., logos, side-buttons) and analyzed their movement throughout the frames to understand what area of the virtual keyboard is being pressed. Similarly, in [267], the authors presented an attack to infer the pattern lock of mobile devices from videos. Different from our approach, in [267], the attacker required a vision of the user's fingertip while drawing the pattern and a part of the device.

PIN and PIN pad attacks represent a branch of side-channel attacks that exploit information leakage from keyboards and numeric keyboards (i.e., PIN pads) to infer what the victim has typed (e.g., passwords or PINs). In this context, some works focused on exploiting the heat transferred from the hand to the keypad when the victim enters the PIN or password [144, 193]. The attacker points a thermal camera to the keypad as soon as the victim has finished entering the PIN. The thermal image shows which keys have been pressed and even highlights the order in which the victim pressed them. The main advantage of this attack is that it does not require the attacker to do anything while the victim is typing the PIN. On the other hand, the attacker must act quickly (i.e., within seconds) for a higher success rate as the heat on the keypad rapidly fades away. Another drawback of the attack is that its effectiveness depends on the keypad's material (e.g., metal PIN pads completely nullify the attack because of their high thermal conductivity).

Timing attacks against PINs represent another type of side-channel attack against this authentication method. In the scenario presented in [15], the attacker recorded the screen of an ATM while the victim is entering the PIN. When analyzing the recorded video, the attacker exploited the PIN masking symbols appearing on the ATM screen to extract timing information about the keystrokes. The attacker used predictive models to infer which keys were most likely typed by the victim, starting from the deduced inter-keystroke timing. In [42], the authors used the ATM's sound whenever a button is pressed. ATM's sound must be independent of which button is being pressed (i.e., a generic feedback sound). This consideration means that one feedback sound will not help the attacker. However, the sound gives enough information to extract a timestamp of the keys being pressed. Moreover, in [42, 167], the authors showed how combining timing, acoustic, and thermal information can significantly reduce the number of attempts to guess a PIN (e.g., 34% of 4-digits PINs are recovered in three attempts). These attacks need to be reevaluated from a feasibility perspective in a real-world setting. In particular, as shown in [144], the heat signature is dissipated abruptly by metal PIN pads. The lack of this information limits the performance of the attacks presented in [42, 167], reducing the probability of guessing a 4-digit PIN in 3 attempts to 5%.

Our work shows several advantages over the state-of-the-art in ATM PIN inference. To the best of our knowledge, we are the first to investigate the security of hand covering protection methods for ATM's PIN entering. Further, our method shows a significant improvement in reconstructing the PIN compared to previous work on metal PIN pads, reaching 41% of success in reconstructing 4-digit PINs in three attempts (and correctly guessing every third PIN in the first guess).

3.2 Threat Model

The attack is performed when a victim interacts with a generic ATM keypad and types the PIN. The ATM is equipped with a PIN pad that emits a feedback sound when a key is pressed. The feedback sound is the same for all the keys of the PIN pad. The ATM is equipped with a monitor where obfuscated symbols appear when users enter a PIN to mask the entered digits. We do not assume that the ATM or its PIN pad have been compromised during the attack. Our approach can be considered an alternative to card-skimmer attacks since we consider a different source of information to retrieve the PIN. Usually, card-skimming attacks rely on fake PIN pads that directly record the entered digits [227], while our approach infers the PINs from a video.

3.2.1 Attacker

The attacker is a malicious user aiming to steal the victim's secret PIN. The attacker can place a hidden camera near the ATM to record the PIN pad. We make no assumptions about the type of camera used by the attacker except that it records in the visible spectrum [3]. We assume that the camera can easily be hidden close to the ATM while keeping a direct view of the PIN pad (i.e., a pinhole camera if the attacker has access to the ATM [4] or any standard camera placed outside the ATM chassis). We also do not assume any specific position for the camera, but we discuss various camera placements' advantages. We primarily consider the scenario where the attacker uses only one camera, but we also discuss the attack performance when using multiple cameras.

[3] We will use cheap and easily-concealable video sensing equipment, where standard RGB cameras fit such requirements.

[4] https://www.sperrywest.com/cameras/

The attack may take place together with different card stealing approaches: i) card skimming both on chip [32] or magnetic stripe [227] (currently, the two payment-enabling technologies work together [56]), ii) exploiting a relay attack on a contactless card [115], and iii) physically stealing the victim's card.

We assume a profiling side-channel attack where the side-channel information comes from the video of the victim's hand while entering the PIN. More precisely, side-channel information is the position of the victim's hand and the hand movements (both moving the hand/fingers to reach different keypads or movements observable due to muscle movements while a certain keypad is pressed). The attacker can record a number of PINs entered on a copy of the ATM device and train a profiling model to predict what key is pressed. The attacker can retrieve the timestamps when the victim has typed the single keys on the keypad and can do so by listening to the audio of the video recording. There are two different types of sound clues that the attacker can exploit: the first one is the feedback sound made by the keypad when a key is pressed [42], the second one is the sound of the physical button of the keypad that is pressed. External noise does not prevent the attacker from extracting the keypresses, as the camera is close enough to the keypad. As such, the sound can still be identified in the audio track. If, for any reason, the attacker has no way to retrieve the timestamps from the recorded audio (or if there is no audio at all), it is possible to place the camera to record both the keypad and the screen of the ATM [15]. This allows the attacker to extract the keypresses' timing by looking at the PIN masking symbols appearing on the screen. Common masking symbols are usually dots and asterisks. The attacker can use any method to build a profiling model to predict what keys are pressed. We consider the top three predictions as a measure of success since most ATMs will allow entering the PIN three times before blocking the card. Finally, we do not assume that the PIN has any specific structure (pattern) that could be used to improve the attack performance further.

3.2.2 Victim

We assume that the victim adopts basic countermeasures against card-skimming attacks, such as covering the hand while entering the PIN. The attacker does not need to be there when the victim types the PIN, as the attacker can freely access the camera's recorded video, either remotely or at a different time.

3.3 Attack Approach

Our attack assumes that the attacker has access to a training device and controls the PIN selection. Additionally, the attacker knows the layout of a target device and will select the training device to be similar. The attacker does not know the specific person to be attacked or the PIN for the attacked device.

3.3.1 Attack Phases

We can divide the attack into three phases: Phase A – Training, Phase B – Video Recording, and Phase C – PIN Inference. Figure 3.1 shows the required steps for the attack.

Phase A – Training

The attacker selects an ATM as the target of the attack. Next, the attacker sets up a replica of the target ATM. This replica does not have to be a faithful copy of the original, as our model takes in as input a crop around the keypad of the ATM. Therefore, the attacker must use a keypad similar to the one on the target ATM. The best situation is when the attacker can retrieve the same PIN pad model. Alternatively, the attacker can also use PIN pads that differ slightly (e.g., the key spacing can vary by a few millimeters). Note that the layout of ATM PIN pads has to follow the ISO 9564 standard [133].

The attacker uses the ATM replica to build the training set, simulating the victim's behavior while entering the PIN (i.e., covering the typing hand). The attacker must enter sequences of PINs on the replica PIN pad, including all ten digits (i.e., all the digits must be included in the training set). Without losing generality, the attacker can use a USB PIN pad that logs the keys pressed and the corresponding timestamps. The attacker uses this information to segment the videos and labels them. Leveraging the logs, the attacker builds a training set containing, for each key pressed, a sequence of frames and the corresponding label (digits). Finally, the attacker trains the predictive model on the collected training set. For a detailed discussion on the implemented model, we refer readers to Section 3.4.5.

Phase B – Video Recording

The attacker hides a camera near the target ATM to record the PIN pad. There are multiple places where the camera can be placed, and depending

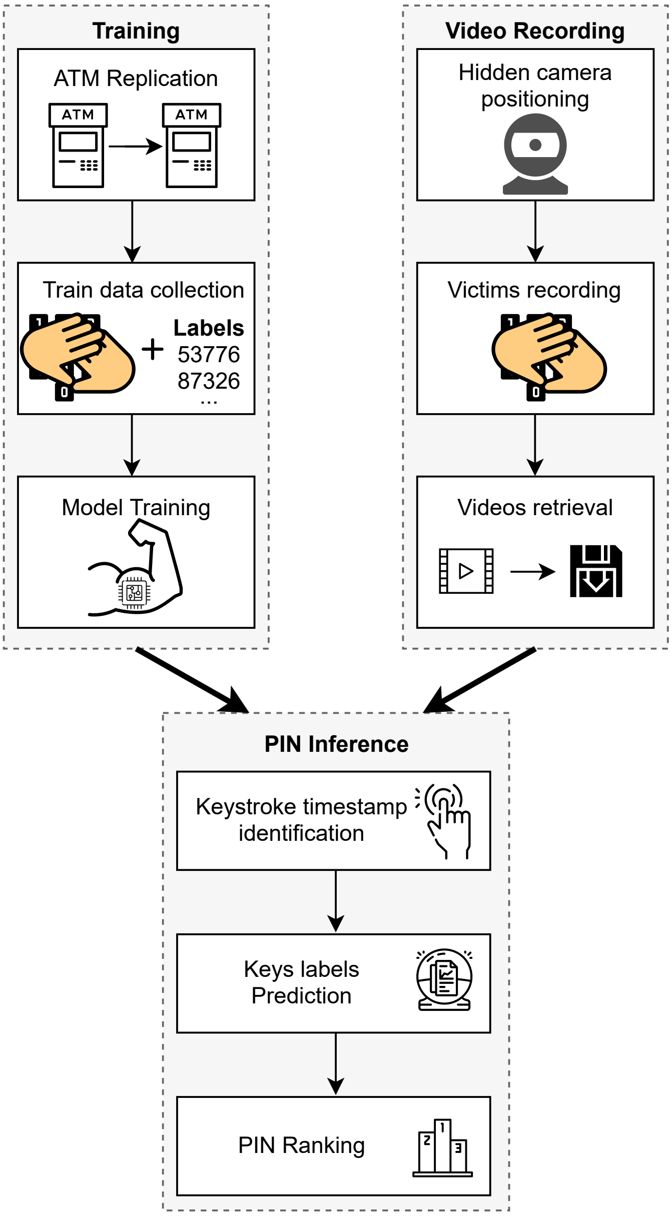


Figure 3.1: The attack step-by-step. The data collection process does not necessarily need to happen before the attacker steals the victim's PIN. Still, it is a required step of the attack.

on this, the attack can be easier or more challenging to succeed. The camera records the victim while entering the PIN and covering the PIN pad with the non-typing hand. The attacker retrieves the recorded video from the remote camera.

Phase C – PIN Inference

The attacker's goal is to infer the victim's PIN based on the video recorded during the PIN entering. First, the attacker retrieves the timestamps from the recorded video. The attacker can use both the pressed keys' feedback sound or the masking symbols appearing on the screen while the victim enters the PIN to perform this task. Leveraging the timestamps, the attacker performs the same procedure as in Phase A to generate an attack set. Differing from the training set, the attack set contains a sequence of frames for each victim key pressed but no information about the related label. The adversary detects in the attack set the frames corresponding to a PIN entry, and splits the video into N sub-sequences where N represents the number of digits composing the PIN. For each sub-sequence, the adversary applies the model trained in Phase A. The model provides the probability of each class (i.e., the ten possible digits) to be the one corresponding to the input sub-sequence. Exploiting the N sub-sequences predictions, the attacker builds a rank of PINs in the descending order of their probabilities. In particular, the probability of a PIN corresponds to the product of the predicted probabilities of its digits.

3.3.2 Attack Settings

We consider three realistic attack scenarios:

1. **Single PIN pad** scenario: the attacker knows the model of the target PIN pad and obtains a copy of it to carry out the training phase. While this scenario may seem unrealistic, we note it is not difficult to obtain a specific keypad copy. Indeed, the attacker can easily obtain information about the ATM to be attacked and then buy the keypad with the same layout. Naturally, there can be certain differences concerning how sensitive the keypad is (for instance, due to usage, pads can become somewhat more difficult to press), but our experiments indicate such differences are not substantial enough to pose issues for deep learning models.
2. **PIN pad independent** scenario: this is the most challenging scenario. The attacker does not know or cannot retrieve the model of the target PIN pad. The training phase is performed on a PIN pad with similar characteristics to the target (e.g., shape, distance between keys, keys layout, and the sensitivity of keys).
3. **Mixed** scenario: as for the *Single PIN* scenario, the attacker knows the target PIN pad model. In this case, the training is performed on two

PIN pads: a copy of the target and at least one PIN pad with similar characteristics. Considering several keypads in the training set makes sense when 1) the attacker is not certain about the keypad model, 2) the attacker assumes that the keypad will behave differently due to environmental conditions, 3) the attacker aims to attack multiple types of keypads (ATMs) with the same machine learning model, and 4) for any reason, the attacker did not manage to obtain enough training examples with a single keypad. We also note that using more keypads in the training set makes the training process more difficult and reduces the chances to overfit (i.e., we can consider different keypads as one keypad with noise, having the regularization effect [31]).

3.3.3 Camera Positions

Since our threat model allows the arbitrary position of the camera, we discuss several representative scenarios. We consider positions at the top of the ATM preferable for the attacker as lower positions of the camera result in no visibility of the hand pressing the keys if the other hand is covering it. We also consider settings at the front side of the chassis as they give better visibility for the attacker and are significantly more difficult for the victim to notice the camera.

Then, without loss of generality, we can discuss three main positions for the camera to provide good results. The camera can be positioned in the top left, center, or right corner. If the camera is positioned in the right corner and the person entering the PIN is right-handed, it will be easier to observe the entered digits. The same happens for the camera in the left corner and the left-handed person. However, if the camera is in the center position, it does not favor any specific setting, making it the most general setting, but it also makes it somewhat more challenging to conduct the attack than the left/right position and left/right-handed persons. We will concentrate on the top center position of the camera mounted on the chassis's front side.

3.4 Experimental Setting

To assess the feasibility of our attack on all the scenarios described in Section 4.2, we collected two datasets containing videos of people covering their typing hands while entering PINs. This section first illustrates the differences between the considered PIN pads and then describes our data collection procedure. Finally, the adopted video pre-processing, the setup used to run the experiments, and the implemented deep learning models are presented.

3.4.1 Devices under Test

We performed two separated data collection campaigns on two different real-world ATM metal PIN pads: DAVO LIN Model *D-8201 F* [5] (Figure 4.3a) and Model *D-8203 B* [6] (Figure 4.3b). In particular, we report the following differences between the two PIN pads:

- Model *D-8201 F* has a dimension of 100 mm x 100 mm, while Model *D-8203 B* has a metal surface of 92 mm x 88 mm and is contoured by rubber protection.
- The horizontal key spacing is 1 mm larger between each key in Model *D-8203 B*.
- The keys of Model *D-8203 B* are harder to press and slightly taller than Model *D-8201 F*.
- For usability reasons, both the PIN pads emit a specific feedback sound (the same for all keys) when a key is pressed. The frequencies of the feedback sounds are 2 900 Hz for Model *D-8201 F* and 2 500 Hz for Model *D-8203 B*.

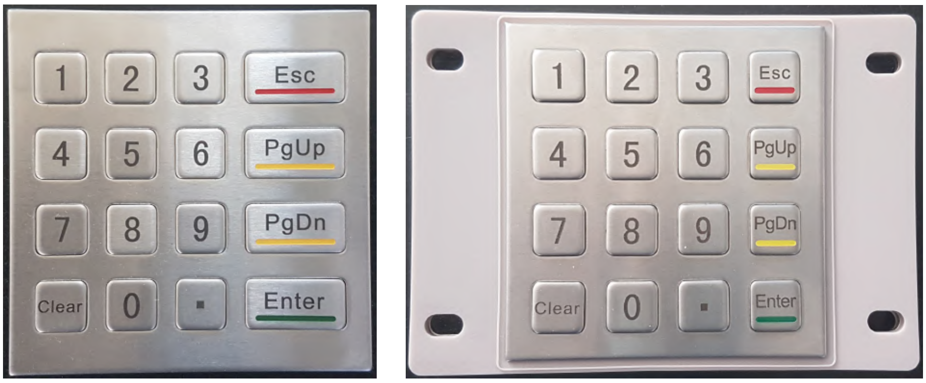
For the data collection, we embedded the PIN pad into a simulated ATM (see Figure 4.2a). We chose the simulated ATM's size based on a real-world ATM [130]. In particular, the simulated ATM has a width of 60 cm, a height of 64 cm, and a depth of 40 cm. At 15 cm of height from the frame's base, we inserted a shelf to position the PIN pad and the monitor. The height of the PIN pad from the ground is 110 cm. We used three *Logitech HD C922 Pro* webcams anchored on the ATM's chassis to perform the video recording. A central webcam was placed 30 cm above the PIN pad, while the other two webcams were placed on the two top corners of the chassis 42 cm away from the PIN pad. The camera's maximum resolution is 1 080p with an acquisition rate of 30 fps. We recorded the videos with a resolution of 720p and an acquisition rate of 30 fps.

3.4.2 Data Collection

The first data collection involved 40 participants (age 38.23 ± 11.43, 24 male and 16 female). The second data collection involved 18 participants (age

[5] https://www.davochina.com/4x4-ip65-waterproof-industrial-metal-keypad-stainless-steel-keyboard-for-access-control-atm-terminal-vending-machine-p00103p1.html

[6] https://www.davochina.com/4x4-ip65-stainless-steel-numeric-metal-keypad-with-waterproof-silicone-cover-p00126p1.html



(a) *DAVO LIN Model D-8201 F* (b) *DAVO LIN Model D-8203 B*

Figure 3.2: The PIN pads used in the data collection.

29.50 ± 5.74, ten male and eight female). Both collections include right-hand participants only. All the participants gave their approval to collect and use the data by signing informed consent. All the data have been anonymized and used by the authors of this paper for research purposes only. Participants were asked to stand in front of the test ATM and cover the typing hand while entering the PIN during the experiment. The participants were left free to type as they pleased. The goal is to emulate an ATM user that is hiding the PIN, preventing possible shoulder-surfing attacks. Each participant typed 100 5-digits PINs randomly generated, divided into four sequences of 25 PINs. This split into four sequences has been performed to include short breaks in the experiments and prevent the participants from getting tired. The PINs were showed one at a time on the ATM screen: once a PIN has been entered on the PIN pad, the user had to press the enter button to move to the next PIN. We recorded a total of 5 800 random 5-digit PINs, resulting in a balanced dataset per digit. Since our study aims to reconstruct the PIN from the video sequence, regardless of the user's typing behavior and familiarity with the PIN or the PIN pad, we decided to randomize PINs rather than asking users to enter the same PIN multiple times. This approach generalizes the attack, which can be applied to mnemonic PINs and One-time Passwords (OTPs). Moreover, we collected the environmental audio (exploiting the webcam microphone) and the keylogs of the PIN pad through the USB interface during the experiment. In particular, for each digit entered, we collect both the key down and key up events. We synchronized the video recordings with the timestamp of the key events. This information was collected to build the ground truth for the conducted experiments. The dataset is available at https://spritz.math.unipd.it/projects/HandMeYourPIN.

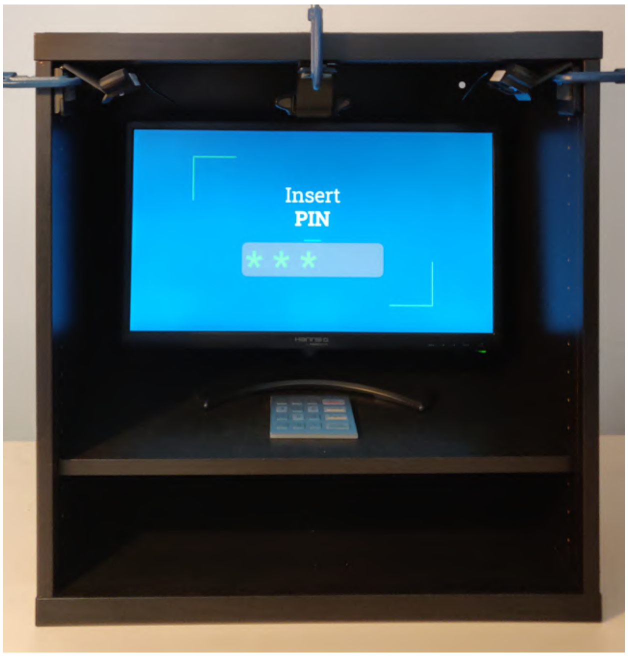


Figure 3.3: Our experimental setup. The cameras are visible but they can be hidden into the frame of an ATM. In all other aspects, we reproduced a common ATM layout in detail.

3.4.3 Pre-processing Video

Once the data acquisition phase is done, we need to pre-process the videos. For each video frame, we applied the following steps: i) convert the video frames to grayscale; ii) normalize the input so that all pixel values lie in the range $[0, 1]$; iii) crop the frames by centering the PIN pad, cutting off the irrelevant part of the background; (iv) resize the image to 250 x 250 pixels. After these steps, we applied a segmentation on each PIN video to obtain sub-sequences of frames corresponding to a single keypress (e.g., 5 sub-sequences for a 5-digit PIN). We extracted the keypress's timestamp from the recorded feedback sound of the PIN pad following the procedure explained in [42]. In particular, we filtered the audio signal using a band-pass filter, centered on the specific frequency of the feedback sound (i.e., 2 900 Hz for Model *D-8201 F* and 2 500 Hz for Model *D-8203 B*). By identifying the peaks of the filtered signal, we could detect the timestamp of the target key (TK). This allowed us to extract a set of frames in each TK neighborhood. For each TK, the maximum number of frames (full-neighborhood) consists of all the frames ranging from the key preceding the TK to the key following the TK. If the TK corresponds to the first digit of the PIN, we consider only the frames between the TK and the next keypress. Analogously, if the TK corresponds to the last digit of the PIN, the frames considered are only those between the TK and its previous keypress. Since our model requires all input samples to have the same length, we decided to keep 11 frames for each sample. This value corresponds to the average number of frames in the full-neighborhood after removing the outliers over 3σ. To keep the TK at the center of the frames' sequence, we decided to consider five frames preceding the target keypress and five frames succeeding it, for a total of 11 frames per sample (including the target frame). There are three borderline cases: the TK is the first digit in the sequence, the TK is the last digit in the sequence, and the full-neighborhood has less than 11 frames. We apply black frame padding to keep the TK at the center of the sequence for these cases. In particular, if the TK is the first digit of the pin, five black frames are added at the head of the sequence, while if TK is the last digit of the PIN, we add five black frames at the end of the sequence. Finally, if there are not 11 frames in a sequence, we pad both the head and the tail (so that the TK is at the center).

3.4.4 Machine Learning Setup

For our experiments, we used a machine equipped with a CPU Intel(R) Xeon(R) E5-2670 2.60GHz, 128GB of RAM, and three Tesla K20m where each GPU has 5 Gb of RAM. To implement the machine learning models, we used Keras 2.3.0-tf (Tensorflow 2.2.0) and Python 3.8.6.

3.4.5 Prediction Models

Our approach aims to predict which key has been pressed on a PIN pad, exploiting only the video of a user covering the typing hand with the other hand. Since we deal with sequences of images, we implemented a model using Convolutional Neural Networks (CNNs) [163] and a Long Short-Term Memory (LSTM) [125]. The CNNs perform spatial feature extraction for each frame of a sequence, while the LSTM exploits these features to extract temporal patterns for the whole sequence of frames. The output of the LSTM passes through a multilayer perceptron (MLP) and a final Softmax activation function layer with ten units (as there are ten digits). This model is known in the literature as Long-term Recurrent Convolutional Network (LRCN) [78]. In Keras [147], such architecture can be implemented using the TimeDistributed wrapper throughout all the CNNs layers, which causes the same convolutional filters to be applied to all the timesteps (i.e., the frames) of the input sequence.

We split our dataset into train, validation, and test sets. Each set's size depends on the attack scenario and is discussed in detail in Section 3.5. We explored different hyperparameters by using the randomized grid search. Based on a preliminary assessment, we set the ranges for specific hyperparameters (i.e., we limit the upper value for specific hyperparameters) to speed up the search. In particular, for the CNNs, we tested $[3x3, 6x6, 9x9]$ kernel sizes. We also varied the number of convolutional layers in the range $[1, \dots, 4]$. In the following dropout layer, we varied the dropout rates in the range $[0.01, 0.05, 0.1, 0.2]$. For the LSTM architecture, we varied the number of layers in the range $[1, \dots, 3]$, and the unit size in $[32, 64, 128, 256]$. We also assessed our network's performance using a Gated recurrent unit (GRU) instead of the LSTM. Finally, we examined the number of layers for the MLP in the range 1 to 4 and the number of units in the range $16, 32, 64, 128$. We tried two types of architectures for MLP: i) all the layers have the same number of units, ii) layers with decreasing number of units (funnel architecture), with every next layer having half the units of the previous one.

After a tuning phase, we selected a structure consisting of four convolutional layers (Conv2D in Keras) with ReLU activation functions, each followed by a pooling layer (MaxPooling2D in Keras). Three convolutional layers have a filter size of 3x3, and one (the second one) has a filter size of 9x9. Each pooling layer has a filter size of 2x2. The number of filters in the convolutional layers doubles at each layer, starting from 32 filters in the first layer, ending up at 256 filters in the fourth layer. We added a dropout layer (dropout rate 0.1) after the last pooling layer to prevent overfitting. The output is then flattened, preserving the temporal dimension to provide a sequence of temporal features to the following LSTM. A single layer LSTM with 128 units resulted in the best validation with a hyperbolic tangent activation function. Finally, for the MLP, we used four fully connected layers, with 64 units each, followed by the Softmax activation layer with ten units (i.e., the number of classes we want to predict). We used the categorical cross-entropy loss function and the stochastic gradient descent (SGD) optimizer. Finally, we set the model to evaluate the accuracy metric. We set the batch size to 16 and the learning rate to 0.1. We tested for 70 epochs since we found that the model always converged within this number of epochs. In Appendix B.1, we provide additional details. Our experiments indicate that the classification task we conduct is relatively difficult, and one needs to use sophisticated deep learning architecture for good results. Still, we note that the architecture we use is in line with the state-of-the-art results for hand tracking problem [132, 162]. Finally, we observed significant changes in the performance depending on the specific hyperparameter choice, indicating a need for detailed tuning for the respective tasks.

In a real-world context, it might not be possible to reproduce precisely the experimental conditions (e.g., the camera might be rotated/tilted slightly concerning the PIN pad, or the distance to the PIN pad might not be the same). Thus, we also used data augmentation to generate synthetic measurements (20% of the training dataset) that cover more scenarios to account for such issues. In particular, we used the following video-based transformations:

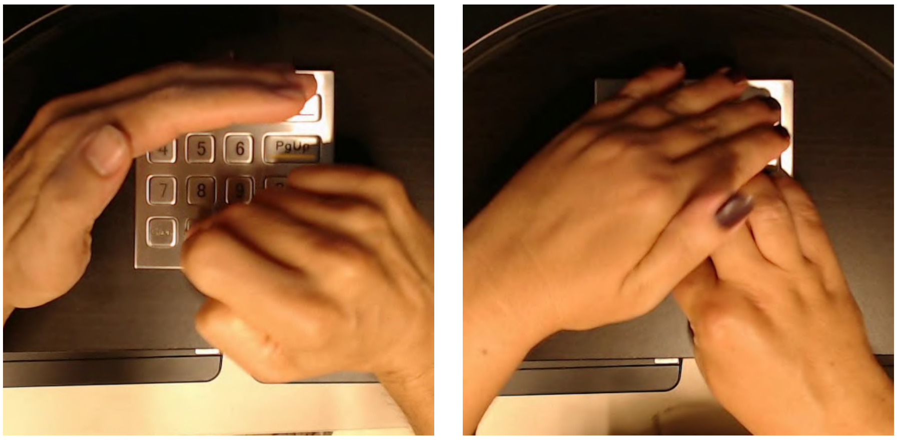
- **rotation** for a maximum of 7 deg both clockwise and counterclockwise;
- **horizontal shift** for a maximum of 10% of the width;
- **vertical shift** for a maximum of 10% of the height;
- **zoom** between 0.9 and 1.1.

Synthetic samples were generated by randomly combining the transformation techniques listed above. We emphasize that data augmentation is also helpful as it makes the predictive model adaptable to different types of ATMs.

3.5 Experimental Results

In this Section, we evaluate the performance of our approach for the three attack scenarios described in Section 3.2. We adopted a user-independent split strategy since, in a realistic context, the attacker does not have labeled videos of victims entering PINs. In this way, we guarantee that videos from a participant appear only once among the three sets. Moreover, since we are interested in evaluating the PINs reconstruction accuracy, we removed all non-5-digit sequences entered by mistake by participants (i.e., the ”enter“ key was pressed after a sequence longer or shorter than 5-digits.) The removed non-5-digits sequences account for 2.2% of the total PINs entered. We conducted the experiments on both 4-digits and 5-digits PINs. To experiment on 4-digit PINs, we removed the last digit of each 5-digit sequence in our dataset.

We define that a PIN is covered when there is no direct view of the entered keys and their surrounding. Still, we observed that some participants failed to obtain a satisfactory coverage level with the non-typing hand despite our instruction before starting the data collection. Since this study aims to infer covered PINs, we decided to exclude the videos of participants that entered badly covered PINs from the validation and test sets. In this way, the validation and test sets consist of videos of covered entered PINs, while the training is composed of videos containing both covered and badly covered PINs. Note that badly covered PINs are still difficult to “read” by simply looking at the video, so we consider such data useful in building a training



(a) *Badly covered PIN that we excluded from the validation and test tests.*

(b) *Covered PIN, where there is no direct view of the pressed key and the surrounding digits.*

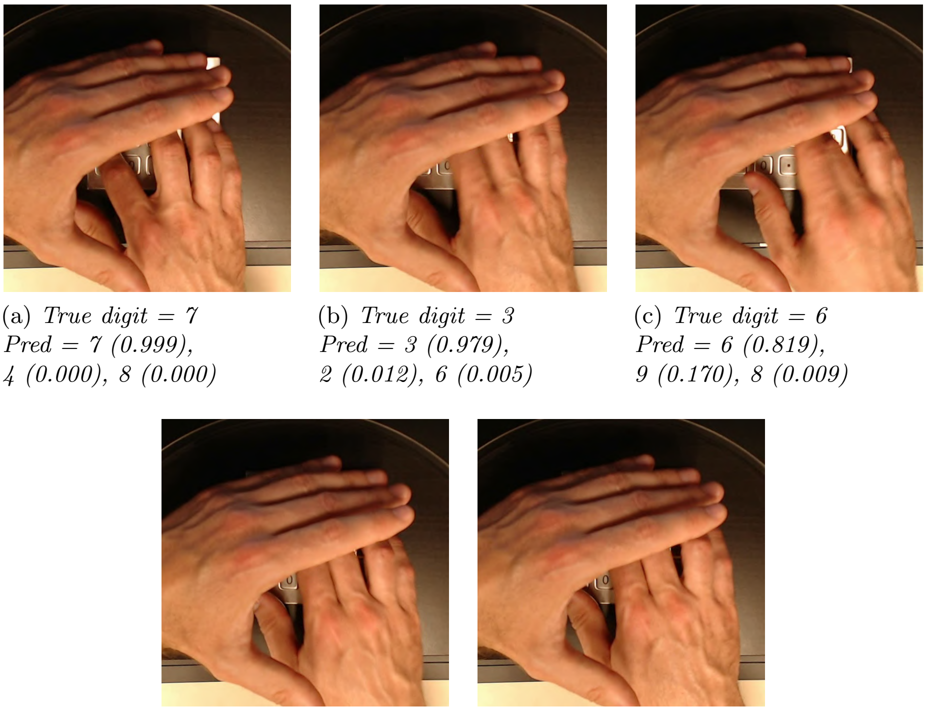
Figure 3.4: Badly covered vs. covered PINs.

set. For the test set, we aim for the most difficult scenario where PINs are properly covered. Under these assumptions, we ”blacklisted“ 16 participants that badly covered the PIN pad: 14 for the first data collection and two for the second data collection. These participants have been excluded from validation and test sets described in the below scenarios [7]. In Figure 3.4, we provide an example of a badly covered PIN and a covered PIN.

To obtain a further indication of the quality of coverage and the difficulty of reconstructing a PIN by a human, we surveyed a random sub-sample of videos of covered PINs (Section 3.7). Finally, there is a question of how to predict the PIN that is not guessed correctly from the first attempt. Since we consider each digit independently, we consider a mechanism where our best guess comprises of individual best guesses (for each digit). If that PIN is incorrect, we consider the digit where the two best guesses have the smallest difference. We change that digit to the second-best guess in our PIN, and we try again. The same procedure is repeated for the third attempt if the second PIN is wrong.

1. **Single PIN pad scenario**. To evaluate the scenario where the adversary knows the target PIN pad model and owns a copy, we considered only the first data collection composed of 40 participants. We applied a user-independent split of the dataset in training, validation, and test sets with the proportions 80/10/10%.

[7] Results are in Appendix B.3



(a) *True digit = 7*
Pred = 7 (0.999),
4 (0.000), 8 (0.000)

(b) *True digit = 3*
Pred = 3 (0.979),
2 (0.012), 6 (0.005)

(c) *True digit = 6*
Pred = 6 (0.819),
9 (0.170), 8 (0.009)

(d) *True digit = 3*
Pred = 3 (0.809),
2 (0.092), 5 (0.069)

(e) *True digit = 3*
Pred = 2 (0.329),
3 (0.315), 6 (0.185)

Figure 3.5: PIN 73633 entered by a user in our test set in the *Single PIN pad* scenario. Our algorithm suggests 73632 as the most probable PIN (probability = 21.32%), 73633 as the second most probable PIN (probability = 20.43%), and 73636 as the third most probable PIN (probability = 11.96%). The algorithm predicts the correct PIN in the second attempt.

2. **PIN pad independent scenario.** In this scenario, the adversary trains the machine learning model on a PIN pad with a similar layout to the target one. This scenario occurs when the attacker cannot obtain the same PIN pad model to collect data. Under these assumptions, we used for training and validation the first collected dataset (composed of 40 participants). We included the videos from 35 participants in the training set and the remaining 5 participants' videos in the validation set. We used the second collected dataset as the test set. We included only the videos of 16 out of 18 participants of the second data collection in the test set since two were in the group that badly covered the PIN pad.
3. **Mixed scenario.** This scenario corresponds to how the attacker owns both a copy of the target PIN pad and a PIN pad similar to the target one. In this case, we merged the two collected datasets and applied a user-independent split in training, validation, and test sets with the proportions 80/10/10%.

We begin the discussion on results by providing an example of a successful PIN attack in Figure 3.5. We consider the 5-digit PIN case and the *Single PIN pad* scenario. We provide an image for each digit. We give the top three digits and the corresponding accuracy values. Notice how the first and second digits are predicted correctly with high probabilities. This happens as the person sets the hand to allow an easy start of typing. Already for the third digit, we observe a significant drop in the accuracy value for the best prediction. Still, the value is significantly larger than the second-best prediction, so there are no issues in getting the correct prediction. This trend continues for the fourth digit and gets very pronounced for the last (fifth) digit. Indeed, the best guess is not correct anymore, but the second-best guess is correct (the difference in probability between those two guesses equals 0.014).

For all three scenarios, Figure 3.6 shows the results for the single key accuracy, while Figure 3.7 reports the results considering 5-digit and 4-digit PINs. Considering the single key accuracy (averaged over all digits), notice that even in the most difficult *PIN pad independent* scenario, our Top-3 accuracy reaches 63.8%, which is significantly higher than the result one would reach with random guessing (30%). At the same time, the results for the *Single PIN pad* scenario and the *Mixed* scenario are rather similar, and the Top-3 accuracy reaches up to 88.7%. Interestingly, we observe somewhat better results for Top-2 and Top-3 accuracy for *Single PIN pad* scenario than the *Mixed* scenario, which is the opposite of the results for 4-digit and 5-digit settings. We hypothesize this happens as we consider independent digits as naturally, the best results happen when the training and test are done on the same device. On the other hand, the *Mixed* scenario gives slightly better results for the PIN reconstruction scenarios as we need to consider a sequence of PINs with the movement between digits. Then, having different devices in the training set allows (slightly) better generalization.

In Figure 6.6e, we observe that the most difficult case is when the attacker does not have access to the same keypad as used by the victim. There, the accuracy for the Top-3 case equals 11.4%. Having access to the same type of keypad improves accuracy in Top-3 to more than 20%. Finally, considering the *Mixed* scenario, we can improve the accuracy for Top-3 to almost 30% (29.7%). Next, in Figure 6.6f, we present results for 4-digit PINs. The results are significantly better than for the 5-digit scenario. The lowest accuracy happens for the Top-1 *PIN pad independent* scenario setting and it equals 10.6% (cf. 6.7% for the 5-digit scenario). The highest accuracy reaches 41.1% for the Top-3 accuracy in the *Mixed* scenario.

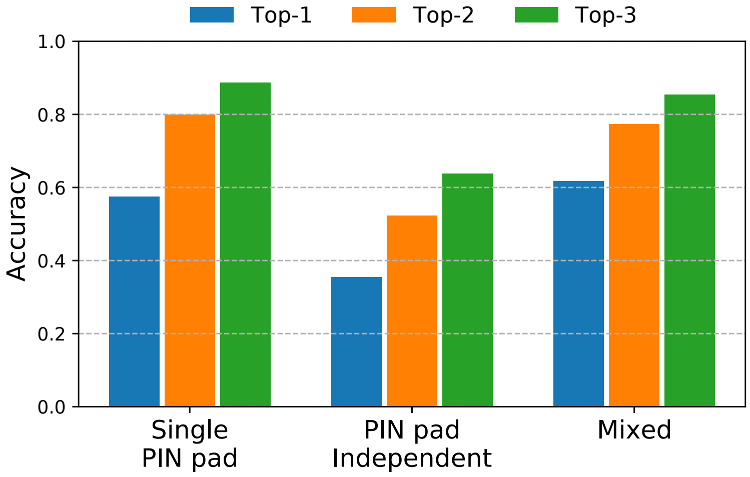
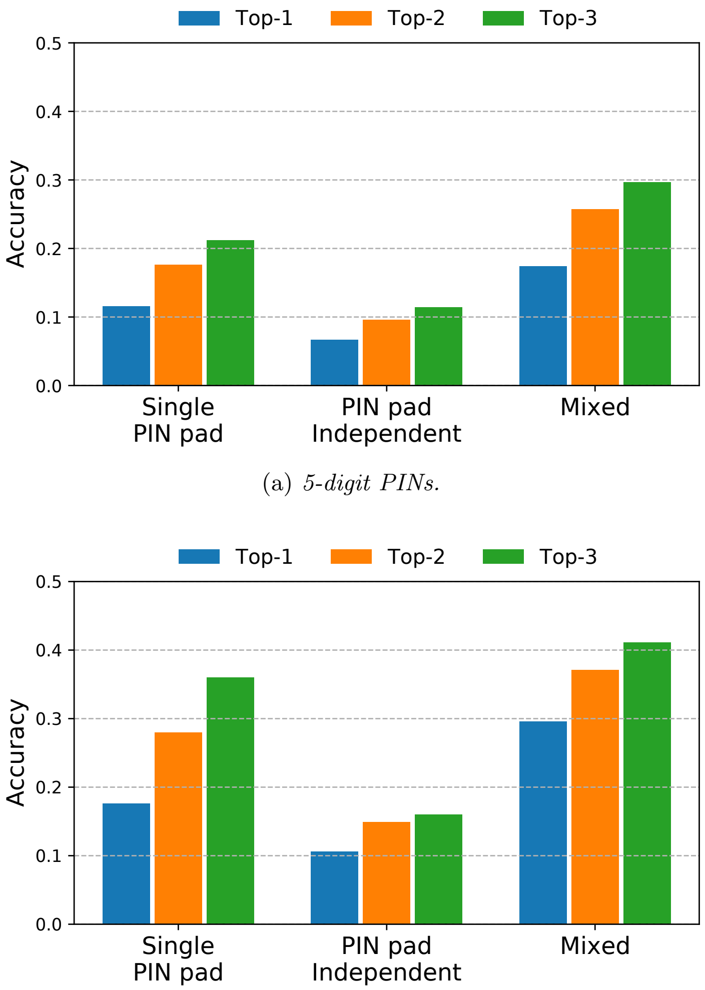


Figure 3.6: Single key accuracy of our algorithm for the three considered attack scenarios. Top-N means that we guessed the digit within the N attempts.

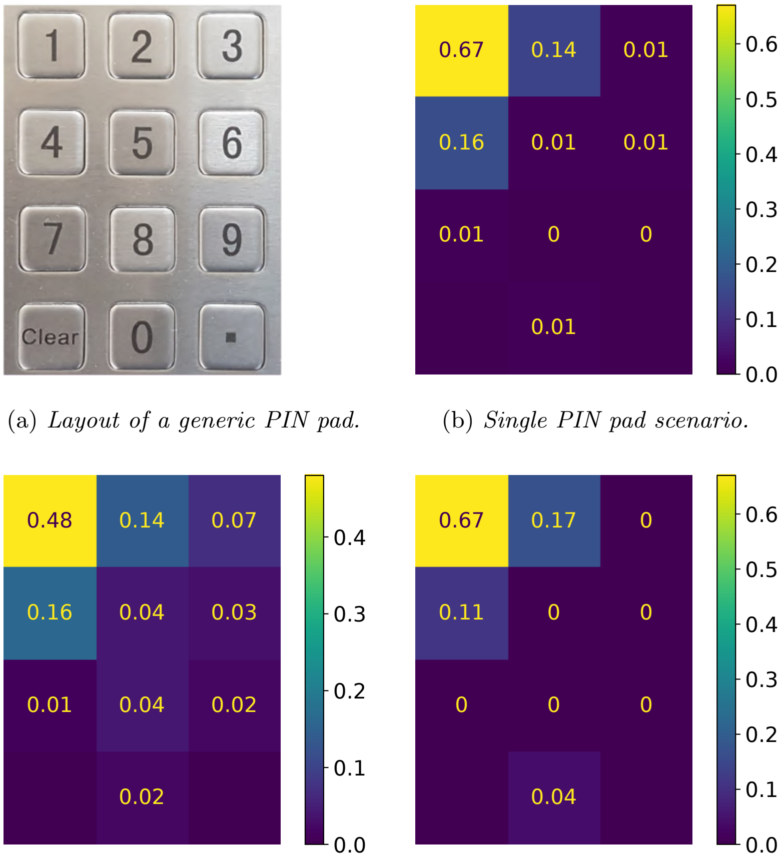
In Figure 3.8, we depict detailed results for the digit 1. We selected this digit since heat maps for others look similar and exhibit similar dispersion. First, in Figure 4.5a, we show the PIN pad layout. Figure 3.8b gives results for the *Single PIN pad* scenario. Notice that the heat map indicates that guess 1 is the most likely one with 67% probability. The digits 4 and 3 are recognized as the second and third best guess, respectively. Still, their probability is significantly lower. For the *PIN pad independent* scenario, we observe that the probabilities are more spread over all digits, which comes at the expense of a lower prediction probability for the correct digit. The second and third best guesses maintain the probabilities, indicating that Top-3 guesses are sufficient to guess a large number of PINs in the most difficult scenario. Finally, Figure 3.8d gives results for the *Mixed* scenario, where we see that the best guess is on the level with the *Single PIN pad* scenario. Interestingly, now the second and third best guesses are swapped compared to the previous scenarios. All the other digits have 0 or negligible probability of being the correct digit. Appendix B.2 provides additional results for the key accuracy.



(a) *5-digit PINs.*

(b) *4-digit PINs.*

Figure 3.7: PIN accuracy of our algorithm in the three considered attack scenarios. Top-N means that we guessed the PIN within the N attempts.



(a) *Layout of a generic PIN pad.* (b) *Single PIN pad scenario.*

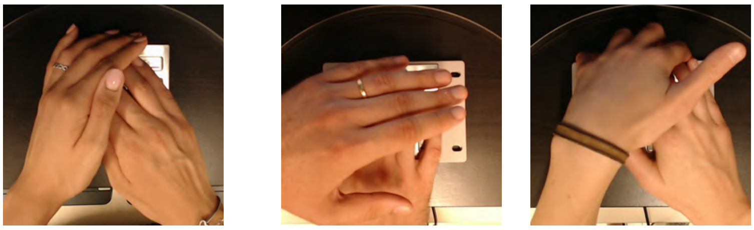
(c) *PIN pad independent scenario.* (d) *Mixed scenario.*

Figure 3.8: Digit 1 predictions heat maps for the three considered attack scenarios.

Based on our results, we provide several observations that we believe generalize beyond these experiments:

- Covering the PIN pad with the other hand is not sufficient to defend against deep learning-based attacks.
- Portability aspect (keypad differences) is quite significant, and the attacker should obtain the same type of keypad for a high probability of success in attack.
- There are three prevailing ways how users cover the typing hand: raised hand not touching the surface, hand resting on fingers and vertically covering the PIN pad, and hand resting on the side of the palm. The examples of all three covering strategies are shown in Figure 3.9.

Finally, Table 3.1 provides a comparison between our attack and several unobtrusive attacks on 4-digit PINs from the literature [42]. We divided the



(a) *Side: hand resting on the side of the palm.*

(b) *Over: raised hand not touching the surface.*

(c) *Top: hand resting on fingers and vertically covering the PIN pad.*

Figure 3.9: Different covering strategies using the non-typing hand.

Attacker Information Source				**4-digit PINs TOP-N Accuracy (%)**		
KT	**OD**	**TT**	**Our Attack**	**TOP-1**	**TOP-2**	**TOP-3**
				0.01	0.02	0.03
	§			0.10	0.20	0.30
§				0.02	0.35	0.72
§	§			3.02	3.72	4.36
		§		3.76	7.52	11.28
§		§		15.54	27.79	33.63
			§	**29.61**	**37.06**	**41.12**

Table 3.1: Comparison of our attack with other unobtrusive attacks on ATM PIN pads. Note that we need to extract the frame for our attack, while for KT, one needs to use the timestamp, which is more precise information.

attacks according to the information that the attacker has: keystroke timing (KT), one digit of the victim's PIN (OD), and the thermal trace (TT) left on the PIN pad by the victim [2]. From the results, it is clear that our attack performs the best for all considered TOP-N accuracies.

Appendix B.3 provides experiments where we: i) resize the images, ii) consider different camera positions, iii) consider setup without data augmentation, and iv) consider the training set that includes the blacklisted participants. Finally, we also provide experiments for the frame detection error (when the feedback sound is not properly synchronized).

3.6 Countermeasures

Different countermeasures could make the attack more difficult to succeed. For instance:

Coverage percentage	Key accuracy	PIN TOP-3 accuracy
25%	0.54	0.22
50%	0.55	0.22
75%	0.50	0.17
100%	0.33	0.01

Table 3.2: PIN shield experiments.

1. Longer PINs. This countermeasure would make the attack more difficult, as evident from the comparison for 4- and 5-digits PINs. This countermeasure would be relatively easy to support from a technical perspective. At the same time, it would have usability drawbacks as longer PINs take more time to type and are more difficult to remember.
2. Virtual and randomized keypad. Instead of using a mechanical keypad, one could consider using a touchscreen where the digits are randomized. More and more ATMs (but not PoS) have this feature, so implementing it would not be too difficult. Unfortunately, we believe this would seriously damage the usability aspect as people are accustomed to digits occurring in the natural sequence, and any changes would probably result in wrongly entered PINs.
3. Screen protectors. On many ATMs, there are already various types of screen protectors that occlude the typing hand. To maintain usability, many screen protectors are short and will not cover the whole typing hand. Making the screen protectors larger would impair usability as it will become more difficult for the user to read the keypad. This countermeasure is potentially not easy to deploy as it requires physical changes to the ATMs.

Next, we analyze how a PIN shield could affect the performance of our attack. We simulated the presence of the shield by applying a black patch to cover the PIN pad. In Table 3.2 we report the performance of our attack in the *Mixed* scenario, applying four different levels of coverage (Figure B.5, Appendix B.3). The coverage of the PIN pad is larger than the percentage shown in Figure B.5 since the coverage given by the non-typing hand is not included in the given percentage. The results show that our attack remains effective even when 75% of the PIN pad is covered, while the performance decays significantly beyond this level of coverage. As such, it becomes clear that our deep learning attack uses information about the whole hand position and movement, and not only the tip of the fingers. Since the last row of the PIN pad has only one number (0), 100% coverage has poor attack results not only because of hiding all the numbers on the keypad but due to hiding of proximal interphalangeal, metacarpophalangeal, and carpometacarpal joints of the fingers. Thus, only PIN shields that offer full PIN pad coverage can be considered effective countermeasures to our attack.

We provide additional results with different covering strategies (`Side`, `Over`, and `Top`) in Appendix B.3. Those results again show that covering the PIN pad from the `Side` gives insufficient protection. On the other hand, using the `Over` strategy significantly decreases the key accuracy and PIN accuracy.

3.7 Deep Learning vs. Humans

If an attacker has direct visibility of the PIN pad, reconstructing a PIN from a video can be considered a trivial task. One of the classic countermeasures to the so-called shoulder-surfing attacks is to cover the hand entering the PIN with the non-typing hand. In this way, the victim obstructs the attacker by removing the direct visibility of the keypad. We designed a questionnaire to evaluate how much the covering with the non-typing hand effectively prevents the PIN reconstruction.

3.7.1 Methodology

The questionnaire consists of 30 videos of people entering 5-digit PINs by covering the PIN pad with the non-typing hand as we noticed that for longer questionnaires, the participants' attention significantly goes down toward the end. For each video, the participants had to indicate the three most likely PINs in their opinion.

To assess human and model performance on both the PIN pads, we decided to use the test set of the *Mixed* scenario (i.e., the only one including both PIN pads). Since the test set was balanced in terms of samples per user, we randomly selected five PINs for each of the six users in the test set. We extracted 30 videos corresponding to the selected PINs from our dataset. We kept the original resolution of 720p and the original audio track containing the feedback sound emitted by the PIN pad for each video. The feedback sound helps the participants to recognize when a digit is entered. To avoid bias in the answers, we randomized the order of the videos in the questionnaire. Moreover, the participants were free to modify all their answers until the final submission. We did not apply any particular restriction to the participants during the filling of the questionnaire. In particular, there were no time restrictions to complete the task. The participants could freely apply the strategy they prefer to infer the PIN (e.g., write down the digits, pausing the video, restart the video any number of times, use the slow-motion option). Finally, we provided the users with the layout of the PIN pad.

To evaluate if people with specific knowledge about the task achieve a better performance, we pre-trained a group of participants. Specifically, we provided participants with a new set of 20 videos of users typing PINs by covering the PIN pad with the non-typing hand and the corresponding typed PIN. To make the training more effective, we decided to provide participants with videos of users included in the questionnaire (none of the videos are present in both training and questionnaire). Additionally, the questionnaire had suggestions on what to pay special attention. For a participant to be considered trained, the complete viewing of all 20 videos is required. In addition, trained participants could also watch the training videos while filling the questionnaire.

3.7.2 Evaluation and Discussion

A total of 78 distinct participants took part in our questionnaire experiment. In particular, 45 participants (14 female age 34.1 ± 10.4 years and 31 male age 29.7 ± 8.3 years) completed the experiment without any training, while 33 participants (10 female age 29.1 ± 3.3 and 23 male age 29.3 ± 5.6) completed the experiment after the training session. None of the questionnaire participants took part in the two data collections described in Section 3.4.2.

The proposed questionnaire's goal is twofold: i) investigate how effective the hand coverage is in preventing a PIN from being inferred by a human, and ii) compare the performance of our deep learning approach with that of a human. Although the coverage of the PIN pad provides an obstacle to the immediate identification of the typed PIN, a human can exploit various information (both local and global) to reduce the probability space about where to look for the entered PIN:

- Knowing the keys' spatial positioning thanks to the given layout of the target PIN pad.
- Understanding which finger pressed the key from the movements of the hand.
- Evaluating the topological distance between two consecutive keys from the feedback sound emitted by the PIN pad. Specifically, two topologically close keys have temporally close sound feedback [42].
- Excluding keys based on the non-typing hand coverage.
- Guessing the finger position based on the hand displacement between the insertion of a key and the next one.

- Deducing the fingers' position of the covered hand.

Although a human can exploit this information, the PIN pad coverage still partially prevents PIN reconstruction. In particular, the participants in our questionnaire could reconstruct on average (of both trained and non-trained humans) only 4.49% of the PINs entered in the videos on the first attempt and 7.92% within three attempts. The performance increasing between Top-1 and Top-3 accuracy suggests a certain ability in estimating the neighborhood of the keys pressed. This ability is also highlighted in Figure 3.11a, where the probability distribution shows how the error decreases with the increase of the topological distance from the target key. The heat maps for other keys look similar and exhibit similar dispersion.

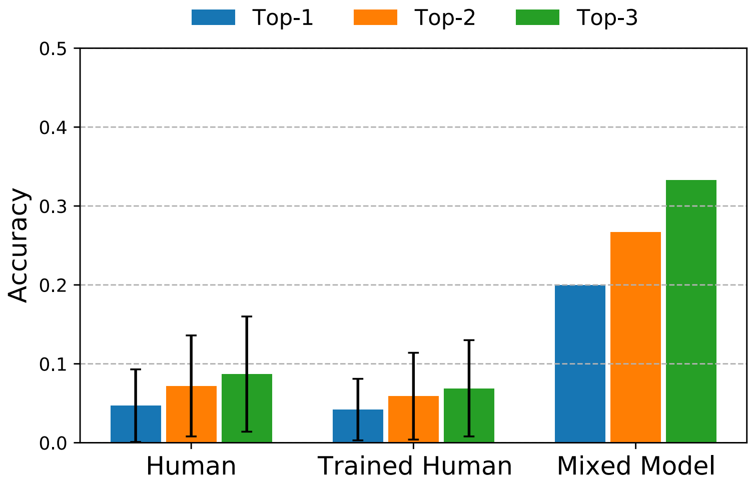
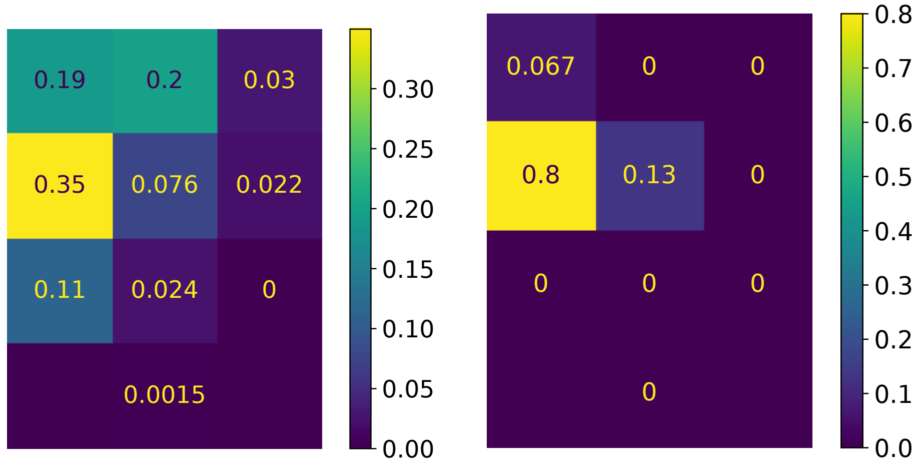


Figure 3.10: Comparison between human (non-trained and trained) and deep learning model performance in the sub-set of videos included in the questionnaire. Top-N means that participants guessed the PIN within the N attempts.

Unlike humans, our algorithm focuses on target key classification and then reconstructs the entire PIN sequence. To compare the model's performance to that of humans on the same task, we evaluated our algorithm's accuracy on the videos included in the questionnaire. Recall that the questionnaire's videos are a sub-sample of the *Mixed* scenario test set, and therefore were not used in the model training phase. As reported in Figure 3.10, our model performs better than humans in all Top-N accuracy scenarios. To evaluate if our algorithm performance and humans' performance in reconstructing 5-digit PINs are statistically different, we applied a series of Chi-square tests [184]. The Chi-square test resulted significant for all Top-1 ($\chi^2 = 14.19, p < 0.001$), Top-2 ($\chi^2 = 15.84, p < 0.001$), and Top-3 ($\chi^2 = 21.37, p < 0.001$) accuracy values for non-trained humans. In particular, our model outperforms humans showing a four-fold improvement in reconstructing a PIN in three attempts. Similarly, for trained humans, the Chi-square test resulted significant for all Top-1 ($\chi^2 = 16.12, p < 0.001$), Top-2 ($\chi^2 = 20.83, p < 0.001$), and Top-3 ($\chi^2 = 28.88, p < 0.001$) accuracy values.

This result comes from the difference in performance in the classification of single keys. The human average accuracy (considering both human data collections) on single key classification equals 0.351, approximately half compared to the model key accuracy of 0.687. The comparison of Figures 3.11b and 3.11a shows how the error in identifying a digit is significantly higher for humans, justifying why the increase in Top-2 and Top-3 PIN accuracy is greater for our algorithm. Finally, comparing trained and non-trained humans, the Chi-square test reported no significant differences with $p > 0.1$ for all Top-1, Top-2, and Top-3 accuracy values. This means that training does not improve a human's ability to identify a PIN within three attempts. Potentially, either a longer training could be required, or additional feedback from an expert should be provided to improve the performance. Appendix B.2 provides additional results for the comparison between our deep learning model and human performance.



(a) *Humans.*

(b) *Mixed* scenario model.

Figure 3.11: Digit 4 predictions heat maps for the videos included in the questionnaire. We report an example from non-trained humans, since the heat maps for both non-trained and trained human are similar.

3.8 Summary

This paper proposed a deep learning attack on PIN mechanisms reaching high accuracy even when the user covers the PIN to be entered. Our attack leverages the information from the hand position but also hand movements while entering the PIN. Our attack works in the profiling setup where the attacker uses a copy of the keypad to train the deep learning model and then attacks a different device while the victim is entering the PIN. For a 4-digit PIN, our attack reaches an accuracy of more than 40%, making it practically applicable and more powerful than the attacks from the related works.

Our data collection phase involved 58 persons, and our questionnaire involved 78 participants. While this required a significant effort and several months of data acquisition, one could still consider the datasets too small to allow general conclusions. Next, our analysis considered only two types of keypads. While most keypads do not have significant differences, including more keypad models in our analysis would be interesting. Additionally, there are several potential sources of bias in our data collection phase. While we managed to get a relatively good male and female participants ratio, we notice that data is skewed from several perspectives. Unfortunately, this was not possible to avoid as the participation was voluntary [8].

1. Our dataset has users ranging from 24 to 50 years. While this provides good variety, it would be good if it included older people. Still, we do not expect any difficulties in running our attack. We consider it even somewhat easier as we noticed older people make more significant hand position adjustments when entering the PIN.
2. Our analysis includes only right-handed persons. We do not expect any issues due to the dataset's limitations as we use a camera positioned in the center. Still, we expect the attack to be more difficult when attacking left-handed persons if the training set does not contain such examples. Finally, from the real-world practicality, there are approximately 90% of right-handed persons vs. 10% left-handed persons [201], so our attack generalizes for the dominant part of the population.
3. All participants were Caucasians. We expect our attack will have difficulties working for people from other races. Still, this can be alleviated by expanding the training set to include more racial diversity.

Possible future work includes:

1. In our data collection phase, we allowed the users to select their covering strategies. Based on the current results, it would be interesting to

[8] The 2021 COVID-19 situation made data acquisition more challenging as participants needed to be in our lab during the data acquisition.

explore if modifications in how the user covers the PIN would allow more protection.

2. We noted several potential sources of bias in our data collection phase. Including participants from other races and left-handed persons would allow us to make more general conclusions.
3. To avoid the need that the attacker should have different keypads, it would be beneficial to assess whether some more straightforward solution like a paper copy of the keypad would suffice (at the expense of losing information about the keypress sensitivity).
4. It would be interesting to investigate if it is possible to extract the timestamp directly from the video (when a person clicks a button, there is a specific movement).

Chapter 4

PinDrop: Acoustic Side-Channel Attacks on ATM PIN Pads

The Automatic Teller Machines Industry Association estimates that over 300 million ATMs are deployed worldwide[1]. In the US alone, over 10 billion ATM transactions are performed every year [195]. ATMs have now become an indispensable part of the self-service banking ecosystem. An ATM typically uses a unique physical card (which a customer possesses) along with a PIN (which a customer remembers) to form a two-factor authentication system, wherein the card uniquely identifies the customer account and the PIN identifies the customer.

In recent years, there have been many attacks aimed at PINs and at information encoded on ATM cards. Such attacks are broadly referred to as skimming operations [247], whereby criminals usually install a card-reader-like device to trick customers into placing (or inserting) their cards and copy the information. This is often done in tandem with installing a video camera on the ATM (or in its vicinity) at an angle that allows the criminal to record PIN entry [230]. Recently studied attacks on PINs (e.g., [15, 42, 261]) went one step further and showed that the attacker does not even have to see the PIN. These side-channel attacks use a recording device (e.g., a video camera [15], a microphone [42], or a thermal camera [261]) placed near the ATM to collect information and use it to infer customers' PINs.

In this chapter, we present a new side-channel *PinDrop* attack on ATM PIN entry. It consists of two steps: (1) the attacker builds an acoustic profile

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

(a signature of click sounds) for each key on the target PIN pad, and (2) at PIN entry time, the attacker records audio emitted by each pressed key and compares them to the acoustic profile to infer the actual keys pressed, thereby learning the PIN. These two steps can be carried out in any order.

Contributions. The main contributions of this chapter are:

- We describe a novel attack targeting PINs: *PinDrop*, based on acoustic emanations from commodity ATM PIN pads. We demonstrate that *PinDrop* reconstructs up to 94% of 5-digit PINs and 96% of 4-digit PINs within three attempts. We show that the threat posed by *PinDrop* is higher compared to state-of-the-art acoustic side-channel attacks on ATM PIN pads [42, 167, 200].
- We evaluate *PinDrop* via extensive experiments, collecting acoustic emanations for 5,800 5-digit PINs entered in a simulated ATM (though using real PIN pads) by 58 distinct participants. The resulting dataset is publicly available [2] to the research community. We believe it will be useful in studying the problem further and developing countermeasures.
- We analyze the performance of *PinDrop* with two recording distances: 0.3 and 2 meters away from the PIN pad. At the distances of 0.3 and 2 meters, up to 96% and 57% (respectively) of 4-digit PINs were correctly learned in three attempts.
- We demonstrate the feasibility of *PinDrop* on two commercially available ATM PIN pad models. The success rate of PIN guessing on both pads is about the same for each distance.
- We analyze the impact of training set size on the performance of *PinDrop*. We evaluated two important factors: the number of attackers participating in the Profiling Step, and the number of digits collected by each attacker. We showed that including training samples from multiple attackers is an effective strategy for appreciably improving attack success rate.
- We assess the performance of *PinDrop* in noisy environments, considering different levels and sources of noise to simulate real-context scenarios. We showed that *PinDrop* is still an effective attack at 2 meters with low/moderate noise, while it remains effective under any noise condition at 0.3 meters.

[2] Dataset link: https://spritz.math.unipd.it/projects/PINDrop

4.1 Related Work

This section overviews attacks based on acoustic emanations from user input devices. We first consider attacks targeting keyboards, followed by those targeting PIN pads. For a comprehensive discussion of keyboard side-channel attacks, we refer to [190].

Attacks on generic keyboards. The first extensive study on keyboard acoustic eavesdropping was conducted by Asonov and Agrawal [10]. It showed that each key can be identified by the unique sound that it emits when pressed. This work investigated the reasons for this behavior, demonstrating that it can be attributed to the placement of keys on the keyboard plastic plate. In particular, when different keys are pressed, the plate produces emits sounds with different timbers.

Subsequent efforts to infer key sequences from acoustic emanations are based on two types of approaches: (i) extraction of features that allow exploiting the uniqueness of acoustic emissions of pressed keys, and (ii) extraction of temporal information. The former tries to distinguish among keys by their characteristic sound, and relies on either supervised [10, 116, 117, 180] and unsupervised [26, 280] machine learning models, depending on the specific attack scenario. Supervised models exploit features, notably Fast Fourier Transform (FFT) coefficients and their derivatives, such as Mel-frequency cepstral coefficients (MFCCs). Supervised algorithms generally achieve better performance in identifying keystrokes. On the other hand, these models have a greater dependence on the keyboard used in training and the users' typing style. A further weakness of supervised algorithms is the need to collect a labeled dataset to be used as a training set. Indeed, the ground truth collection is not a trivial task and could significantly affect the attack's effectiveness. One possible solution is discussed in [6, 46]. which take advantage of the audio recorded during a VoIP call to collect a ground truth dataset directly. In this scenario, the attacker can exploit the text typed by the victim in a shared medium (e.g., in the VoIP chat or an email sent to the attacker during the call) to label the keystroke sound.

Unsupervised methods are used to group collected samples into unlabeled clusters. The label-cluster association is made by exploiting the characteristics of the input language. In particular, Zhuang et al. [280] perform labeling using letter frequency, while Berger et al. [26] make an association by selecting words from a dictionary that match specific constraints. Unsupervised approaches overcome the need for a ground-truth dataset. However, the scenarios where these attacks can be applied are limited by the strong assumptions on input text and therefore their performance drastically declines on random letter sequences.

The second approach involves the extraction of temporal features of pressed keystrokes. To this end, many efforts focused on analyzing the Time Difference of Arrival (TDoA) information. They use one (e.g., Liu et al. [166]) or more (e.g., Zhu et al. [278]) microphones positioned around the input device.

Pin Pad-focused Attacks. PIN pads are numeric keypads specifically designed for Point-of-Sale (PoS) terminals and ATMs, They facilitate users to enter their Personal Identification Numbers (PINs). Attacks on PIN pads tend to be different from those on regular keyboards. For instance, it is rather challenging to apply unsupervised techniques with PIN pads since the assumptions about the victim's language are no longer applicable. However, the other types of attacks, such as those based on the uniqueness of the acoustic emission and those based on the temporal information are usually applicable. PIN pads also prompt a new set of assumptions, usually dictated by the specific conditions under which they operate. This paves the way to new and more efficient side-channel attack scenarios. Below, we briefly discuss these attacks.

Balagani et al. [15] demonstrates how to obtain PIN information by exploiting inter-keystroke timings. This information is leaked by recording the timing of appearance of masking symbols (usually, asterisks) on the screen while the victim is entering the PIN. On a related note, Cardaioli et al. [42], show how inter-keystroke timing information can be inferred with higher accuracy from the feedback sound emitted by the PIN pad when a key is pressed. It also shows that combining multiple side-channel information (e.g., inter-keystroke timing and thermal residue) can significantly improve the probability of reconstructing a 4-digit PIN. Similarly, Liu et al. [167], propose a user-independent attack based on inter-keystroke timing on a plastic PIN pad.

PIN pad acoustic emanations can also be used to improve security of PIN-based authentication systems. For example, Panda et al. [200] show that inter-keystroke features obtained from PIN pad-emitted audio, can be used as an additional layer of authentication. The same work also showed how to perform a close-by attack (i.e., with the microphone placed a few centimeters from the PIN pad) on an arbitrary subset of keys. Exploiting the inter-keystroke features on this subset, a 60% accuracy in the identification of the pressed key can be reached. Acoustic information is also used by Souza Faria and Kim [72], where a Point-of-Sale (PoS) terminal is tampered with by inserting multiple microphones into it. This allows identifying the pressed key position using triangulation, reaching the average accuracy of 88% for a single key, on three PoS models. Although very effective, this approach requires full physical access to the PoS, thus reducing the attack's applicability and scalability.

4.2 *PinDrop* Attack

Assumptions: We assume that the victim interacts with a generic ATM, performing PIN-based authentication. The ATM is equipped with a PIN pad that emits a feedback sound when a key is pressed. The feedback sound (as perceived by the human ATM users) is the same for all keys. The attacker aims to learn the victim's PIN by placing a microphone near the ATM to record acoustic emanations of the PIN pad. The microphone stores recorded audio. How the microphone stores that audio is not relevant for *PinDrop*, i.e., it can be stored locally or off-loaded to a remote site. *PinDrop* attack relies only on that recorded audio.

Preliminaries: To set up *PinDrop*, the attacker must select a target ATM and hide a microphone nearby. The exact placement of the microphone can vary, though in the *PinDrop* setting the maximum distance form the PIN pad is 2 meters (just over 6′):

1. Concealed on the attacker's body, in case of a real-time attack. Albeit, strictly speaking, concealment is not required, since a regular smartphone microphone can be used, and it need not be hidden from view (as it is unlikely to arouse suspicion).
2. On any surface (walls, floor, ceiling) near the ATM. In this case, it might be in plain sight, especially, if its size and shape are inconspicuous enough not to be noticeable. It could also be partially hidden from view (e.g., behind a column or a light fixture), or even within or behind some normal-looking object, e.g., a vent, a light-switch or a garbage can.

As shown in Figure 4.1, *PinDrop* consists of four phases: 1) PIN Recording (Section 4.2.1), 2) Data Processing (Section 4.2.2), 3) Model Generation (Section 4.2.3), and 4) PIN Inference (Section 4.2.4),

4.2.1 PIN Recording

The goal of this this phase is to come up with two datasets (training and attack) with audio recordings of entered PINs. This takes two steps:

A.1 **Audio Recording** using a microphone placed near the ATM.

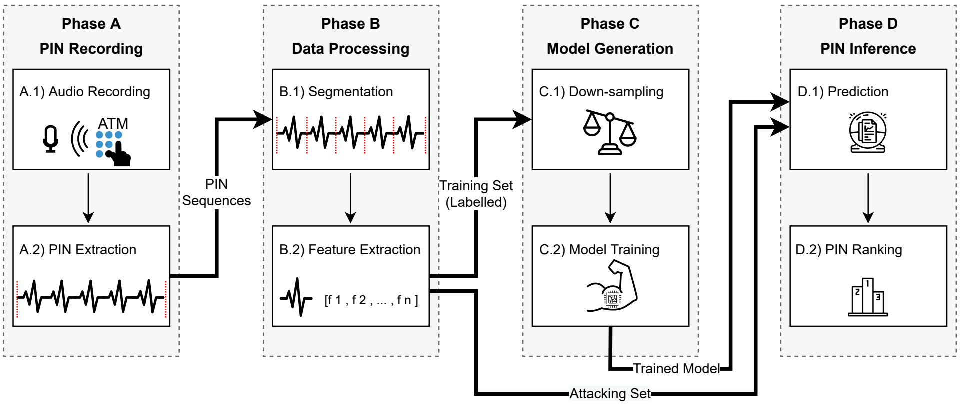


Figure 4.1: $\mathcal{P}$*inDrop* attack phases.

A.2 **PIN Extraction**, i.e., isolation of the sequences of feedback sounds emitted by the PIN pad, given the knowledge of the number of digits in the PIN, e.g., the beginning and the end of the 6-digit PIN entry.

To build the *training set*, the attacker must enter a set of PIN sequences on the target PIN pad. The sequences must be representative of all ten numeric keys. Once this step is completed, the attacker has a table of entered PINs and their corresponding audio. The *attack set* consists of the audio recordings entered by the victim.

4.2.2 Data Processing

This phase is conducted on the data entered by both the attacker and the victim. It also consists of two steps: segmentation of the PIN audio signal into individual key-press sounds, and extraction of corresponding features.

B.1 **Segmentation:** The attacker uses the feedback sound emitted by the PIN pad as a signal that a key has been pressed. This can be achieved via the characteristic frequency of the feedback sound, as in [42]. The attacker segments the signal, using time windows centered at the detected key-press. The window size is chosen to comprise the entire audio segment related to a single key-press.

B.2 **Feature Extraction:** The attacker extracts features descriptive of a key-press sound. Prior results show that short-term power spectrum can be used for this type of a classification problem. In particular, [46] shows that mel-frequency cepstral coefficients (MFCC) [169] achieve the best performances for discriminating among the sounds of different keys. This step yields two feature sets: (1) a labeled training, and an (2) unlabeled attacker.

4.2.3 Model Generation

This phase is applied to the labeled training set in order to train a classifier.

C.1 **Down-sampling:** Since we make no assumptions about how often a victim uses a specific digit in the PIN, it may be necessary to under-sample the data by classes before proceeding with training. The under-sampling leads to a balanced dataset and mitigates over-fitting.

C.2 **Model Training.** The attacker trains a multi-class classifier to predict the digit based on its emitted key-press sound. The class labels output by the classifier are the keys (digits) of the PIN pad. Together with the predicted digit, classifiers also output the prediction probability of each class.

4.2.4 PIN Inference

In this phase, the attacker utilizes the trained classifier to guess a victim's PIN. The output is a sequence of all possible PINs ordered by probability. This ordering allows the attacker to minimize the number of attempts to guess the PIN. In a real-life setting, ATM cards are usually blocked after three failed attempts. This phase involves two steps:

D.1 **Prediction:** The attacker reconstructs the PIN entered by the victim applying the classifier trained in the previous phase to the attack set. As input to the classifier, the attacker feeds the features of a single key of the victim's PIN. This is repeated for each digit of the PIN.

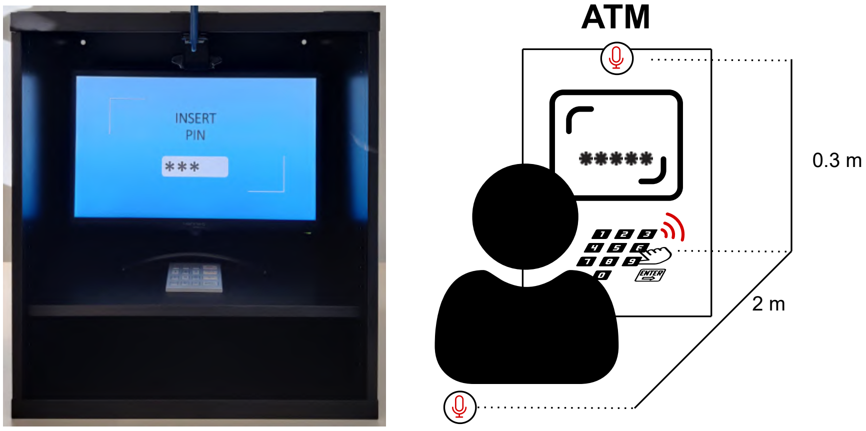
D.2 **PIN Ranking.** The classifier yields a probability for each digit to be the one actually pressed by the victim. Combining the probability set of each input, the attacker builds a ranking of the most likely PINs. The probability assigned to a PIN is the product of the probability of each digit in that PIN.

4.3 Experimental Setting

To assess the feasibility of $\mathcal{P}inDrop$, we collected a large dataset of keystroke sounds, as detailed in this section.

4.3.1 Data collection

We performed two separate data collection efforts on two commercially available (commodity) metal PIN pads: DAVO LIN Model *D-8201 F* (Figure 4.3a) [3] and Model *D-8203 B*(Figure 4.3b) [4]. For clarity's sake, we refer to *D-8201 F* as *PAD-1* and *D-8203 B* as *PAD-2*. For usability reasons, both pads emit a specific feedback sound (the same for all keys) when any key is pressed. In all experiments, we embedded each PIN pad into a simulated ATM (Figure 4.2a).



(a) *The simulated ATM.* (b) *The testbed configuration used in the experiments.*

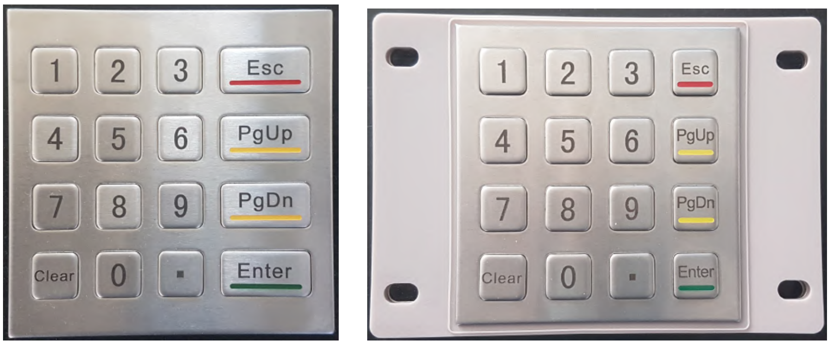
Figure 4.2: $\mathcal{P}inDrop$ experimental setup: The photo on the left shows the ATM layout. The figure on the right show the position of the microphone with respect to the ATM and the victim: the closer one at $0.3m$ is placed over the PIN pad, while the farther one at $2m$ is placed in front of the ATM, and behind the victim.

The simulated ATM size is based on a real ATM [130]. It is 0.6m wide, 0.64m high, and 0.4m deep. At 0.15m above the ATM base, we inserted a shelf upon which we placed the PIN pad and the monitor. The keyboard is 1.1m above the ground. To record keystroke sounds, we used the microphones of two *Logitech HD C920 Pro* webcams: one placed on the ATM's chassis 0.3m above the PIN pad, and another microphone – 2m in front of the ATM, as shown in Figure 4.2b.

The first data collection effort involved 38 participants (23 male and 15 female, average age 38.97 ± 11.36), while the second involved 20 participants (11 male and 9 female, average age 29.50 ± 5.74). Together, that makes the total of 58 participants who entered 5,800 5-digit PINs. We used both these

[3] https://www.davochina.com/4x4-ip65-waterproof-industrial-metal-keypad-stainless-steel-keyboard-for-access-control-atm-terminal-vending-machine-p00103p1.html

[4] https://www.davochina.com/4x4-ip65-stainless-steel-numeric-metal-keypad-with-waterproof-silicone-cover-p00126p1.html



(a) *PAD-1: DAVO LIN Model D-8201 F*

(b) *PAD-2: DAVO LIN Model D-8203 B*

Figure 4.3: Two commodity metal PIN pads we used.

data collections to obtain datasets of 4-digit PINs by removing the last key entered by the participants from each 5-digit PIN.

During the experiments, participants were asked to stand in front of the simulated ATM, and remain silent for the duration. A participant's task consisted of typing 100 5-digits PINs randomly generated, divided into four batches of 25 PINs. This split was made to allow for short breaks betwen batches in order to lower fatigue. PINs were displayed one at a time on the ATM screen: once a PIN is entered, the participant presses the Enter button to proceed to the next PIN.

Regardless of the individual's typing behavior and familiarity (or lack thereof) with a given PIN or the PIN pad, we decided to randomize the order of PINs, rather than ask users to enter the same PIN multiple times. This approach generalizes the $\mathcal{P}$*inDrop* attack, which is actually applicable to both mnemonic PINs and One Time Passwords (OTPs).

We also collected the key logs of the PIN pad via the USB interface. In particular, for each pressed key, we collected both the "key-down" (press) and "key-up" (release) events. Moreover, we synchronized the recordings with the timestamp of these key events. All recordings were done with a sampling frequency of $44,100$Hz and then saved in the 32-bit WAV format.

4.3.2 Classification Methods

To identify the key pressed by the victim, we experimented with four well-known and popular classifiers: Support Vector Classification (SVC), k Nearest Neighbors (KNN), Random Forests (RF), and Logistic Regression (LR).

We applied a repeated nested crossfold validation to evaluate the performance of our approach. The pipeline varies on the number of attackers (i.e., a single attacker or a group) included in the training set.

In the outer loop, we randomly selected the attacker(s) among the participants in the dataset. This procedure was repeated 10 times generating 10 groups of attackers. The inner loop consists of a k-fold cross-validation, where k depends on the number of attackers. If the training set contains samples from a single attacker, we used 5-fold cross-validation, since a user-independent split is not applicable. If samples from at least two attackers are present in the training set, we use a k-fold cross-validation user-independent where k is the number of attackers.

We varied hyper-parameters by using the grid search on all four considered classifiers. For SVC, we considered a linear kernel and varied C among: $[10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}]$. For KNN, we varied the number of neighbors to among: [1, . . ., 20]. For RF, we considered from 10 to 100 estimators (steps of 10 and extremes included) and a max depth from 6 to 31 (steps of 5 and extremes included). Finally, LR was evaluated for ℓ_1 and ℓ_2 penalties, with C ranging from 10^{-4} to 10^{4}.

4.4 Experimental Results

We evaluate $\mathcal{P}inDrop$ in different scenarios, showing its performance in the different conditions in which the attacker may find himself. Section 4.4.1 describes how we evaluate different classifiers and consequently selected the best for our purpose. Sections 4.4.2 and 4.4.3 report the results for our algorithms on the key classification task and PIN classification task, respectively. Finally, Section 4.4.4 compares the performance of $\mathcal{P}inDrop$ with the results obtained in the state-of-the-art.

4.4.1 Model evaluation

To assess the performance of our classifiers, we evaluate different attack scenarios. In particular, we considered two settings: (i) number of distinct attackers and (ii) the number of digits entered by each attacker. We varied the number of attackers included in the training set between 1 and 10. This range has been selected to reflect a realistic attack scenarios. We varied the number of digits entered by each attacker in increments of 100, i.e., 100, 200, 300, 400, or 500.

The performance of our attack was evaluated on all possible combinations between the number of attackers and the number of digits entered by each attacker.

To select the best classifier, we compared the PINs validation accuracy of all the classifiers across different scenarios (i.e., PIN pads, and distances) and settings (i.e., number of digits per attacker, and number of attackers). SVC and LR achieved comparable performance, outperforming KNN and RF. In particular, LR achieved higher validation accuracy on *PAD-1*, while SVC showed better performance on *PAD-2*. In Table 4.1, we report a comparison of the validation accuracies for all the investigated classifiers, considering five attackers that train the classifiers with 500 digits each (i.e., training size = 2500 digits).

	PAD-1		***PAD-2***	
	Distance 0.3 m	**Distance 2 m**	**Distance 0.3 m**	**Distance 2 m**
SVC	0.90±0.04	0.35±0.12	**0.86±0.06**	**0.21±0.07**
LR	**0.92±0.04**	**0.40±0.11**	0.85±0.06	0.19±0.04
KNN	0.65±0.07	0.13±0.07	0.17±0.05	0.02±0.01
RF	0.78±0.07	0.10±0.06	0.31±0.06	0.02±0.00

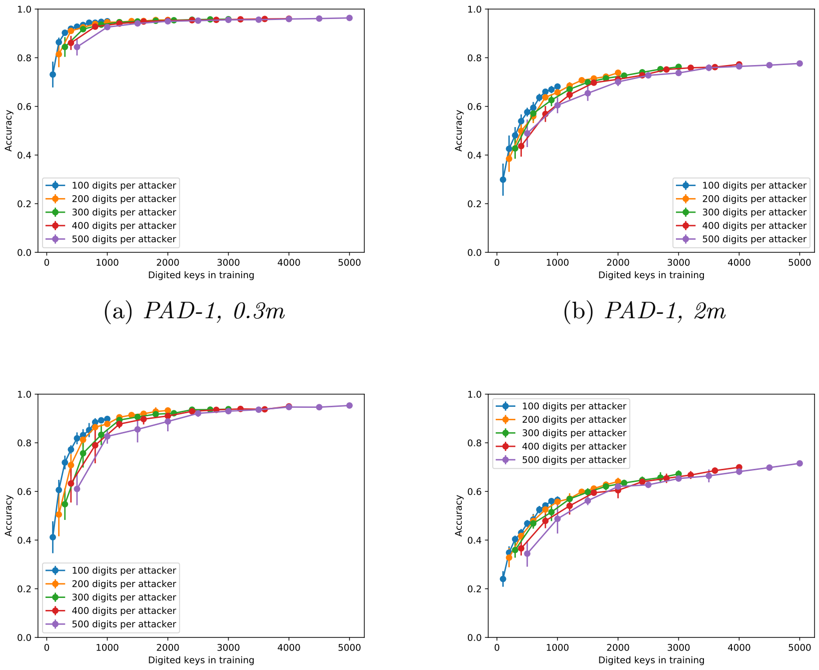
Table 4.1: PIN accuracies on the validation set for the investigated classifiers. The training set includes samples from five distinct attackers. The results show that for *PAD-1* the best performing model is the Logistic Regression (LR), while for *PAD-2* the best model is the SVC.

4.4.2 Single Key Inference

We report the LR classifier performance for the *PAD-1* and the SVC classifier performance for the *PAD-2* based on the validation results. In Figure 4.4 we show single key accuracy comparison for all the considered settings (i.e., the number of attackers and the number of digits entered by each attacker) in our four scenarios. Each graphic depicts how the accuracy varies in the considered scenario as the number of entered keys included in the training set varies. Further, each graphic shows five curves representing the number of digits entered by the attackers, while the bullets of a curve represent the number of attackers included in the training set. The bullets have an increasing value from left to right: the first bullet (from left) of each curve indicates that only one attacker has been included in training, the second indicates two attackers were included in training, and so on. Therefore, the number of numeric keys included in the training set varies according to the number of attackers and the number of digits entered by each attacker.

We note that the accuracy is significantly affected by the training set's size (i.e., entered keys in training) and the distance.

Interestingly, with the same number of entered keys in training, the accuracy improves due to the number of attackers. For example, if we set the number of entered keys in training at 400, we can see that in all scenarios, the accuracy obtained by four attackers typing 100 keys each (i.e., 20 5-digit PINs per attacker) is significantly higher than a single attacker typing 400 keys (i.e., 80 5-digits PINs). This may depend on the variability of the data used to train the classifiers. Each person has a slightly different typing style [200] (e.g., pressure strength, typing speed), and adding more attackers would introduce higher variance in the training set and helps our classifiers to generalize and improve their classification performance over a test set.



(a) *PAD-1, 0.3m*

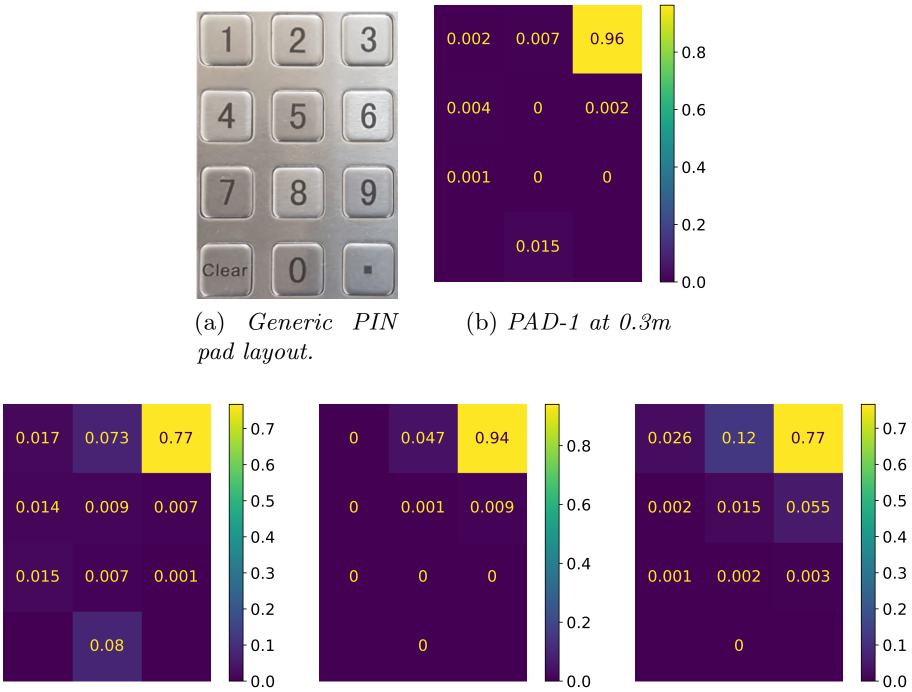
(b) *PAD-1, 2m*

(c) *PAD-2, 0.3m*

(d) *PAD-2, 2m*

Figure 4.4: Key accuracy on the testing set for the best classifiers.

Furthermore, we analyzed how our classifiers mis-classify the true key to investigate how spatial locality interferes in the classifiers' predictions. In



(a) *Generic PIN pad layout.*

(b) *PAD-1 at 0.3m*

(c) *PAD-1 at 2m*

(d) *PAD-2 at 0.3m*

(e) *PAD-2 at 2m*

Figure 4.5: Digit "3" prediction heat maps for the four considered attack scenarios. We reported the results for the experiment with 5 attackers and 500 digits entered per attacker.

Figure 4.5, we report an example for the digit "3" for all the four scenarios (a similar behavior is shown by all the other keys).

Interestingly, we note a different distribution of classification errors between *PAD-1* and *PAD-2*. In the first case, the error is uniformly distributed over all digits, whereas in the second case, a higher concentration of errors is prominent around the true digit (i.e., digits 2, 5, and 6).

4.4.3 PIN inference

In a realistic context, an attacker generally has three attempts to guess the victim's PIN (i.e., the max number of incorrect PIN entries allowed before blocking the card). In this section, we report on the performance of our approach in PIN reconstruction in TOP 3-accuracy, i.e., only the three most probable PIN predictions. In Figure 4.6 we show the performance of the classifiers in the reconstruction of 4-digit and 5-digit PINs according to the different settings (i.e., PIN pad and distances). Further, similar to Figure 4.4, each graphic reports the performance for all possible combinations of the settings.

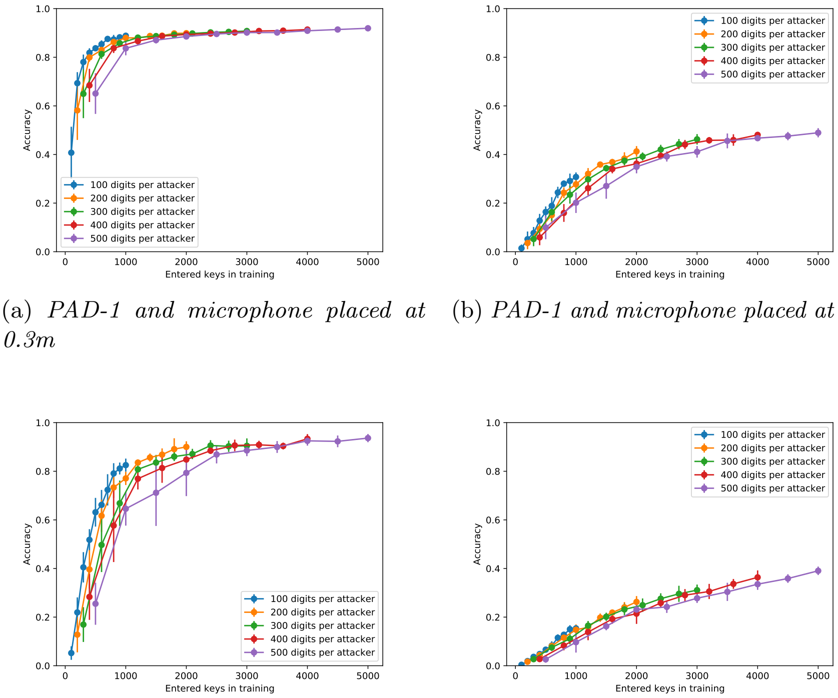
The results show that the effectiveness of the attack in each scenario. In particular, at 0.3m away, we can reconstruct correctly within three attempts up to 94% 4-digit PINs for *PAD-1* and up to 96% PINs for *PAD-2*. Although the performance worsens by increasing the distance at which the microphone is placed, $\mathcal{P}inDrop$ manages to reconstruct within three attempts up to 57% of the 4-digit PINs for *PAD-1* and up to 50% for *PAD-2* at 2m away. At 0.3m, the accuracy graphs reach a plateau at around 1500 digits in training. On the contrary, at 2m, the accuracy seems not to reach the plateau even with a training of 10 attackers and 500 digits per attacker (i.e., 5000 digits in training). This behavior is particularly marked in *PAD-2*, where the increase appears almost linear also with a high number of digits in training. This could be partially due to the classifier used in the specific scenario (i.e., LR for *PAD-1* and SVC for *PAD-2*) in addition to the physical differences between the two PIN pads.

Comparing the performance on two PIN pads (fixing the number of attackers and entered keys per attacker), the accuracy on *PAD-1* appears generally higher than the one on *PAD-2*. This applies to both distances. The number of attackers significantly affects performance with the same number of entered keys in training. For example, in *PAD-1* at 0.3m, the threshold of 80% of 4-digit PINs reconstructed in three attempts is reached with three attackers whom enter 100 digits each (i.e., 300 total digits), or two attackers whom enter at least 200 digits each (i.e., at least 400 total digits).

4.4.4 Comparison with the state-of-the-art

To evaluate $\mathcal{P}inDrop$, we compare its with that of state-of-the-art attacks exploiting acoustic emanations of PIN pads [42, 167, 200, 72]. Table 4.2 summarizes the results (with 10 attackers entering 500 digit each) in terms of key accuracy and PIN reconstruction accuracy within three attempts.

Both [167] and [42], exploit inter-keystroke timing. Although in [167] the distance at which the acoustic information is collected is unspecified, such attacks can be carried out from a distance over one meter, as demonstrated in [42]. The distance significantly decreases the risk of the attacker being detected. However, the reported performance is rather poor, since the PINs correctly reconstructed within three attempts were less than 1% for both attacks. However, from a greater distance (i.e., 2m) $\mathcal{P}inDrop$ outperform [42, 167] achieving the accuracy of 44% and 54% on 5-digit and 4-digit PINs, respectively.



(a) *PAD-1 and microphone placed at 0.3m*

(b) *PAD-1 and microphone placed at 2m*

(c) *PAD-2 and microphone placed at 0.3m*

(d) *PAD-2 and microphone placed at 2m*

Figure 4.6: 5-digit PINs inference performance within 3 attempts for the best classifiers.

Most effective attacks are those carried from a significantly shorter distance. In particular, [200] records acoustic emanations with a microphone placed at 0.05m from the PIN pad. This work obtains 60% key accuracy on a sub-set of keys (i.e., 6 on 10). Since we can not estimate the real accuracy considering all the 10 digits we decided for fairness, to leave this upper-bound. Under this assumption, we derived that this attack may achieve 4-digit and 5-digit PIN accuracies of 27.36% and 16.42%, respectively. Comparing these results with the performance of $\mathcal{P}inDrop$, we can see how $\mathcal{P}inDrop$ achieves better accuracy for both 0.3m and 2m.

The last method we consider was proposed by De Souza [72]. This attack assumes that two microphones are placed inside a PoS under the PIN pad. Unlike other methods, it uses the time of arrival of the acoustic signals. The performance achieved by the De Souza is slightly better to $\mathcal{P}inDrop$ from 2m. However, $\mathcal{P}inDrop$ has better performance from 0.3m (i.e., a 26% increase in 4-digit PINs and a 33% increase in 5-digit PINs). Moreover, $\mathcal{P}inDrop$ differs from [72] in that it does not require physical tampering with the device, even if the attack is performed from 0.3m away.

	Key Accuracy	4-digit PINs	5-digit PINs	Recording Distance
Liu [167]	NA	0.26% *	0.11% *	NA
Cardaioli [42]	NA	0.72%	NA	1.50m
Panda [200]	60.00%	27.36% **	16.42% **	$\sim 0.05m$
De Souza [72]	87.60%	68.40% **	59.92% **	0.00m***
PinDrop	95.84%	94.64%	92.79%	0.30m
PinDrop	74.58%	53.75%	43.99%	2.00m

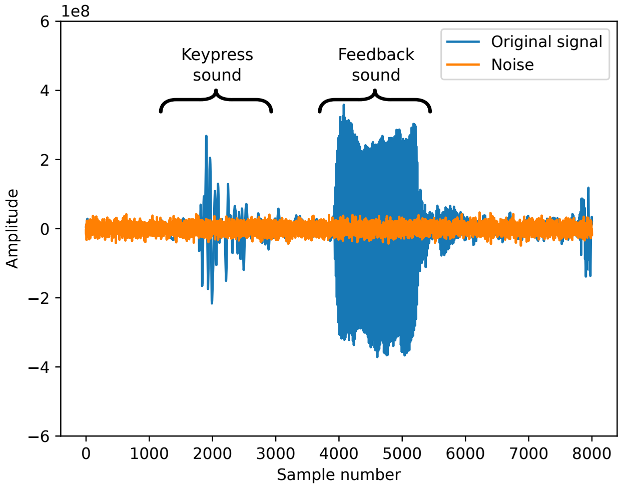
* Performance derived from the proportion of human-chosen PINs and the accuracy of each PIN strength level reported in the paper.
** Performance estimated from reported key accuracy, assuming the prediction error to be equally distributed.
*** Multiple microphones are integrated in the device.

Table 4.2: Comparison between *PinDrop* and the state-of-the-art results on single key accuracy and percentage of guessed PINs within three attempts. If the score cannot be derived from the reference paper, we report N/A.

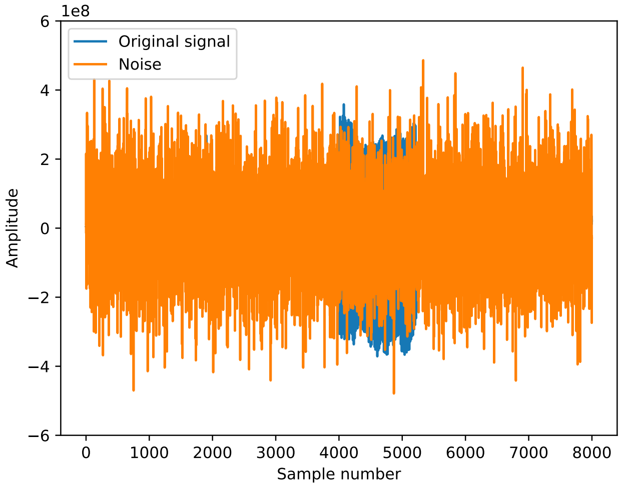
4.5 Impact of Noise on *PinDrop*

In Section 7.6 we demonstrated the effectiveness of *PinDrop* in a noise-controlled environment. This scenario can be traced back to ATM rooms commonly found in banks or city centers. To evaluate the effectiveness of *PinDrop* in other contexts (e.g., external ATMs), we simulated two different noise sources: i) road noise produced by urban traffic and ii) Gaussian noise. We modulated the two sources to obtain four levels of SNRs (Signal to Noise Ratios): very low noise (SNR 10dB), low noise (SNR 5dB), high noise (SNR -5dB), and very high noise (SNR -10dB). In Figure 4.7, we show the comparison between the audio emitted the sound emitted by a key press (with the corresponding feedback sound) and two amplitude levels of the modulated Gaussian noisy signal. Following the procedure described in Section 4.3, for each considered SNR, we trained and tested *PinDrop* with the perturbed signals obtained from the sum of the original signal with the corresponding modulated noise.

To simulate the noise produced by urban traffic, we extracted a set of urban noises from the *AudioSet* [240] dataset made available by Google. Accordingly to the four considered SNRs levels, we modulated the urban noises, and we added them to the original signal. In particular, 99% of



(a) *SNR 10 dB*



(b) *SNR -10 dB*

Figure 4.7: Comparison between very-low and very high levels of Gaussian noise with the original sound signal of a keypress.

the power of the considered set of urban noises ranges between 125Hz and 2500Hz, in line with the literature [14, 215].

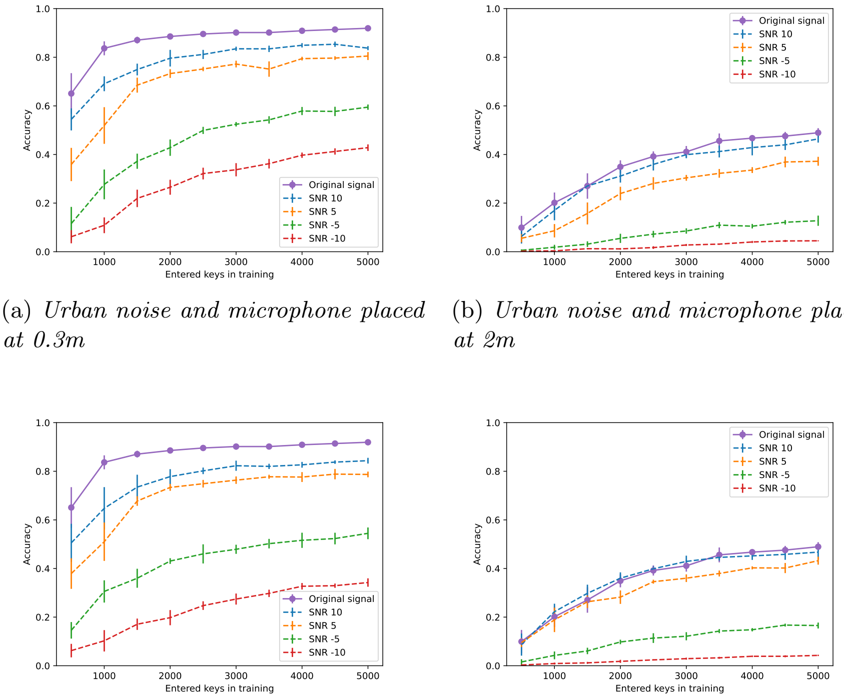
Similarly, to evaluate whether the addition of a noise that covers all frequencies affects the performance of $\mathcal{P}inDrop$, we perturbed the original signal with four modulated Gaussian noises amplitude, according to the four SNRs considered.

Figure 4.8 shows the results of $\mathcal{P}inDrop$ trained on the perturbed PAD-1 dataset (configuration 500 digits per attacker) in inferring 5-digit PINs within three attempts. From the graphs, it emerges that both at 0.3m and at 2m distance regardless the source of noise, $\mathcal{P}inDrop$ remains very effective when low noisy signals are added (i.e., SNR 10dB and 5dB). Further, Figure 4.8 highlights how the addition of low noises has a greater impact on the performance of $\mathcal{P}inDrop$ at 0.3m than at 0.2m. This difference in performance can be related to the more significant background noise component already present in the original signal recorded at 2m, making the algorithm more robust at low perturbation levels.

For higher noise levels (i.e., SNR -5dB and -10dB), $\mathcal{P}inDrop$ still manages to reconstruct a significant percentage of PINs when the attack is performed from 0.3m (e.g., up to 59% with SNR -5dB and up to 43% with SNR -10dB). However, the performance obtained at 0.3m by $\mathcal{P}inDrop$ on sounds perturbed by Gaussian noise are slightly lower than those obtained with urban traffic perturbation. This difference can be reconducted to the range of frequencies perturbed by the two sources of noise: Gaussian noise affects the entire spectrum, while urban noise has a limited frequency band. At 2m, the performance of $\mathcal{P}inDrop$ degrades significantly with high-noisy perturbation, suggesting that the information contained in the original signal is no longer sufficient to make the attack effective in a very noisy environment. In contrast to the attack scenario at 0.3m, at a distance of 2m we do not notice significant differences between accuracies of PINs reconstructed from audio perturbed with urban noise and those reconstructed from audio perturbed with Gaussian noise. This suggests that the high-frequency component (i.e., above 2500Hz) is less effective in the reconstruction of the PINs at 2m compared to 0.3m scenario.

4.6 Potential Countermeasures & Future Work

The relatively high accuracy of $\mathcal{P}inDrop$ highlights its danger and the importance of robust countermeasures. Barring wholesale replacement of PINs with other login means, we consider the following possibilities:



(a) *Urban noise and microphone placed at 0.3m*

(b) *Urban noise and microphone placed at 2m*

(c) *Gaussian noise and microphone placed at 0.3m*

(d) *Gaussian noise and microphone placed at 2m*

Figure 4.8: Impact of noise source and SNR in the inference of 5-digit PINs within three attempts for PAD-1 and 500 digits per attacker.

- *PIN Pad noise reduction*: This idea is simple, though challenging to deploy. It consists of masking the noise emitted by the PIN pad by covering it with soundproofing material. This approach could help in reducing the effectiveness of longer-range attack.
- *Noise emanation*: This countermeasure involves the emission of white noise by the ATM when entering the PIN. As shown in Section 4.5, high noise levels negatively affect attack performance.
- *On-screen PIN pad*: An effective countermeasure could be to virtualize the PIN pad using a touch screen. (This is in fact already done on some ATMs). This countermeasure would also allow dynamic rearrangement of digits, making it much more challenging to implement $\mathcal{P}inDrop$-like attacks. On the other hand, on-screen keypads are generally less user-friendly and can pose a problem for visually impaired users;
- *Feedback distortion*: If removing the characteristic sound emitted by each key is not possible, an alternative is to add noise that does not

allow individual keys to be profiled. By emitting a masking sound at each key-press, *PinDrop* can be made more difficult, especially, its training phase;

- *Personal PIN pad*: Another possible countermeasure is to use a trusted device, such as a smartphone, to replace the physical PIN pad. The PIN could then be transmitted to the ATM using a wireless medium (e.g., NFC);
- *Behavioral biometrics layer*: An additional layer of security might be possibly via behavioral biometrics. One possibility is to involve user authentication based on keystroke dynamics. While this method can yield a high rate of false positives, it is completely transparent to the user (until or unless, a false positive occurs).

Possible future directions range from improving applicability of *PinDrop* to exploring its effectiveness on other kinds of PIN pads. An interesting direction might be to apply more sophisticated (e.g., parabolic) microphones. Such a microphone could significantly extend the effective recording distance of *PinDrop*. Another direction is looking at *PinDrop* in the context of screen-based PIN pads that are fairly common on modern ATMs. This setting is more complicated due to lack of physical keys the sound of which can be profiled. However, it would be interesting to study whether sounds emitted by the touchscreen still allow the attacker to infer information about keys pressed.

4.7 Summary

This chapter demonstrated *PinDrop*, a highly accurate acoustic side-channel attack on PIN pads. It takes advantage of acoustic emanations produced by ATM users entering their PINs into the commodity ATM's metal PIN pads. These emanations can be surreptitiously recorded and used to accurately profile all PIN pad keys, allowing *PinDrop* to yield the victim's PIN with high probability. Specifically, this work shows that *PinDrop* is effective when applied from a very short (and perhaps not always realistic) distance away from the PIN pad (0.3m) as well as from a rather safe and inconspicuous distance (2m).

We demonstrated the effectiveness and robustness of *PinDrop* by conducting extensive experiments that involved a total of 58 participants and two commodities (commercially available) metal ATM PIN pads. We experimented with *PinDrop* in several configurations, showing how its performance can be optimized based on the training set size and the number of attackers.

$\mathcal{P}$*inDrop*'s accuracy reaches 93% and 95% in reconstructing 5-and 4-digit PINs, respectively, within three attempts, from 0.3 meters away. Also, at 2m away, $\mathcal{P}$*inDrop* outperforms state-of-the-art results, reaching over 44% accuracy. This translates into an average accuracy improvement of 44% and 53% in 5-digit and 4-digit PINs, respectively. Finally, we proved that $\mathcal{P}$*inDrop* is effective at 2 meters with low/moderate noise, reaching a lower-bound accuracy of 37%, while it remains effective under any noise condition at 0.3 meters. We believe that, due to its real-world applicability and performance, this work significantly advances the state-of-the-art in acoustic side-channel attacks.

Part II

Securing Computer-Human Interaction

Chapter 5

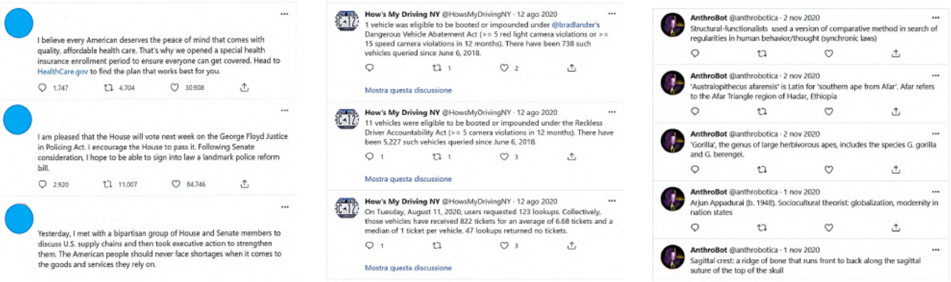
It's a Matter of Style: Detecting Social Bots through Writing Style Consistency

Social bots are algorithms acting like humans in social networks, able to share posts, load images, and interact with other profiles. Unlike other kinds of automated agents, such as web-crawlers or service bots, social bots are designed for imitating human behavior online [108]. Social bots cover a wide spectrum of types varying from bots that simply perform isolated actions of the communication process (e.g., liking or sharing) over partially human-steered accounts that may accomplish automatic tasks (the so-called hybrid bots, or "cyborgs" [47]), to agents enhanced with artificial intelligence and learning skills that may operate in a completely autonomous mode, like Microsoft's Zo or Replika.ai[1]. The usage of social bots for political manipulation and disinformation has been so considerable that they have inflated a huge debate [28, 96, 217], which led some governments to establish a proper regulation[2].

Since the detection of social bots remains a challenge [58], the actual number of social bots is not certain. Different estimations exist: according to Varol *et al.* [248] 9%–15% of active Twitter accounts should be social bots, while platforms themselves claim a magnitude of millions of accounts [219].

[1] https://www.zo.ai/, accessed: September 2020

[2] For example, the implementation of SB-1001 in California in July 2019, see https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180SB1001, accessed: September, 2020.



I believe every American deserves the peace of mind that comes with quality, affordable health care. That's why we opened a special health insurance enrollment period to ensure everyone can get covered. Head to HealthCare.gov to find the plan that works best for you.

I am pleased that the House will vote next week on the George Floyd Justice in Policing Act. I encourage the House to pass it. Following Senate consideration, I hope to be able to sign into law a landmark police reform bill.

Yesterday, I met with a bipartisan group of House and Senate members to discuss U.S. supply chains and then took executive action to strengthen them. The American people should never face shortages when it comes to the goods and services they rely on.

(a) *Genuine account*

How's My Driving NY @HowsMyDrivingNY · 12 ago 2020
1 vehicle was eligible to be booted or impounded under @bradlander's Dangerous Vehicle Abatement Act (>= 5 red light camera violations or >= 15 speed camera violations in 12 months). There have been 738 such vehicles queried since June 6, 2018.

How's My Driving NY @HowsMyDrivingNY · 12 ago 2020
11 vehicles were eligible to be booted or impounded under the Reckless Driver Accountability Act (>= 5 camera violations in 12 months). There have been 5,227 such vehicles queried since June 6, 2018.

How's My Driving NY @HowsMyDrivingNY · 12 ago 2020
On Tuesday, August 11, 2020, users requested 123 lookups. Collectively, those vehicles have received 822 tickets for an average of 6.68 tickets and a median of 1 ticket per vehicle. 47 lookups returned no tickets.

(b) *Twitter bot about traffic violations*

AnthroBot @anthrobotica · 2 nov 2020
Structural-functionalists used a version of comparative method in search of regularities in human behavior/thought (synchronic laws)

AnthroBot @anthrobotica · 2 nov 2020
'Australopithecus afarensis' is Latin for 'southern ape from Afar'. Afar refers to the Afar Triangle region of Hadar, Ethiopia

AnthroBot @anthrobotica · 2 nov 2020
'Gorilla', the genus of large herbivorous apes, includes the species G. gorilla and G. berengei.

AnthroBot @anthrobotica · 1 nov 2020
Arjun Appadurai (b. 1948). Sociocultural theorist: globalization, modernity in nation states

AnthroBot @anthrobotica · 1 nov 2020
Sagittal crest: a ridge of bone that runs front to back along the sagittal suture of the top of the skull

(c) *Twitter bot about anthropology*

Figure 5.1: Tweets from genuine and bot accounts

Both indications should be considered with caution since the tools used for detecting bots have been found to be not enough accurate [61, 187]. Given the volume of accounts and tweets, automated methods for detecting bots are needed. It has become increasingly difficult for a human to discern between legitimate users and bot-driven accounts [248].

This work aims to use the consistency of posts' style over time for detecting social bots, hypothesizing that bots are expected to keep a certain regularity in the posts' style, contrary to humans. While the posts of a bot are produced by an algorithm that composes the sentences according to fairly deterministic processes, when humans write are influenced by many factors that may alter their style, like rush, anger, boredom, excitement, or weariness. Specifically, our approach proposes to model a Twitter user through the mean and standard deviation values of a set of stylistic features computed on the tweets posted by the user. In particular, mean and standard deviation values aim to capture the central tendency and the width of fluctuations in the different users' style traits. Previous work [9] showed that stylometry-based features could help discriminate bot profiles from human ones. However, while previous work found that humans and bots tend to make different usages of character-based and emotions-based features, our study (i) mainly focuses on the writing style consistency of bot-driven Twitter accounts compared to human-operated ones, and (ii) demonstrates that bot-driven accounts exhibit a lower variability in specific writing characteristics.

An example of the substantial differences existing in the posts' styles is reported in Figure 5.1, where tweets from a genuine account (Figure 5.1a) and from two different bots (Figure 5.1b and 5.1c) are shown. The bot in Figure 5.1b was created by Brian Howland to put attention on the fact that there are many people driving and parking dangerously. The content of the tweets always has the same reporting structure. In particular, we observe numbers, statistics, and recurring propositions distinctive of an account exhibiting bot-like characteristics. The tweets in Figure 5.1c are posted by a bot that is a generator of metaphors, analogies, and similes about anthropology; the figurative outputs exhibit recurrent characteristics. For instance, the bot tends to ration words, to favor brevity and suggestiveness over verbosity and detailed exposition.

Computer-assisted techniques for characterizing and recognizing an author's style are developed and investigated by *stylometry* [196]. Research in stylometry has evolved during the 21st Century, expanding the spectrum of interest from plagiarism detection to fine arts. Further goals of stylometry envisaged unmasking identity deception in social media applications, identifying email impersonation, multi-modal authentication on mobile devices, attributing SMS messages to the original author and recognizing speech writers [101, 123, 159, 208, 237]. Stylometry is widely applied to detect style similarities of authors, which is the main issue of this work: to evaluate to which extent the style of an author, human or bot, remains alike over time.

Contribution. In summary, the main contributions of this chapter are the following:

- We propose a novel approach for evaluating the stylistic consistency of social network posts that could be used to accomplish other kinds of analysis based on authors' style, like psychological traits extraction, anger detection, identity theft;
- We identify the features capturing the stylistic consistency of posts which can distinguish when are realized by humans or bots with statistical evidence;
- We train a set of machine learning detectors built on top of stylistic consistency features that are able to identify human-operated and bot-driven Twitter accounts with high effectiveness (i.e., F-measure scores up to 98%).

This work is intended to fulfill two needs claimed by the current literature on bot detection: a larger employment of *"natural language processing techniques to detect automated or repeated content"* [198] and the identification of features complementary to the behavioral ones that are vulnerable to adversarial attacks [62]. Additionally, compared to the methods proposed so far [198], our approach offers the following advantages: (i) it leverages machine learning (ML) algorithms (i.e., more scalable and less expensive than deep learning-based approaches), achieving classification performance that is in line with the state-of-the-art techniques, (ii) it is exclusively based on the written contents posted by users (i.e., the features required to feed the ML algorithms are easy to collect and compute), and (iii) it is highly portable (i.e., it can be applied "as-is" to other platforms in which further social network features may not be available or easily accessible) and extensible (i.e., it can be used in conjunction with state-of-the-art approaches leveraging other features).

5.1 Related Work

Several methods have been developed to build social bot detectors and classifiers. Most of these strategies leverage supervised machine learning algorithms to highlight the hidden patterns characterizing automated behaviors (see Section 5.1.1). Other approaches are based on the topology of social networks, such as graph-based inspections and structure-based information (see Section 5.1.2). And finally, human-based approaches are proposed for inspecting user posts and profile analysis (see Section 5.1.3).

5.1.1 Feature-based approaches

The use of Artificial Intelligence (AI) in this context is mainly based on the assumption that bots and humans are clearly separable and that each machine actor has individual features that make it distinguishable from human ones. Chu *et al.* [48] used entropy measures for characterizing the differences between bots and humans on Twitter in terms of tweet content, tweeting behavior, and account properties. They have determined that humans have high entropy, i.e., complex timing behavior, whereas bots and cyborgs have a low entropy, i.e., regular or periodic timing. SentiBot [76], is a sentiment-aware architecture that considers four classes of features related to tweet syntax, tweet semantics, user behavior as well as network-centric user properties and uses an ensemble of six classifiers (Naive Bayes, SVMs, AdaBoost, Gradient Boosting, Random Forests, and Extremely Randomized Trees).

Igawa *et al.* [131] relied on pattern recognition in posts to distinguish bot from human-operated accounts, while Bara *et al.* [20] search for similar patterns in the posts produced by the same author for identifying bot. Since both the techniques are close to the one proposed here, there is a significant difference: our method relies on the evaluation of style through the stylometric metric bench, while those other methods consider patterns of words, the entity in the text (like URLs and emoticons) and a reduced set of punctuation. Cresci *et al.* [59] focused on the problem of fake followers, a type of automated accounts commonly used to increase the popularity of famous people's accounts. Starting from a manually constructed dataset of human accounts, comparing two types of classifiers, one based on rules and another based on features, they showed that black-box and feature-based classifiers perform better at spotting fake followers than white-box, rule-based classifiers. Davis *et al.* [68] proposed BotOrNot, the first social bot detection framework publicly available for Twitter. The tool applied a random forest approach, based on more than 1,000 features among network, user, friends, temporal, content, and sentiment features, with an 86% success rate. Other features-based studies took BotOrNot as a baseline, as the work on Bot-Hunter [27]. The authors propose a tiered approach to bot detection: single tweet text (tier 0), account and one tweet (1), account and full timeline (2), and account, timeline, and friends timelines (3).

Kabakus and Kara [142] examined the systems used for detecting spambots, which have been classified into four techniques. The first one is account-based and measures the account's properties like the number of tweets, the number of followers, and likes. Tweet-based spam detection considers parts of a tweet such as mentions, hashtags, the number of likes the tweet received, the number of retweets the tweet received, and the tweet's content lexical analysis of the tweet, the URL of the tweet, and so on. Graph-based spam detection analyzes the relationships between the sender and the mentions of a tweet, such as connectivity and distance among accounts. It evaluates the strengths of their connections to reveal the possibility of a spam connection. Finally, there are hybrid approaches that use a combination of the other three. According to the authors' analysis, tweet, account, and graph-based approaches can reach 99% of accuracy. Contrary to these techniques specific for detecting spambots, our approach is designed to be applied to any kind of social bots.

5.1.2 Network-based approaches

The network-based detection techniques focus on detecting Sybil accounts, i.e., fake accounts, which are mainly used to forge other users' identities and disseminate misinformation and malware. There are many works that solve the Sybil detection problem by using structures of the network. Cao *et al.* [40] developed SybilRank to detect Sybil accounts (bots) in social networks that analyze the social graph to compute Sybil-likelihood scores. These scores can facilitate the manual verification of such users and the eventual countermeasures. The work examines the social graph of Facebook and highlights how feature-based approaches with machine learning models failed to be effective for social network intrusion detection. SybilInfer [66] is an algorithm that recognizes Sybil attacks using a combination of Bayesian inference and Monte-Carlo sampling techniques to estimate the set of honest and Sybil users. The bounds provided by SybilInfer depend on the number of colluding entities in the social network and not on the number of trust relations between an honest and a dishonest node. SybilGuard [268] adopts the assumption that malicious users can create many Sybils. Still, the Sybils can have few connections to honest accounts, and nodes do not require the complete network topology knowledge, unlike SybilInfer.

5.1.3 Crowdsourcing-based approaches

Most of the aforementioned approaches rely on a partial manual labeling process to assess the automated classifiers' value. In the work by Cao *et al.* described in [40], human annotation is defined as ineffective when it comes to large-scale evaluations due to the effort required for a single classification. In [61], the authors assessed the human performance in discriminating between genuine accounts, social spambots, and traditional spambots. They proved that crowdsourcing annotators were not able to distinguish new waves of spambots from legitimate users. However, in the detection of traditional social bots, human annotation and human-based methods are often used as ground-truth, as it happens in the work by Chu *et al.* [48]. Among the collected data, they randomly chose different samples and classified them by manually checking user logs and homepages of humans, bots, and cyborgs. Authors of [248] proposed a bottom-up approach for the identification of bots with similar online behavior and the classifier adopted is the one adopted by BotOrNot, while the dataset used also included a manually annotated collection of Twitter accounts. The PAN 2019 [209] evaluation campaign was aimed at author profiling, held each year since 2013 and 2019, and envisaged bots detection and gender profiling. The best results (accuracy score of about 92%) in bots detection using the English language have been obtained with a variety of stylistic features and Random Forest [139].

From the analysis of the related literature, it emerges that human-based approaches are expensive in terms of time and individuals involved. Graph-based methods fit well large-scale evaluations, using low computational complexities, and the feature-based methods seem to be more suitable to infer over bots and humans. For this reason, as mentioned before, our work rather focuses on a feature-based approach to distinguish bot-driven from human-operated accounts, relying on a metrics suite of stylometry that has not been yet largely investigated for social bot detection.

5.2 Study Design

The *goal* of this study is to verify whether the consistency of writing style is effective to discern human-operated from bot-driven Twitter accounts. The *context* consists of Twitter posts authored by both human- and bot-operated accounts, and it is detailed in Section 5.2.1. To validate our hypothesis, we pose two research questions:

- RQ_1: ***Do bots and humans exhibit different stylistic consistency in social network posts?*** This research question aims at evaluating whether the stylistic consistency of bots is greater than the one observed in human authors.
- RQ_2: ***To which extent are machine learning models based on stylistic indicators able to discern bot-driven from human-operated Twitter accounts?*** The purpose of this research question is to investigate the effectiveness of stylistic consistency indicators when used to train machine learning models aimed at predicting the nature (i.e., either bot- or human-operated) of Twitter accounts.

The writing style consistency of users is measured through a suite of metrics selected from the literature about stylometry, which will be introduced in Section 5.2.2, while in Section 5.2.3 we discuss the analyses performed to answer our research questions.

5.2.1 Context

The source of the dataset used in this research originates from [61]. It contains a collection of spambots, social spambots, fake followers, and genuine accounts. More specifically, the fake followers considered in our study are Twitter accounts created to inflate the number of followers of a target account and were bought from three different Twitter online markets [59], while the traditional spambot and social spambot accounts have been manually validated in previous work [61, 263].

As some of the accounts in the original dataset could have no tweets associated [199] and our approach relies on the stylistic consistency of tweets, we avoided considering accounts with no tweets associated. The dataset used in our work includes 12,179 among genuine and fake accounts (i.e., fake followers, traditional spambots, and social spambots) and more than 11.5 million tweets. The numbers of accounts and tweets considered in each category are reported in Table 5.1.

Table 5.1: Experimental dataset

Group name	**Description**	**Accounts**	**Tweets**
genuine accounts	human-operated accounts from [61]	3,211	7,896,356
fake followers	fake follower accounts from [59]	3,099	195,757
traditional spambots	spammer bot accounts from [263]	998	145,085
social spambots #1	retweeters of an Italian political candidate from [61]	989	1,610,171
social spambots #2	spammers of paid apps for mobile devices from [61]	3,420	427,890
social spambots #3	spammers of products on sale at Amazon.com from [61]	462	1,418,619
Overall		**12,179**	**11,693,878**

5.2.2 Metrics Suite

The metrics suite has been built by selecting the indicators used in stylometry's literature for measuring four properties of a text since they fit better than the others the purpose of our study: the structural traits, the semantic traits, the lexical traits, and the readability.

The structural analysis assumes that the text is simply a sequence of characters: this assumption allows to perform different analyses. The analyst may count: the number of characters in the text, punctuation signs or only the upper- or lowercase, the frequency of a specific letter, or n-grams, which are language-independent, differently than words. Furthermore, the sampling in character sequences allows obtaining a dataset richer than the one based on words for short texts. Some of these structural properties showed to be effective in the authorship attribution [107].

The automatic computation of data concerning a text's semantics has not yet reached the same levels as other fields of linguistics, like phonology, lexical, and syntactic. There are several attempts to make quantitative the semantic data, like in Gamon *et al.* [104] or in Argamon *et al.* [8] which maps a functional trait with a specific word. A similar mechanism is used to perform Sentiment Analysis, which associates a word with a label depending on the feeling it transmits.

Even the lexical choices done in a text may indicate the historical period when the text was written, the provenience, and the author's education. Mistakes can lead to determine the identity of an author. The lexical traits may be measured by examining the vocabulary's richness or the number

Table 5.2: Metrics of Stylometry used in the Experiment

Metric	Formula	Description
Number of Total Characters	C	where C is the total number of characters in the text.
Number of Uppercase Characters	$\sum_{i=0}^{C} u(c_i)$	where $u(c_i)$ is 1 if the i-th character is an uppercase character and 0 otherwise.
Number of Lowercase Characters	$\sum_{i=0}^{C} l(c_i)$	where $l(c_i)$ is 1 if the i-th character is a lowercase character and 0 otherwise.
Number of Special Characters	$\sum_{i=0}^{C} s(c_i)$	where $s(c_i)$ is 1 if the i-th character is a special character and 0 otherwise.
Number of Numbers	$\sum_{i=0}^{C} n(c_i)$	where $n(c_i)$ is 1 if the i-th character is a number and 0 otherwise.
Number of Blanks	$\sum_{i=0}^{C} b(c_i)$	where $b(c_i)$ is 1 if the i-th character is a blanck character and 0 otherwise.
Number of Words	W	where W is total number of words in the text.
Average Length of Words	$\frac{1}{W}\sum_{i=0}^{W} len(w_i)$	where W is the number of words while $len(w_i)$ is the length of the i-th word.
Number of Propositions	P	where P is the total number of propositions in the text.
Average Length of Propositions	$\frac{1}{P}\sum_{i=0}^{P} len(p_i)$	where $len(p_i)$ is the length of the i-th proposition.
Number of Punctuation Characters	$\sum_{i=0}^{C} z(c_i)$	where $z(c_i)$ is 1 if the i-th character is a punctuation character and 0 otherwise.
Number of Lowercase Words	$\sum_{i=0}^{W} h(w_i)$	where $h(w_i)$ is 1 if the i-th word is a lowercase word and 0 otherwise.
Number of Uppercase Words	$\sum_{i=0}^{W} j(w_i)$	where $j(w_i)$ is 1 if the i-th word is an uppercase word and 0 otherwise.
Vocabulary Richness	$\frac{dw}{W}$	where dw is the length of the text without duplicated words.
Number of URLs	$\sum_{i=0}^{W} q(w_i)$	where $q(w_i)$ is 1 if the i-th word is a url and 0 otherwise.
Flesch Kincaid Grade Level	$0.39*(E)+11.8*(G)-15.59$	where G is the average number of syllable per word, while E is the average number of words per proposition.
Flesch Reading Ease	$206.835-(84.6*G)-(1.015*E)$	where G is the average number of syllable per word, while E is the average number of words per proposition.
Dale Chall Readability	$0.1579*(PDW)+0.0496*ASL$	where PDW is the percentage of difficult words (words that do not appear on a specially designed list of common words familiar to most 4th-grade students), while ASL is the average length of a proposition in words.
Automated Readability Index	$4.71*\frac{C}{W}+0.5*\frac{W}{P}-21.43$	where W is the number of words contained in the text, C is the number of the total amount of characters in the text, while P is the number of propositions in the text.
Coleman Liau Index	$0.0588*L-0.296*S-15.8$	where S is the average number of propositions per 100 words while L is the average number of letters per 100 words.
Gunning Fog	$0.4*(\frac{W}{P}+100*\frac{DW}{W})$	where W is the number of words contained in the text, DW is the number of words consisting of three or more syllables, while P is the number of propositions in the text.
SMOG (Simple Measure of Gobbledygook)	$1.0430*\sqrt{\frac{DW*30}{P}}+3.1291$	where DW is the number of words consisting of three or more syllables while P is the number of propositions in the text.
Linsear Write	l_w	For each word with two or less syllables an index is increased by 1, while for each word with more than three syllables, the index is increased by 3. Finally, the resulting number is divided by the number of propositions. If the result is greater than 20 it is divided by 2, otherwise it is divided by 2 and 1is subtracted from this number.

of words used in a sentence. This technique can be applied by leveraging a tokenizer, according to a bag of words model. Different values can be computed: the ratio between different words and the number of words, the number of hapaxes, which occurs only once in the text.

The metrics of readability capture the ease of understanding of a written text. The greater this feature, the easier the reader will distinguish letters and words. The readability is related to the speed and effort of reading, but it is specifically relevant for the readers with poor text comprehension abilities. Different metrics have been proposed for evaluating readability. The Flesch Kincaid Reading Ease uses the number of words and syllables while the Flesch-Kincaid grade level considers the same pair of parameters but with different weights [238]. The Dale-Chall [103], designed for quantifying the readability in books for children, assigns a score to a text. The greater the score, the higher is the level of education required to understand the text: it uses the length of the words and the percentage of difficult words. The Automated Readability Index [232] evaluates the readability utilizing the percentage of characters per word and words per sentence. The Coleman Liau Index [183] considers only the number of characters that make up the words, while the Gunning Fog [112] is an indicator of the formal number of years necessary to educate a person who can easily understand the text. SMOG (Simple Measure of Gobbledygook) [182] aims at expressing the same property but is more accurate and easier to compute than Gunning Fog. The Linsear Write formula [95] was developed by the US Air Force for computing the readability of their technical handbooks and relies on the length of sentences and the number of syllables contained in words. The list of metrics selected for the suite, the mathematical formulation, and the corresponding definition are provided in Table 5.2.

5.2.3 Analysis Method

To answer if bots and humans exhibit different stylistic consistency (RQ_1), we perform a statistical analysis on the extracted stylometric features. In particular, once computed every metric M (detailed in Table 5.2) for each post in our dataset, we group these values by the authors of the posts, U, obtaining the sets M_U. In order to characterize the style of a generic user U, for each set M_U, we compute the mean, A_{M_U}, and the standard deviation, SD_{M_U}. To assess whether the differences in the style consistency between the two types of users (i.e., bot-driven and human-operated) can be observed, for each considered metric M, we compare the distributions of A_{M_U} and SD_{M_U} values obtained for the bot-driven users with the distributions of A_{M_U}

and SD_{M_U} values obtained for the human-operated accounts. For comparing the pairs of distributions, we apply (i) the Mann-Whitney test [185] (with α fixed to 0.01), which is used when it is not assumed any specific distribution for two independent groups, and (ii) the Cliff's d effect-size measure [49], to quantify the amount of difference between groups. As recommended by the guidelines given in [109], we interpret the effect size as *small* for $|d| < 0.33$, *medium* for $0.33 \leq |d| < 0.474$, and *large* for $|d| \geq 0.474$.

To answer if ML models can discern between human and bot accounts (RQ_2), for every user in our dataset, we use the mean (A_{M_U}) and the standard deviation (SD_{M_U}) values of each metric M computed in the previous phase to (i) train five different machine learning (ML) classifiers, namely decision tree (DT), random forest (RF), logistic regression (LR), linear Support Vector Machine (SVM linear), and Support Vector Machine with rbf kernel (SVM rbf), and (ii) evaluate the extent to which such classifiers are able to distinguish between bot-driven and human-operated Twitter accounts. We chose these specific ML algorithms as they have been successfully employed in previous work concerning author profiling tasks [210]. In particular, we perform a stratified nested 10-fold cross-validation to split the dataset in training and test set, and a 10-fold inner cross-validation for hyper-parameters selection and model validation. In this configuration, each split of the nested-cross-fold validation consists of 80% of the dataset in training, 10% in validation, and 10% in testing. This approach allows us to estimate an unbiased generalization performance and evaluate both models and features robustness through different splits. We a-priori select a set of the hyper-parameters for each model to perform the model validation using grid search. In particular, for the DT, the depth varies between 1 and 4, and the CART learning algorithm is used. For the RF classifier, the depth varies between 1 and 4, and the number of estimators is selected in $\{20, 50, 100\}$. For LR C is set in $\{10^{-3}$, $10^{-2}, \ldots, 10^{1}\}$, and penalty in $\{L1, L2\}$. For SVM linear, C is set in $\{10^{-3}$, $10^{-2}, \ldots, 10^{1}\}$. Finally, for SVM rbf C is set in $\{10^{-3}$, $10^{-2}, \ldots, 10^{1}\}$ and γ in $\{10^{-3}$, $10^{-2}, \ldots, 10^{0}\}$. For LR, SVM linear, and SVM rbf, we normalize the data using a standard scaler. We evaluate the performance achieved by the different ML models through widely-known metrics in the information retrieval field: *Accuracy, Precision, Recall, and F-measure* [12]. As recommended by Demšar [74], we apply (i) the Friedman test [102], to investigate whether the differences observed in the performance (in terms of F-measure) achieved by the experimented classifiers are statistically significant, followed by (ii) an eventual post-hoc Nemenyi test [197], to identify the specific pairs whose differences exhibit statistical evidence.

5.3 Results

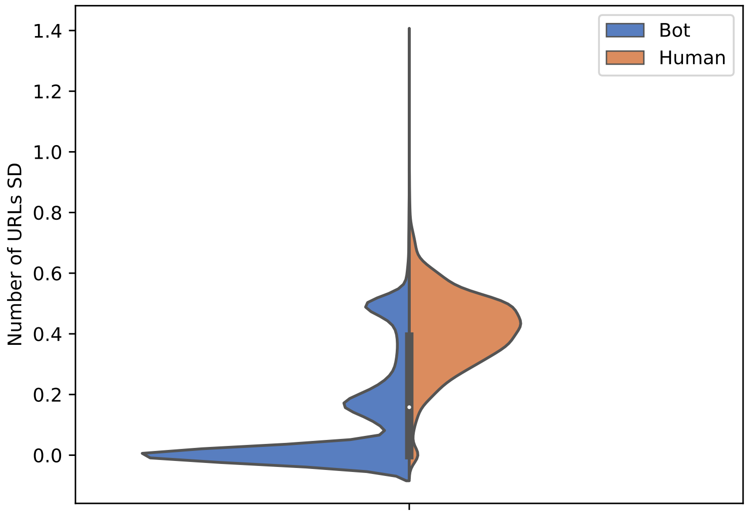
We assess the statistical differences in bot and human tweets' collected data applying the Mann-Whitney test and Cliff's delta. The Mann-Whitney test resulted in significant differences on all the features ($p < 0.001$).

Regarding the effect size, four features show large values of d, specifically: the *standard deviation of URLs* ($d = 0.747$), the *standard deviation of the number of numbers* ($d = 0.589$), the *average of URLs* ($d = 0.576$), and the *average of the number of numbers* ($d = 0.571$). Besides, five features exhibit medium effect size, namely the *average of the number of uppercase words* ($d = 0.396$), the *standard deviation of the number of punctuation characters* ($d = 0.369$), the *average of the number of uppercase characters* ($d = 0.367$), the *average of the number of propositions* ($d = 0.351$), and the *average of the number of punctuation characters* ($d = 0.337$).

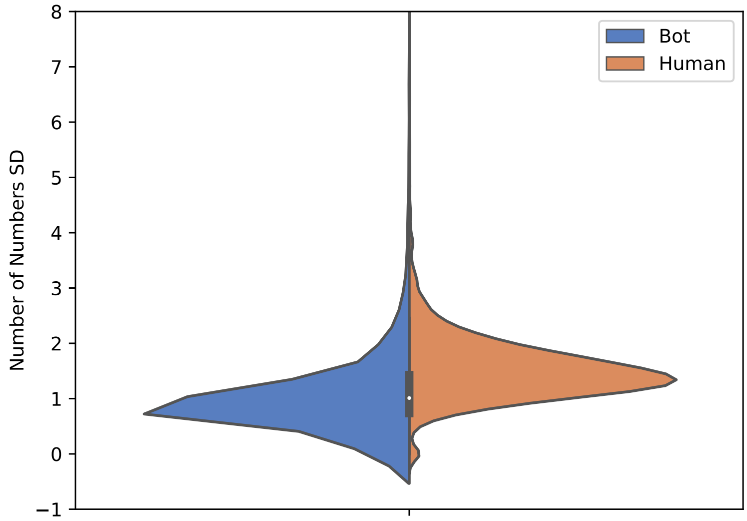
From the statistical analysis, it emerges that tweets generated by bot-driven accounts report a relatively stable consistency in including links and numbers. This behavior is evidenced in Figure 5.2, comparing the distributions between bots and humans for the two features with the largest effect size. In particular, both the *standard deviation of URLs* and the *standard deviation of the number of numbers* show higher average values for humans, confirming a greater writing consistency in bots accounts for these two features. On the other hand, we report higher usage of uppercase characters and uppercase words in human-generated tweets. While the consistency in the usage of links and numbers could depend on bots' primary goals (i.e., spreading contents), more recurrent uses of uppercase characters and words are likely connected with the emotional sphere of human users [151].

To assess whether the stylometric features provide adequate information to discern bot-driven from human-operated Twitter accounts, we evaluated different ML classifiers' performance. In the bar chart of Figure 5.3, for each considered metric and ML algorithm, we report the average (represented by the height of the colored bar) and the standard deviation (represented by the error bar) of the results obtained on each (cross-validation) split.

The results show that all classifiers obtain an F1 score above 0.95, confirming the robustness of the features extracted. The tree-based classification algorithms exhibit the worst results among those considered in this study, achieving F1 scores of 0.955 ± 0.003 and 0.959 ± 0.007 for DT and RF, respectively. The best performance for all metrics is achieved by the SVM rbf that reaches a F1 score of 0.982 ± 0.004. Moreover it is possible to notice how the variance of the results is generally limited for all the metrics, highlighting the robustness of the models in the different splits of the cross-validation.



(a) *Standard deviation of the number of URLs*



(b) *Standard deviation of the number of numbers*

Figure 5.2: Violin plots comparing the distribution of humans and bots for the two features with the largest effect size.

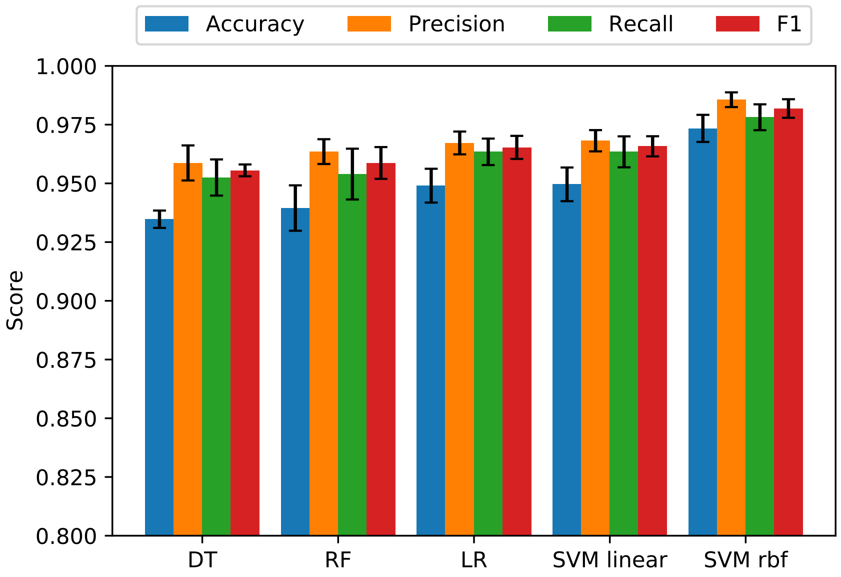


Figure 5.3: Classification metrics of decision tree (DT), random forest (RF), logistic regression (LR), linear Support Vector Machine (SVM linear) and Support Vector Machine with rbf kernel (SVM rbf) classifiers, in nested 10-fold cross-validation.

To assess whether the classifiers' differences are statistically significant, we apply Friedman's test on the F1 metric. Specifically, Friedman's test returns $p < 0.001$, suggesting that the classification results achieved by the different ML algorithms are not equivalent. In Figure 5.4 we report the critical difference (CD) diagram among the classifiers applying the Nemeny's test. Classifiers are ordered by average rank on all splits, specifically: $DT = 4.6$, $RF = 4.3$, $LR = 2.5$, $SVM\ linear = 2.6$, and $SVM\ rbf = 1.0$. According to the Nemenyi post-hoc analysis, the tree-based algorithms show significantly different performance compared to LR, SVM linear, and SVM rbf. On the contrary, these latter techniques do not present significant CDs between them. Although we observe differences in experimented models' performance, the F-measure in the classification of human accounts and bots is above 0.95 for all the investigated ML techniques. This suggests that the high effectiveness in discerning human-operated from bot-driven Twitter accounts is mostly related to the stylistic consistency features used in this study rather than the specific predictive model selected.

Cresci *et al.* [61] compared the detection performance of different state-of-the-art techniques on two test sets sampled from the same dataset we used in this study. Such a comparison highlighted that the best bot-detection performance is achieved by the methods proposed by Ahmed *et al.* [4]

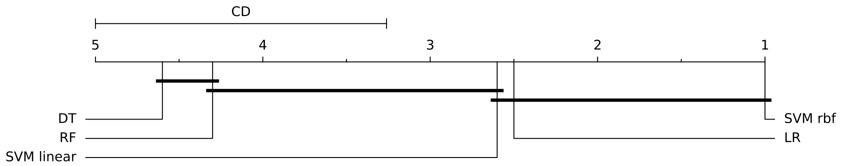


Figure 5.4: Critical difference (CD) diagram of the post-hoc Nemenyi test ($alpha = 0.10$). If the gap between the ranks of two algorithms is larger than CD, their difference is significant.

and Cresci *et al.* [60], that obtain F-measure values higher than 0.92 on both test sets. While the approach proposed by Ahmed *et al.* [4] leverages interaction-based, post-based, URL-based, and tag-based features, Cresci *et al.* [60] exploit DNA fingerprinting techniques based on the types of tweets (i.e., simple tweets, reply tweets, or re-tweets) posted. In terms of F1 score, our SVM rbf model obtains slightly better classification performance than both prior approaches. Besides, when using them on other social networking platforms, both previous methods need adjustments. Differently, our approach can be used *"as-is"* on other platforms since it only analyzes posts' contents.

5.4 Threats to Validity

Threats to construct validity are related to possible imprecision in measurements we performed. To carry out our study, we measure different factors that could not be sufficient to exhaustively model the writing style of a Twitter user. To partially mitigate this threat, we selected a set of well-known metrics previously used for tackling similar issues [170, 210, 275].

Threats to internal validity concern confounding factors that could affect our results. An important confounding factor could be related to possible inaccuracies in identifying actual human and Twitter users in our dataset. To alleviate this issue, in our study, we leverage publicly available datasets in which both human- and bot-operated accounts were manually verified [59, 61, 263].

Threats to conclusion validity concern the relationship between treatment and outcome. Appropriate statistical procedures have been adopted to draw our conclusions. Specifically, we used the Mann-Whitney U test for exploring the statistically significant differences occurring in the stylistic features which characterize the two groups of users (i.e., human-operated and bot-driven). The magnitude of the significant differences is then quantified by using Cliff's delta effect size measure. Besides, as recommended by Demšar [74], we used the Friedman test, followed by a post-hoc Nemenyi test, to investigate whether the differences in the performance achieved by the selected machine learning models were statistically significant.

Threats to external validity concern the generalizability of the findings. The leveraged dataset collects tweets posted by three main types of bot-driven accounts: (i) *fake followers*, (ii) *spambots*, and (iii) *social bots*. However, different types of bots (or bots operating on other platforms) could exhibit higher style inconsistency and different writing behaviors. Thus, further research to verify whether our findings generalize to other types of bots is needed. To cope with this matter, in the future, we plan to replicate our study at a larger scale, considering more heterogeneous bots.

5.5 Summary

In the last years, the number of bot-driven accounts has been extraordinarily growing in social networks. Recently, massive usage of social bots has been observed for spreading misinformation or conditioning electoral campaigns. As approaches for bots identification are not yet accurate enough, we proposed a method for detecting social bots based on the writing style's characterization. The underlying assumption is that a human author tends to change the writing style over time as it is influenced by external factors, while a bot should show a style that is mainly consistent over time, as it is produced by an algorithm whose result is fairly deterministic. For measuring the style of a post's author, we used a set of metrics defined by stylometry literature.

The experimentation carried out on 12,179 among bot-driven and human-operated Twitter accounts demonstrated that our conjecture was correct. Indeed, most of the selected metrics exhibited differences in style consistency over time between human-operated and bot-driven accounts. that are statistically significant ($p < 0.001$). ML classifiers were trained with the stylometry metrics for evaluating their ability to distinguish between a human author and a bot. All the ML algorithms achieved F-measure values above 0.95, while Support Vector Machine with rbf kernel showed the best performance, achieving an F1 score of 0.982 ± 0.004. We will apply this method for detecting identity theft, fake profiles, and profile masquerading in social networks in future work.

Chapter 6

Face the Truth: Detection of Spontaneous and Posed Emotional Facial Expressions

Facial expressions represent an innate and automatic behavioral component of emotional and social communication [67, 136, 192, 281]. Emotional facial expressions, in particular, have a communicatory function that conveys specific information to the observer [7, 67, 94, 134, 135]. For example, an expression of happiness through a smile in response to a particular behavior increases the probability that the action will be repeated in the future, differently from an angry or sad face [192]. In this sense, the nature and the interpersonal function of the emotional facial expressions conveys a message that predicts different social outcomes [67, 83]. It is precisely for this reason that accurately deciphering what someone is trying to communicate through facial expression, is extremely important in day-to-day social interactions [140]. Importantly, emotions conveyed by faces can change under several parameters. We can display different varieties of expressions: some intense and sustained, while others are subtle and fleeting [5]. One of the most high-level and critical communication features is related to the perception of authenticity of the emotion expressed [171, 216]. We can express emotions spontaneously, triggered by real circumstances (i.e., "event-elicited")[69]. For example, someone might be scared because he is genuinely afraid of a snake or be sad because of the loss of a loved one. Conversely, we can deliberately feign or pose emotions in the absence of a congruent underlying context to receive adaptive advantages. These expressions reflect the strategic intent of the sender in the absence of felt emotions [93]. For example, pretending to be sad can be a useful strategy to take advantage of a perceiver's reciprocal kindness or compensatory behavior in response [212]. The endogenous nature of emotional experiences (i.e., genuine or posed) completely changes the observer's perception and reaction. In social interactions, perceiving others' emotional reactions as genuine might promote social interaction and increase the expresser's trustworthiness [212].

Social media represents a field in which the sharing (conscious or not) of emotions is a major factor. Humans' nature to share emotions derives from different reasons such as obtaining help, care, or support, drawing attention, getting closer to someone, facilitating social interactions, and so on [213, 214]. In the last decades, the easy use of social media started to dig deeper and deeper down into society's brain stem, catalyzing the human tendency to widespread every aspect of their lives [36, 71, 250, 258, 259]. The current social media platforms promote emotional self-expression, inviting users to post their positive and negative emotional expressions online regularly [259]. TikTok, for example, is one of the fastest-growing social media platforms in the world, which allows users to share their personal content. According to the latest statistics, 689 million people are monthly active users. Among them, 55% upload their own videos displaying feelings, reactions, and emotions[1].

Social media platforms are also a theater where everyone may fake their feelings. In fact, many people on social media do not display genuine emotions for a number of reasons: increase or appease their followers, present idealistic self-representation, regulate their emotions by sharing their feelings, and so forth [13, 250]. Consequently, it is common to see fake (altered) facial expressions of emotions on social media. Many users may pose their emotional reactions, may hide their inner feelings, or overreact to scenarios they create through their social media profiles. It is in human nature to lie. Therefore, how can we distinguish spontaneous emotional reactions from posed ones?

It is well known how people are completely unable to recognize deceit in emotional displays, in particular, if they have to rely on visual cues only [23], without a real context of interaction (like in social media). Several studies demonstrated how people tend to perform not far from the chance level when asked to detect such behaviors [205, 206, 251]. The research community is tackling the problem by applying machine learning algorithms to discern between genuine and posed emotions. However, the models used so far yielded great variability among the results and a lack of robustness [114, 138].

[1] https://www.oberlo.in/blog/tiktok-statistics

Consequently, to date, there is still skepticism about the interpretability and real-world applications of the results obtained.

In this chapter, we want to evaluate how intra-individual variability in expressing posed or genuine emotions affects the performance of predictive models.

Contributions. The contributions of the current chapter are multiple:

- We developed a framework for the automatic detection of spontaneous and posed emotional facial expressions from clips. We applied the framework in two scenarios to classify the genuineness of emotional expressions ad hoc for each user (i.e., user-dependent scenario) and investigate the relevancy of inter-individual variability in the emotional lie detection (i.e., user-dependent vs user-independent scenario).
- We assessed the performance of our framework in genuineness discrimination through extensive experiments. Predictive models achieve an average accuracy of 84.4% in a user-dependent scenario and 67.0% in a user-independent scenario.
- We created a novel dataset (PEDFE) that includes a considerable amount of emotional clips (i.e., 1458) for both spontaneous and posed emotions. The same emotion is displayed genuinely and posed for each participant, allowing a direct comparison (i.e., intra-subject and between-subject) between these two ways to express the emotional facial expressions.
- To elicit the emotion as naturally as possible, we applied a novel protocol that uses a multimodal sensorial perception, avoiding any restrictions or influences by the researcher. To the best of our knowledge, the current emotion elicitation protocol has more tasks (i.e., 15) than the other reported methods (see Miolla et al. [188] for a review).
- We validated all stimuli through a survey by asking 122 participants to rate each clip according to the emotion, genuineness, and intensity of the facial expression perceived. It implies an essential step in creating emotional datasets that most of the datasets displaying genuine and posed emotions neglected (see Miolla et al. [188] for a review).

6.1 Related Work

So far, only a few studies applied machine learning to discriminate genuine/non-genuine emotions based on the dynamics of facial movements. The first attempt to automatically detect genuine from fake facial expressions was carried out by [23] where a system based on nonlinear SVM and AdaBoost for real-time recognition of facial actions was implemented. The authors trained 20 different classifiers (one per AU) and used the individual Gabor filters applied to the videos as features. The input of the SVM consists of 200 features per AU, 4000 in total, derived after a feature selection performed by the AdaBoost algorithm. In this work, the authors focused only on the recognition of true and posed pain facial expressions, reaching up to 72% of accuracy. Then, another deception detection analysis was proposed by the same authors [24], obtaining an accuracy of 85%. In this work, 50 genuine and fake facial expressions videos of pain were analyzed by using 20 facial actions extracted by the Computer Expression Recognition Toolbox (CERT) [22]. Each facial action was processed with temporal Gabor filters at eight different frequencies. The authors detected the zero-crossing for each frequency and calculated the resulting area under the curve and over the curve. The distribution of these measures was used as input to train a nonlinear SVM with a Gaussian kernel. As the previous one, also in this work, the authors focus only on pain expressions, not on emotion deception, using non-interpretable models.

Most advanced machine learning and computer vision analysis focused on the difference between the activation and the kinematics of the muscle movements, also called Action Units (AUs) [91], in spontaneous and posed facial expressions (please see [138] for a review). This method originates from Ekman's theories [92, 89], which identified six basic emotions characterized by a specific facial configuration in their display: happiness, sadness, anger, disgust, surprise, and fear. Previous research's aim was to identify the keystone about the emotional lie detection in facial displays, identifying a "common pattern" in detecting spontaneous and posed emotional facial expressions.

Different features were investigated to discriminate spontaneous and posed emotions automatically. For example, spontaneous smiles seem to have a slower onset speed, and larger duration than posed ones [114, 155, 228, 229]. Conversely, onset and offset speeds tend to be greater in posed smiles than the genuine counterpart [228]. Other features were also considered in the detection of spontaneous and posed facial displays, such as intensity [154, 157], symmetry of both sides of the face [86, 114], or the degree of irregularity (i.e., number of pauses or discontinuous changes in the phases of the expressions) of the emotional expressions [114, 124]. However, approaches used so far yielded great variability among the results and a lack of robustness [114, 138]. Consequently, to date, there is still skepticism about the real-world applications of the results obtained.

The problem of determining whether a given emotion is fake or not was proposed in the "ChaLearn Looking At People Real Versus Fake Expressed

Emotion Challenge"[253]. Five teams achieved the final stage and published their analysis. The best two models, which achieved the same performance, were proposed by NIT-OVGU (NTO) [226] and HCILab (HCL) [129]. The first proposed a method based on Support Vector Regression (SVR) ensembles to estimate AUs intensity frame by frame. The time series were smoothed using a first-order Butterworth filter, and 440 features were extracted. Finally, a Rank-SVM Ensemble was trained to detect the most authentic expression between two videos of the same participant, achieving an average of 67% accuracy. The second one (HCL) proposed a method based on an LSTM trained on each emotion after extracting facial landmarks from the videos. As in [226], the algorithm ranks a couple of videos to detect the most authentic expression between them. Also, in this case, the overall achieved accuracy was 67%. Even if this approach is suitable for the challenge set, the use of ranking algorithms between pairs of videos strictly limits the applicability of these methods.

Although some studies have reported promising detection accuracy on specific datasets (i.e., intra-dataset testing scenario), the performance can vary widely using the same detection method with different databases [138]. Indeed, the generalization and the improvement of these models is a problem still unsolved so far [138]. The machine learning models used so far yielded great variability among the results and a lack of robustness [114, 138]. Consequently, to date, there is still skepticism about the interpretability and real-world applications of the results obtained. The weak consistency among the results may be due to the high inter-individual variability in the facial displays of emotions [80, 126, 224, 260]. In [126] the inter and intra-variability of the subjects were measured in controlled smiles. The results showed how inter-individual variability achieved up to 60%, whereas the intra-variability was constant at 10%. Likewise, [106] investigated the facial muscles activity during elicited emotional experiences by means of EMG. The relative results showed how the corrugator activity evidenced substantial differences and individual variability between the subjects.

The poor performances maybe thus be because the datasets used for training models do not adequately take into account the real-world scenarios variability, an effect called dataset bias effect [148]. This could explain why the accuracy of these models drastically drops in real-world situations with spontaneous expressions [224, 80]. The previous analyses are, in fact, based on averaged values of subjects, an approach that may be called user-independent, and do not consider the specific individual variations [126]. This bias is particularly important considering that different factors such as gender, age, culture, morphological appearance strongly affect how emotions are exhibited

[54, 99, 110, 224, 257, 272]. Previous works on pain facial expression analysis have proved that person-specific models are advantageous in comparison with generic ones [224]. Accordingly, it would be an understatement to neglect the inter variability among the subjects in favor of a generalist approach. Facial displays are not identical for different subjects, and perhaps even each person does not have a unique expression for the same emotion [222].

In the current study, we made a step forward, trying to identify a specific pattern in the genuineness of the emotional displays for each subject (different from the previous user-independent scenario). In other words, Machine learning (ML) models were used to detect a unique fingerprint of genuineness singularly for each user. Moreover, a comparison with a more generic approach (i.e., user-independent) that neglects the specificity of the subjects' emotional displays in favor of an ensemble method was also investigated.

6.2 Data Collection

This section describes the procedures adopted to collect videos of genuine and posed facial expressions, the methods used to elicit them, and the process used to select the clips.

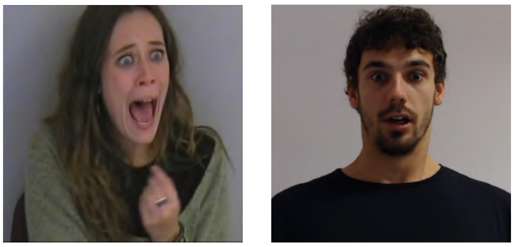
6.2.1 Participants Selection Procedure

Fifty-seven participants, aged between 20 and 30 years, took part in the experiment. Participants were enrolled using an advertisement on the University Website and were compensated for their participations. Participants signed an informed consent before the beginning of the experiments. After reading this informed consent, they were still unaware of the purpose of the study and were unaware of being filmed. The participants were informed that they had the right to quit the experiment and withdrew their consent at any time. At the end of the session, participants were debriefed, and the study's real aims were revealed. They were also told they were recorded. One participant withdrew her consent, and her clips were permanently removed from the database. The experimental procedure and the emotional elicitation protocol submitted to the participants and described in the following paragraphs were approved by the Ethics Committee of the University of Padua (Protocol number: 2917). The participants' video recordings were included in the database only after they signed a written consent to use their videos for research purposes.

6.2.2 Data Collection Setup

The aim of the experimental procedure was to record spontaneous (i.e. stimulus elicited) emotions of participants while they watched emotional video or were performing simple tasks. For this reason, participants were left alone in an experimental room to decrease the possibility that embarrassment and social inhibition could affect the spontaneity of expressed emotion, impacting on the overt manifestation of emotions. The doors and windows were kept shut during the entire protocol to avoid external interference and allow participants a more in-depth emotional excursion during the tasks. Participants were set about one meter in front of a Lenovo ThinkPad T490. As it is known that awareness of the experimental aim can interfere with the spontaneity of overt emotional expression [118, 231], participants were unaware of the purpose of the experiment. For this reason, a cover story was created. In particular, participants were told they have to rate emotional valence of the videos, as already did for a previous study [118]. They were also told that, in order to accurately assess emotions, they had to try to get immersed in the viewing experience and feel free to experience their emotions. Moreover, subjects were allowed to sit at their ease without any other restrictions inside the experimental room to avoid possible suspects or limit the emotions' naturalness.

The same protocol was submitted in two different ways to enrich the database and differently deal with acknowledged limitations of previous dataset. The first setting was created based on the well-known assumption that awareness of being filmed might impacts on spontaneity of overtly expressed emotions. Thus, in this first setting, a hidden camera placed at the right room's top angle was used. Participants were thus totally unaware of being recorded, preserving the emotional reactions' spontaneity. The clips were recorded with a AW-HE40HWEJ–Panasonic at a distance of at least 2 meters, with an angular size of 20°, varying in accordance with the head movements of subjects. The second setting was thought with the aim to create video depicting the participants on a frontal view. For this reason, in the second setting, a Logitech C920 HD Pro Webcam, Full HD 1080p/30fps, was placed at the top of the computer screen used for the tasks. In this setting, to preserve the subjects' expressions' spontaneity, participants were told that the recording was necessary to study the eye movements and pupil dilatation while performing the valence rating task. The two experimental setups guarantee more options to the experimenter who will use the emotional stimuli by having the same emotions (both spontaneous and posed) with a front and a lateral view (see Fig. 6.1).



(a) *First setting* (b) *Second setting*

Figure 6.1: Examples of fear expressions for the two settings.

6.2.3 Emotion Elicitation Procedure

Spontaneous emotional reactions were elicited with a multimodal protocol described in Table 6.1. Emotions were mostly triggered by watching emotion-inducing videos, which resulted to be the most effective stimuli for evoking emotional responses [45]. The clips were selected from different stimuli that have been used for similar studies [220], and from other sources such as international films, commercial spots, and YouTube clips. The length of the clips did not exceed 5 minutes according to the recommended size of the emotional video [220]. The emotions were not only elicited through passive elicitation by watching emotion-inducing videos. For example, anger was also triggered by using a rage game, well-tested stimuli to provoke anger, in which the emotion was elicited as a result of the encoder actively engaging with the game [236]. Indeed, the typology of these games was designed to make the task very difficult to purposely increase a high level of frustration and anger to the players. As, in pilots trails, we found that anger is often repressed, we provide participants with a desktop punching ball. Finally, as olfactory stimuli can reliably elicit disgust and have been resulted in very efficiently in previous studies [120, 121, 273], an unpleasant odor was presented to the subject to induce a disgusting feeling. The spontaneous emotion elicitation protocol is summarized in Table 6.1.

After the end of each task, participants were asked to identify the emotion they experienced/felt within the six basic emotion and neutral. They were also given the possibility to report if they felt an emotion that was not included within the six basic ones. Furthermore, besides identifying the emotion felt, they were also asked to rate how much the emotion they felt was genuine on a Likert scale ranging from -7 to +7 where -7 corresponded to “completely not genuine” and +7 corresponded to “completely genuine”, according with previous literature [69]. Finally, participants rated the intensity of the emotions experienced during the tasks on a Likert scale ranging from 0

("Emotion not felt/No intensity") to 9 (Emotion felt very intense/Very Strong Intensity") [69].

When the multimodal emotion elicitation protocol was successfully concluded, participants were asked to pose the six basic emotions multiple times, modulating the intensity of the posed emotions. In particular, participants were asked to pretend to feel a target emotion and to pose that emotion for at least 15 second different times trying to modulate its intensity. During this task, they were also asked to use the same objects they used during genuine emotion elicitations (i.e. punching ball and olfactory stimuli). After the end of each trial, they were debriefed about the emotions they felt and expressed and all of them confirmed they did not felt any kind of emotions, and thus that emotions expressed are to be considered not genuine as they were only posed but not felt.

6.2.4 Video Extraction

A certified Facial Action Coding System (FACS) coder, extracted the facial expression of emotions present in the recorded videos. The clips' selection was made considering both the FACS's criteria and participants' self-reports.

FACS is a widely used protocol for recognizing and labeling all visually discernible facial movements, called Action Units (AUs). The FACS manual proposes a list of possible combinations of AUs which are typically associated with expression of emotions [88]. The current method was used to reliably and accurately extract the emotional facial expressions shown by participants. In other words, the clips were selected only if the emotion elicited and conveyed by the face (e.g., happiness) matched: i) the target emotion for each task (in order to avoid to include emotions affected by other emotions); ii) FACS criteria (e.g., AU6+12) and iii) participants' self-report (e.g., they declare to have experienced happiness). Conversely, if participants reported having

[1] https://www.youtube.com/watch?v=URGUQlcAoNUab_channel=larablacklady

[2] https://www.youtube.com/watch?v=F2bk_9T482g&ab_channel=xXJEashXx

[3] https://www.youtube.com/watch?v=cLCE9_JHjPE&ab_channel=Mercating

[4] https://www.youtube.com/watch?v=JHXObtJYcyI&ab_channel=PhilBeastallFilms

[5] https://www.youtube.com/watch?v=4_B6wQMd2eI&ab_channel=WIACZO

[6] https://www.youtube.com/watch?v=OgrANlx7y2E&ab_channel=PhillipNorthfield

[7] https://www.youtube.com/watch?v=FZluzt3H6tk&ab_channel=ChaZacIsa

[8] https://www.youtube.com/watch?v=v3iPrBrGSJM&ab_channel=Quirkology

[9] https://flappybird.io/

[10] https://www.gioco.it/gioco/scary-maze

[11] https://www.youtube.com/watch?v=beAxdoCFnhw&ab_channel=COMPILATIONPOPPINGVIDEOS

Table 6.1: Multimodal protocol for Spontaneous and Posed emotion elicitation. Tasks are presented in this table in the same order they were presented to participants.

Task	**Emotion**	**Activity**	**Description**	**Lenght**
T1	Sadness	Watch a VIDEO: Death of Mufasa, from the Lion King [2]	The clip displayed the saddest part of the movie, when Mufasa dies because of Scar, and the touching reaction of Simba.	02:42 min
T2	Sadness	Disney Pixar Up[3]	The scene where Ellie and Carl are shown. Their relationship is being shown as time passes from their wedding to Ellie's death.	04:21 min
T3	Sadness	"Giving without expecting anything in return is the best communication"[4]	Spot for Telecom in Thailand. The story is about kindness rewarded over the course of 30 years.	03:08 min
T4	Sadness	"Love is a gift"[5]	It's a short film about a man counting down the days to Christmas so he can continue his yearly tradition sparked by a tragic moment from the past.	02:25 min
T5	Sadness	"Edeka 2015 Christmas Commercial"[6]	Edeka's holiday commercial reminds people of the important things in life in a tragic piece of storytelling.	01:30 min
T6	Surprise	The Invisible Gorilla[7]	An experiment in Change Blindness.	01:00 min
T7	Happiness	When Harry met Sally[8]	This is a classic and funny part to a very good movie. The restaurant/deli scene where Sally fakes an orgasm to prove a point.	02:46 min
T8	Surprise	Colour Changing Card Trick[9]	An experiment in Change Blindness.	02:43 min
T9	Anger	Flappy Bird[10]	A so-called "Rage game", namely a game while gaming and can't accomplish your goal whatever that is, and you get random from your lack of success.	05:00 min
T10	Fear	Scare Jump[11]	A so-called jump scare, namely a game intended to scare the audience by surprising them with an abrupt change in image, co-occurring with a frightening sound.	04:00 min
T11	Anger	Abused dog in a metro	The clip showed the abuse of a dog, beaten by his owner on a public metro.	03:00 min
T12	Fear	Scare jump horror clip	A classic horror clip aimed to scare participants with frightening scenes and spectral sounds.	02:28 min
T13	Disgust	Pimples squeezing[12]	Disgusting huge and ingrown pimples are squeezed in the clip.	05:00 min
T14	Disgust	Stinky potion	A solution characterized by an unpleasant smell that causes a strong reaction of disgust.	01:00 min
T15	-	Simulation Session	Participants were asked to pose each emotion for 30 seconds each, trying to change their intensity.	06:00 min

felt constrained and not natural in the emotional experience (e.g., a score of -4 on the genuineness scale), all the expressions associated with the task were removed. Likewise, if participants showed a facial expression associated with an emotion (e.g., a scowl that may reflect anger), the facial change was not selected if participants did not report to have experienced anger. In fact, a scowl is not always a cue of anger but could instead reflect confusion or concentration. This strict procedure aims to reduce the selection of facial expressions that do not convey authentic and spontaneous emotions. Each clip was cut from the onset point (i.e., the first frame when the expression is visible) to the apex (i.e., the period during which the movement was held at the highest intensity reached) of the emotion. Additionally, if the same emotion(s) was repeatedly elicited in a task, the target expressions were selected multiple times, in order to increase the number of clips included in the final dataset and provide more trials of the same emotion for each participant. Lightworks[13], a non-linear editing system (NLE) for editing and mastering digital video, was used to extract the emotional clips' perfect range frame.

6.3 Dataset Summary

PEDFE contains clips and static pictures of 56 participants, displaying subtle to full-blown elicitation of different emotions. Overall, the number of emotional clips is 1731 (the exact number clips for each emotion and category are provided in Fig. 6.2.

The duration of the facial expressions varied in accordance with the emotion displayed. For example, sad clips last longer (M=5.35s; SD=2.92s) than other emotions such as happiness (M=2.89s; SD=1.25s), disgust (M=2.81s; SD=1.33s) or anger (M=2.92; SD=1.38) because of the gradual evolution of sadness over a longer time-frame. Conversely, emotions like surprise (M=1.94s; SD=1.04s) or fear (M=1.86s; SD=0.92s) emerged and disappeared faster, lasting a few seconds at the most[90] The considerable amount of clips (i.e., 1731), as well as the self-reports given by participants, revealed the effectiveness of the elicitation protocol (please see Fig. 6.3, Fig. 6.4). In fact, most participants reported, on average, to have experienced the emotion that the elicitation tasks aim to do (except for Task 3). This was also confirmed by the intensity reported for each task, reflecting from medium to very high intensity (for the disgust tasks). Furthermore, the genuineness distribution rating revealed the spontaneity and genuineness of the emotional expressions

[13] https://www.lwks.com/

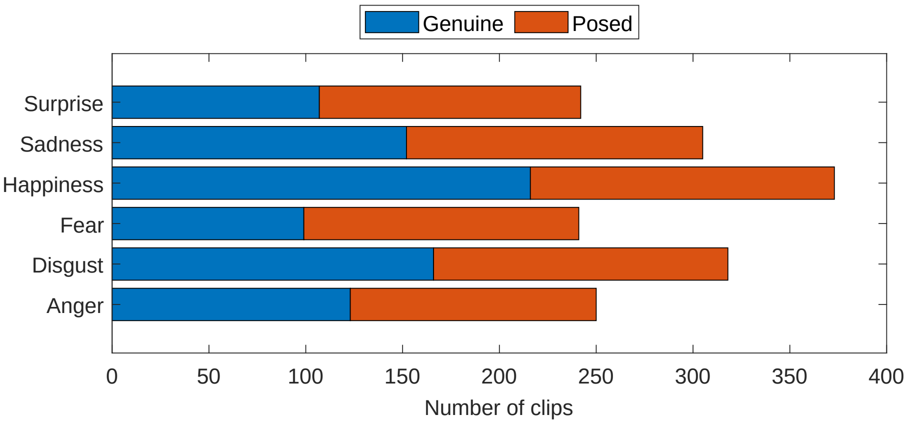


Figure 6.2: Number of clips before the validation, divided for emotion and type.

displayed by participants. However, as expected and already reported in similar studies [118], the elicitation and recording of facial expressions occurring spontaneous emotional experiences is empirically not easy [243]. Indeed, the emotional induction varied according to the subjective perception and sensitivity of the participants. For example, Task 1 (”The Lion King”) was reported as very sad by most of the subjects, while a few experienced fear or anger. Yet, in Task 11 (“Abused dog in a metro”), most participants revealed to have experienced anger. However, others reported sadness, surprise, or even no emotions (i.e., neutral).

Likewise, the intensity of the emotional excitement perceived varied across the tasks and between the subjects Importantly, the intensity reported in self reports is not predictive of the emotional expressions shown. For example, even though fear is reported as the second emotion per high level of intensity, the number of the clips is relatively low compared to other emotions (e.g., happiness). Moreover, not all subjects display the entire range of emotions. While happiness and disgust were easy to induce (see Fig. 6.1), other emotions such as fear and anger were challenging to elicit.

6.4 Dataset Validation

In this section we describe the validation process of PEDFE, providing an analysis of the results.

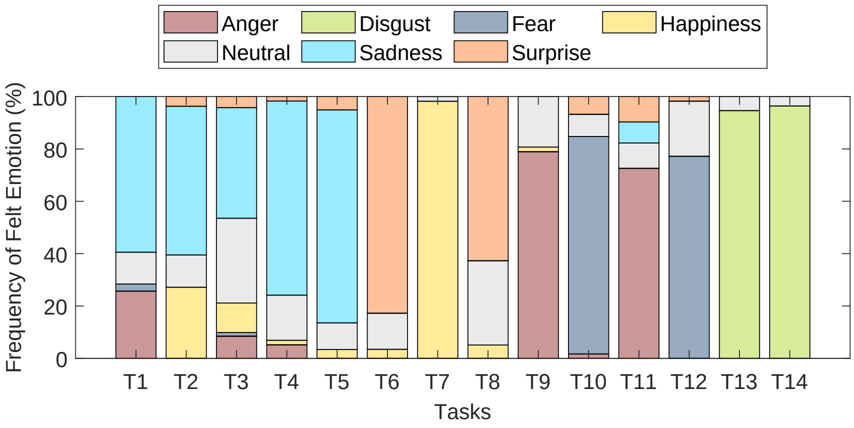


Figure 6.3: Emotion distribution from self-report for each task.

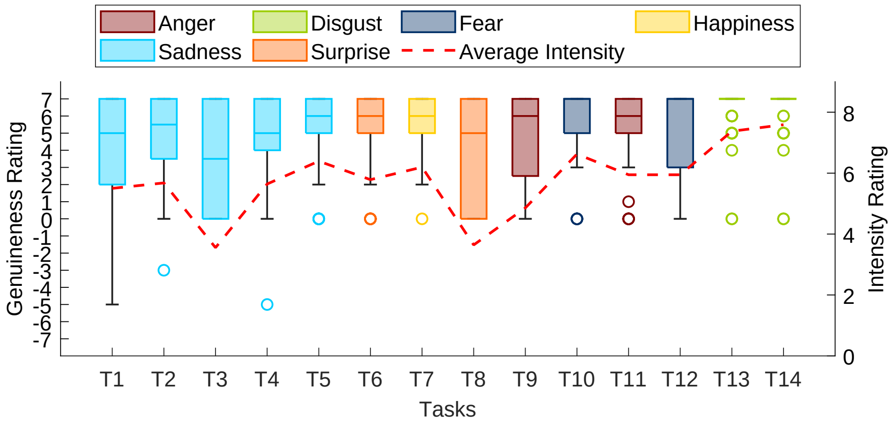


Figure 6.4: Genuineness and Intensity rate distribution for each task.

6.4.1 Participants

Being the number of stimuli very high (n=1731), they were split into four independent blocks, each of them including approximately 400 stimuli. Each rater was randomly assigned to one block. A total of 122 participants were recruited for the validation study, resulting in each block being validated by 30 independent raters. A further 29 subjects did open the link to the rating task but never started it (i.e., 23.8% drop-out). Of all 122 participants, 98 (80.3%) completed the entire rating, while 24 raters (19.7 %) did not. Among these, 25% (6 out of 24) completed more than 70% of the questionnaire. The rest of participants (18 out of 24) partially rated the validation (23.8% on average), and their data is included. Participants were all graduate students at the University of Padova (Italy). The majority of the participants were recruited through the institute's participant pool. Others were recruited from online University discussion forums.

6.4.2 Validation Procedure

The Validation Procedure was sent online to participants' email addresses using Qualtrics software [14]. Participants were shown short clips displaying facial expressions of anger, disgust, fear, happiness, sadness, and surprise from the PEDFE. During the validation session, the original audio was removed from the video, in order to avoid the results on emotion recognition and genuineness to be inflated by the presence of the audio. The validation was conducted according to [69].

After each of the emotional clips, participants were asked to categorize the emotion (they have to choose one within the six basic emotions, or neutral, or other, to give them the possibility to indicate an emotion not included within the six basic ones [100], and the type of expression (i.e., genuine or fake, on a Likert scale ranging from -7 -not genuine at all- to 7 –totally genuine-; [69]) displayed. The neutral midpoint "0" corresponded to "I do not know". This method allowed us to assess the ratings in absolute terms (i.e., genuine or fake).Furthermore, it provided information regarding the gradient of genuineness perceived by raters (e.g., +7 indicates that the emotion was perceived as genuine without any doubt by the observer, a different gradient from a score of +1, very close to "0"). Last, participants evaluated how intense the observed emotions looked to them on a Likert scale ranging from 0 (no intense at all) to 9 (extremely intense) [69].

Regarding the emotion recognition, we calculated the "hit rates" by dividing the number of accurately recognized emotions by the total number of displays for that emotion. Regarding genuineness recognition, we calculated the "hit rate" of genuineness by dividing the number of accurately recognized emotions as genuine or posed by the total number of displays. Simultaneously, the mean and the Standard Deviation (SD) of the gradient of genuineness were also calculated. Finally, the mean and SD of perceived intensity were calculated for each clip. The questionnaire took about 2 hours and 30 min to be completed. However, participants were strongly suggested to divide the questionnaire into three days (i.e., 45 minutes of task per day).

[14] http://www.qualtrics.com

6.4.3 Validation Results

The “hit rate for emotion” was adopted as the main exclusion criteria for the original 1731 clips. In fact, all the clips recognized with a ”hit rate for emotion” less than 30% were removed from the entire dataset, obtaining 1458 emotional clips (i.e., 707 spontaneous and 751 posed) in total. The list of the final stimuli, including the hit rates for emotion and genuineness, intensity and genuineness rating, as well as the duration of each clip is provided in Supplemental Material T1. In 6.2, the total number of clips included in PEDFE, as well as the hit rates, divided for emotion (e.g., disgust) and genuineness (i.e., spontaneous and posed), are reported respectively. Furthermore, the same analysis was conducted more in detail for every single subject actor included in the PEDFE's clips Notably, on average, regardless of genuineness (i.e., spontaneous or posed), all the emotions were categorized with an accuracy of 78.6%, ranging from 58.01% (for fear) to 93.66% (for happiness). As expected, happiness is the best-labeled emotion (both for spontaneous and posed expressions). Conversely, fear is the worst in accordance with the literature that reveal lower recognition rates of fear than the other basic emotions [221]. Further analyses were run in order to investigate if the cause of the low accuracy rating of fear was due to the misclassification with the surprise. To do this, we calculate the number of times the emotion was categorized as a surprise for each clip.

Results confirmed that, on average, fear is labeled as a surprise 29.76% of the time (SD 19.71%). Additionally, to evaluate if the intensity perception of the emotional expressions affects the emotion's discrimination, we conducted the Pearson correlation test. Importantly, the hit rate seems to be moderately affected by the intensity of the emotions expressed ($r = 0.44$, for 1458 items), in particular for anger expressions ($r = 0.67$ for 166 items). For what concerns the hit rate for the genuineness categorization, the global accuracy is stable across all the emotions (i.e., 62.51%), ranging from 60.22% (for disgust) to 65.25% (for fear). More precisely, genuine emotions were categorized better (i.e., 71.92% on average) than the posed ones (i.e., 53.65% on average), regardless of the emotion displayed (please see Fig. 6.5). Chi-squared test among all the binary responses extract by raters for each emotional stimulus confirmed the significant effect of the type of the stimuli (i.e., spontaneous or posed) on the hit rate of genuineness for each emotion with a $p < 0.00001$. In particular, anger $\chi^2(N = 4662) = 100.65$, disgust $\chi^2(N = 7719) = 221.97$, fear $\chi^2(N = 4049) = 164.53$, happiness $\chi^2(N = 10876) = 376.52$, sadness $\chi^2(N = 6619) = 172.65$, and surprise $\chi^2(N = 5823) = 100.94$. In other words, people tended to classify posed emotions as genuine more often than

Table 6.2: Total number of clips included in PEDFE, followed by their respective hit rates.

	TOT	POS	GEN	HR Emo TOT (%)	HR Emo POS (%)	HR Emo GEN (%)	HR Type TOT (%)	HR Type POS(%)	HR Type GEN (%)
Anger	166	90	76	64.88	69.30	59.64	60.92	56.36	66.33
Disgust	305	149	156	84.48	87.10	81.98	60.22	49.69	70.28
Fear	156	93	63	58.01	53.95	64.01	65.25	57.66	76.47
Happiness	370	156	214	93.66	93.42	93.84	65.02	47.85	77.53
Sadness	251	132	119	71.09	73.57	68.35	60.66	55.18	66.74
Surprise	210	131	79	78.70	85.44	67.52	62.84	58.81	69.51
ALL	1458	751	707	78.61	79.51	77.66	62.51	53.65	71.92

Note. TOT: Total number of clips; GEN: Number of Genuine clips; POS: Number of Posed clips; HR Emo TOT: Emotion Hit rate for the total number of clips; HR Emo POS: Emotion Hit rate for Posed clips; HR Emo GEN: Emotion Hit rate for Genuine clips; HR Type TOT: Genuineness Hit rate for the total number of clips; HR Type POS: Genuineness Hit rate for Posed clips; HR Type GEN: Genuineness Hit rate for Genuine clips .

they classify genuine as posed. Differently from the hit rate for emotion, these results are completely unrelated to the intensity ($r = 0.11$, for 1458 item) or the emotion ($r = 0.06$, for 1458 item) expressed.

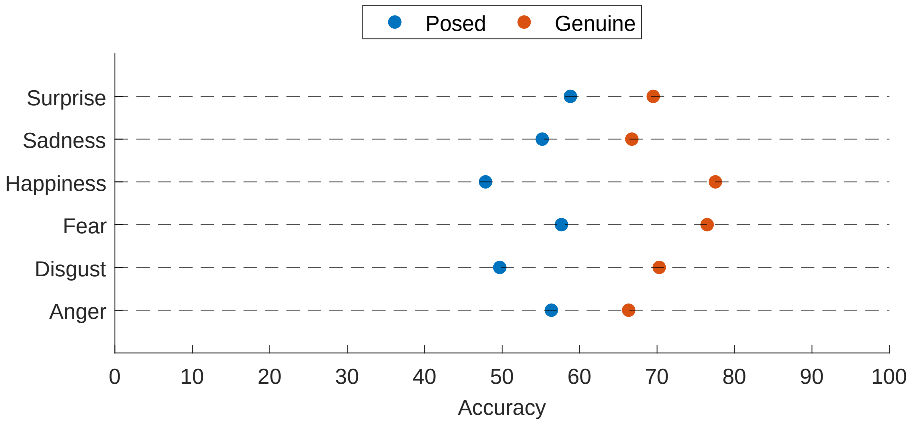


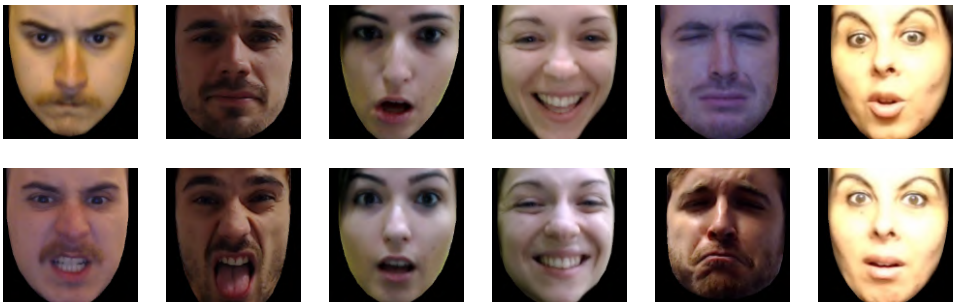
Figure 6.5: Genuineness Hit rate for each emotion.

6.5 Experimental Setting

In the current section, the framework used for the prediction of posed and spontaneous emotions is described. In Figure 6.7 the steps followed in our approach are depicted. In particular, four main steps have been identified:

1. *Stimuli*: Clips displaying spontaneous and posed facial expressions of the six basic emotions (i.e., happiness, sadness, anger, fear, surprise, disgust) were used for the current study. Stimuli were collected as

described in Section 6.2. A comparison between spontaneous and posed facial expression for the six basic emotions is reported in Figure 6.6 The dataset was divided into training (i.e., posed or spontaneous labeled clips) and test sets (i.e., unlabeled clips). Sets' size and composition depends on the configuration considered (i.e., user-independent or user dependent) and is discussed in detail in Section 6.5.1.



(a) *Anger* (b) *Disgust* (c) *Fear* (d) *Happiness* (e) *Sadness* (f) *Surprise*

Figure 6.6: Peak intensity images of genuine (first row) and posed expressions (second row) of the six emotions included in PEDFE and processed by OpenFace.

2. *Data Processing*: The Facial Action Coding System (FACS) represents the gold standard to detect and describe every single facial appearance accurately, also note as action units (AUs) [91]. To automatically extract AUs from the set of emotional stimuli (AUs Activation Detection), all the videos were processed using OpenFace [17]. OpenFace, as far as we know, is the best state-of-the-art free software for AU extraction. It estimates the activation level of 17 AUs for each frame (see Table 6.3), providing two metrics: binary (active/non-active with predetermined threshold) or continuous (it assumes a value between 0 and 5, where 0 corresponds to inactive and 5 to maximum activation). A total of 136 features per video were extracted from the metrics provided by OpenFace (Feature Extraction). More precisely, 5 groups of features were calculated for each AU, as reported in Table 6.4.

3. *Model Generation*: Five binary Machine Learning models were trained on different groups of features and validated to select the best model/feature configuration for the prediction of posed and spontaneous emotions. In particular, Support Vector Machine with RBF kernel (SVM RBF), linear Suppor Vector Machine (SVM Linear), Ridge classifier (RC), Decision Tree (DT), and Random Forest (RF) were

Table 6.3: List of the Action Units detected by OpenFace with the description of their facial movements.

AU	FACS name	Appearance Changes
1	Inner brow raiser	The inner portion of the eyebrows run vertically from the top of the head to the eyebrows
2	Outer brow raiser	The lateral (outer) portion of the eyebrows upwards
4	Brow lowerer	The eyebrows are pulled together and lowered
5	Upper lid raiser	The upper eyelid is raised, widening the eye aperture
6	Cheek raiser	The cheeks are lifted upwards, raising the infraorbital triangle
7	Lid tightener	The eyelids get tightened
9	Nose wrinkler	The skin along the sides of the nose upwards towards the root of the nose causing wrinkles
10	Upper lip raiser	The upper lip is pulled upwards and towards the cheek, pulling the upper lip up
12	Lip corner puller	The corners of the lips are pulled back and upward (obliquely)
14	Dimpler	The corners of the mouth are tightened and pulled inwards, narrowing the lip corners
15	Lip corner depressor	The corners of the lips are pulled down
17	Chin raiser	The chin is pushed upwards, pushing up the lower lip
20	Lip stretcher	The lips are pulled back laterally, elongating the mouth horizontally
23	Lip tightener	The lips and skin around the lips are tightened and thinned
25	Lips part	The mouth is opened, separating the lips
26	Jaw drop	The mandible is lowered by relaxation
45	Blink	The eyes are closed and opened very quickly

used as prediction models. A 5-fold cross-validation was applied on the training set to select the best combination of features. Further, hyper-parameters were varied by using the grid search on all five considered classifiers. Specifically, for SVM RBF C was varied among $[10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}]$, and γ in the range $[10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}]$. For SVM Linear C was varied in $[10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}]$. For both the SVM models a Standard Scaler was applied to normalize the input. The α parameter for RC was tested in the range $[10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}]$. For DT the max depth was varied in the range $[2, 3, 4, 5]$. Finally, for RF number of estimator were set in $[20, 50, 100, 500]$, and the max depth was varied in $[2, 3, 4, 5]$.

4. *Prediction*: The selected models and group of features are used to perform the prediction on the testing clips.

6.5.1 Application Scenarios

The significant amount of clips, as well as the several samples for the same subject and for the same emotion, allowed us to use two main analysis scenarios: user-independent and user-dependent. These two scenarios were applied in order to investigate how the role of the intra-individual variability affects the detection of spontaneous and posed emotional facial expressions in automatic classification.

User-independent scenario The user-independent scenario intends to identify a common (deception) cue to detect each emotion's spontaneous or

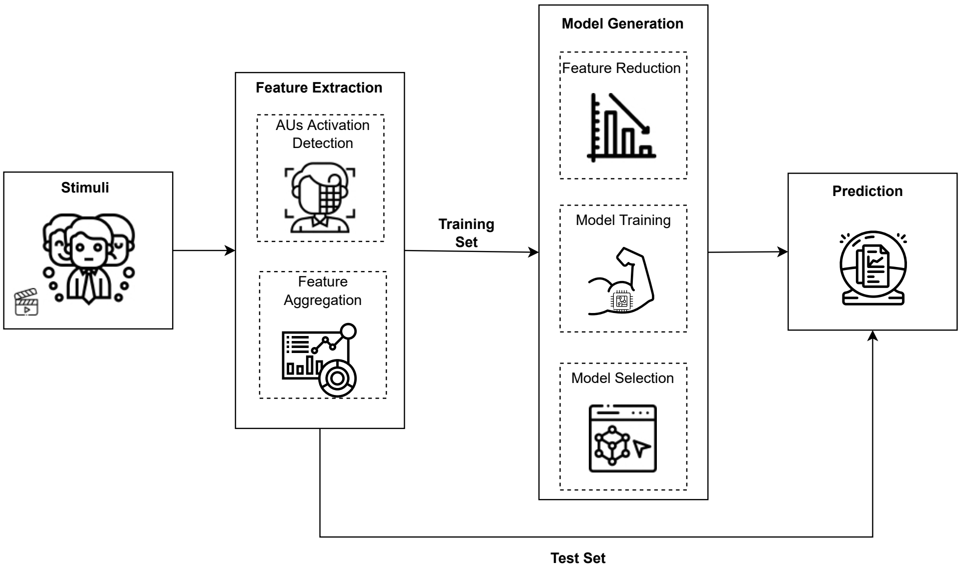


Figure 6.7: Framework for the automatic detection of spontaneous and posed emotional facial expressions.

Table 6.4: Groups of feature extracted from OpenFace output leveraging the Action Units (AUs) and the Activated Action Units (AAUs) activity. Where $i, k \in 1, 2, 4, \ldots, 45$ are variables ranging over the action units, and $n \in \{0, \ldots, N\}$ is the frame number.

Feature Group	**Description**	**Formula**
Activation	Average and standard deviation of AUs'intensity	$Mean_i = \frac{1}{N}\sum_{n=0}^{N-1} AU_i(n)$ $SD_i = \sqrt{\frac{1}{N}\sum_{n=0}^{N-1} (AU_i(n) - Mean_i)^2}$
Normalized Activation	Normalized average activation and normalized standard deviation of the AU activation per frame	$NMean_i = \frac{1}{N}\sum_{n=0}^{N-1} \frac{AU_i(n)}{\sum_{k=1}^{45} AU_k(n)}$ $NSD_i = \sqrt{\frac{1}{N}\sum_{n=0}^{N-1} \frac{(AU_i(n)-NMean_i)^2}{\sum_{k=1}^{45} AU_k(n)}}$
Duration	Activation duration of the AU normalized on frames number	$Dur_i = \frac{1}{N}\sum_{n=0}^{N-1} AAU_i(n)$
Speed	Average and standard deviation of the changes in AU activity	$SMean_i = \frac{1}{N}\sum_{n=0}^{N-1} AU_i(n+1) - AU_i(n)$ $SSD_i = \sqrt{\frac{1}{N}\sum_{n=0}^{N-1} [(AU_i(n+1) - AU_i(n)) - SMean_i]^2}$
Entropy	Average dispersion of AU activity across subsequent frames	$Ent_i = \frac{1}{N}\sum_{n=0}^{N-1} \lvert AAU_i(n+1) - AAU_i(n) \rvert$

posed facial expressions. This scenario assumes that the proposed approach is applied to an unknown subject (i.e., the subject is not present in the training set). To simulate this scenario, a leave-one-subject-out was applied: all the subjects were used for training the models except one used for testing (i.e., the unknown subject). This procedure was looped for each subject included in the dataset to avoid information leaking from the tested subject. In other words, no previous information (i.e., clips) about the tested subject is available in the training phase. In particular, a nested cross-validation (CV) was implemented. The outer loop consists of a leave-one-out CV per subject, while the inner loop consists of a group 5-fold CV used in the *Model Generation* phase for model validation, model selection, and feature selection (please see Figure 6.7). This procedure was repeated for each emotion separately, generating a total of six models (i.e., one for each emotion).

User-dependent scenario The user-dependent scenario aims to classify the genuineness of emotional facial expressions of a specific subject based on its already known emotional displays. In other words, the information (i.e., clips) of the subject was used for training the models in order to classify the genuineness of a new clip of the same subject. The application of the aforementioned scenario is twofold: first, to identify a fingerprint of genuineness in the emotional facial expression of each user; second, to investigate the impact of the inter-individual variability among the users' emotional displays. Contrarily to the user-independent scenario, all the clips of the specific user (i.e., previous information) were used to train the models except the one used for testing (i.e., the subject's unknown clip).

This procedure was looped for each clip of the subject by using a nested CV per clip. In particular, in the outer loop, a leave-one-out CV per clip was performed, while in the inner loop, a 5-fold CV was implemented for the *Model Generation* phase. Subjects with less than 20 clips were excluded from the analysis for lack of sufficient information in the training phase of the models.

6.6 Experimental Results

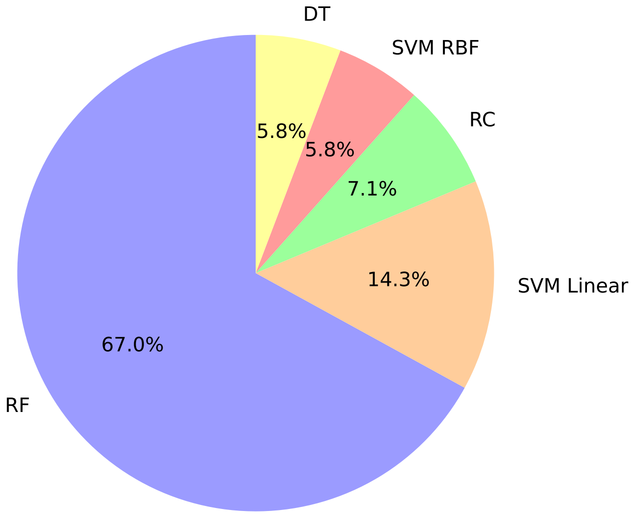
The experimental results obtained in the user-independent scenario, yielded an overall accuracy of 67.0%. In particular, for anger was obtained an accuracy of 62.4%, for disgust 61.7%, for fear 67.4%, for happiness 65.4%, for sadness 69.9%, for surprise 75.5%. In this scenario, RF resulted the most selected model, followed by SVM Linear (see Fig. 6.8a).

A significant improvement was obtained in the user-dependent scenario, where an overall accuracy of 84.4% was achieved. Specifically, the following accuracies were obtained for anger, disgust, fear, happiness, sadness, and surprise respectively: 90.1%, 82.2%, 84.6%, 81.7%, 89.8%, 83.2%. Differently from the first scenario, SVM RBF resulted the best model for the majority of the users, followed by RC (see Fig. 6.8b). The significant improvement in the genuineness classification can also be noted for each emotion singularly (see Fig.6.9). In particular, the user-dependent scenario increased the performances by 27.7% for anger, 20.5% for disgust, 17.2% for fear, 16.3% for happiness, 19.9% for sadness, 7.7% for surprise, with an overall improvement of 17.4%.

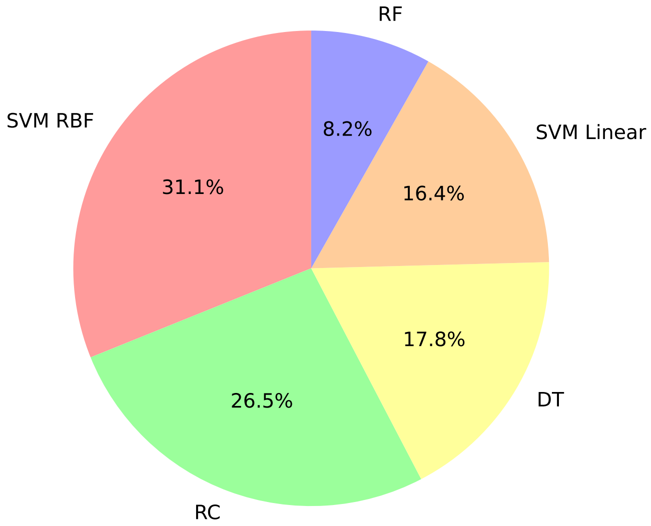
A further investigation was conducted by analyzing the differences in the classification for each user in both scenarios. The radar graph displayed in Fig. 6.10 confirmed the enhancement of performances in the genuineness classification for every single user. Our framework showed an increase in the overall accuracy between user-dependent and user-independent scenarios for 94% of the users. Analyzing each emotion (see Fig. 6.11) for the 90% of the users the prediction accuracy improved for anger genuineness classification. Sadness, fear, and disgust showed an improvement in 87%, 85%, and 83% of the users, respectively. Finally, surprise and happiness reported an improvement in 76% and 75% of users.

6.7 Discussion

In the current chapter, Muscle Movement (Action Units) based ML models were used to automatically discriminate spontaneous and posed emotions. More precisely, five binary machine learning models were adopted in two different settings, namely user-independent and user-dependent. In the first setting, a leave one out nested CV per subject was used across the whole dataset of clips, recursively and randomly subdividing training, validation, and test set for each emotion regardless of the subject's identity displayed in the clips. The user-independent approach was used to identify the common differences between spontaneous and posed emotions with no regard to the inter-individual variability of the subjects that performed the emotional facial expressions. Contrarily, in the second setting, a leave one out nested CV per clip was used singularly for each subject, thus splitting training, validation, and testing set only across the clips of the subject regardless of the emotion displayed. The latter approach was used in order to take into account the potential inter-user variability in the emotional display. The results were of particular interest as they revealed a significant difference between the two



(a) *User-Independent scenario*



(b) *User-Dependent scenario*

Figure 6.8: Frequency of selected models in our framework per scenario.

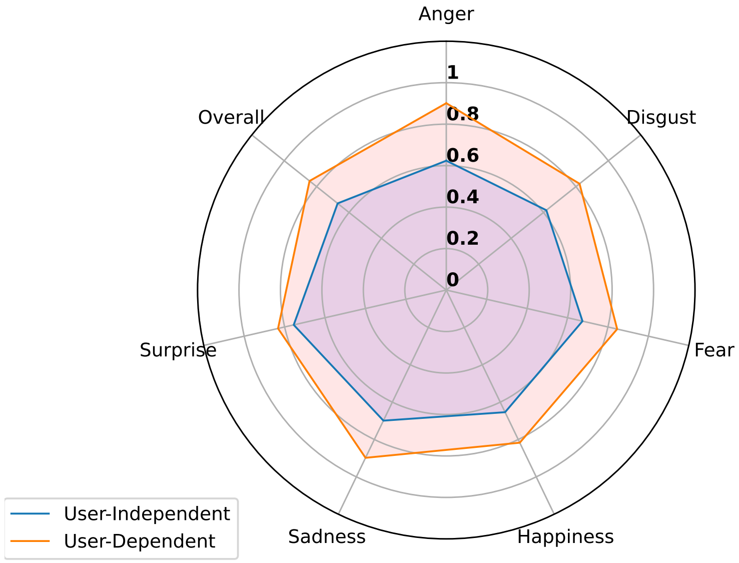


Figure 6.9: Accuracy detection in user-independent and user-dependent scenarios across all the emotions. Each axis of the radar graph represent the emotions investigated while the y axis inside the graph reflects the accuracy.

approaches even though the same models were implemented on the same features. In particular, the user-independent approach achieved on average 67.0% of accuracy. In contrast, the user-dependent approach performed with a 84.4% accuracy, reaching up to 90.1% accuracy for sadness emotion.

The comparison of the performance between the two scenarios, highlights the significant differences across all the subjects' emotional displays. The general framework spontaneous vs posed emotions used in the user-independent scenario was partially able to identify a keystone about the emotional lie detection in facial displays. However, the current analysis totally neglected the individual variations in the emotional displays, causing a drop in the performances if compared to the second approach. In fact, the same approach gained an overall 17.4% improvement if adopted singularly for each subject, and thus if they were specialized ad hoc for each user without trying to generalize a unique facial patterns to every user (i.e., user-dependent scenario). The implications of the present research are relevant on multiple levels.

First, concerning the emotional lie detection applications, it seems that it would be more reliable to focus on detecting the unique deceptive cues for each subject instead of identifying a common rule to discriminate spontaneous and posed emotional facial expressions generally. In other words, the significant

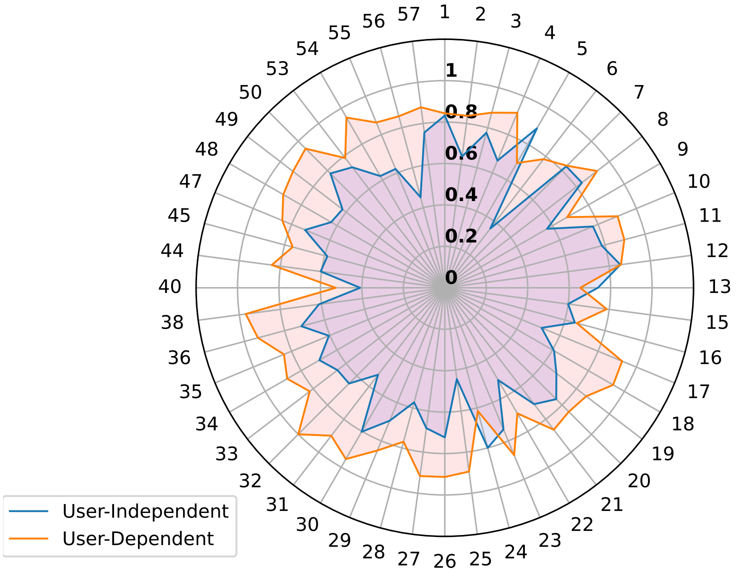
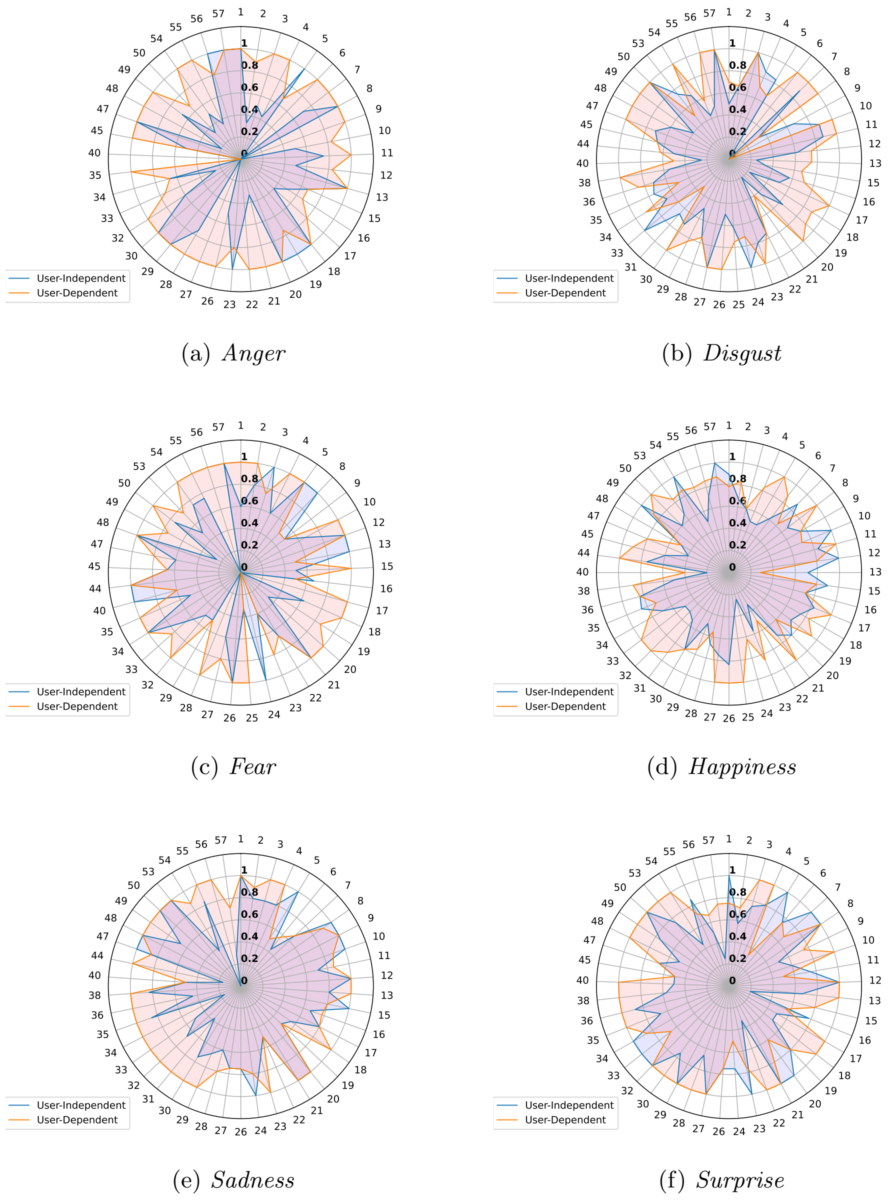


Figure 6.10: Accuracy detection in user-independent and user-dependent scenarios across all the subjects on average for all the emotions. Each axis of the radar graph represent the number of the subject while the y axis inside the graph reflects the accuracy.

inter-individual variability in people's emotional display may underestimate the intra-individual differences between spontaneous and posed emotional displays of each subject. Consequently, it may seem that, despite some similarities (detected with 67% accuracy by user-independent scenario), each subject tends to have a specific strategy or deception fingerprint that discern spontaneous from posed emotions. This factor may partially explain the inconsistent results obtained so far in the automatic detection of the genuineness of emotional facial expressions [114, 138, 141, 173].

Second, these results remarked the higher inter-individual variability in the facial displays of emotions, already highlighted in previous studies [80, 126, 224, 260]. In other words, the valuable differences and individual variability between the subjects reflected in different characteristics (e.g., gender, age, morphological traits [54, 99, 110, 224, 257]) was revealed to be an essential factor to be considered.

Third, these results are particularly interesting also in relation to the universality of emotions (i.e., basic emotion approach) proposed by [84]. The basic emotion theory claims the existence of prototypical facial configurations for some given emotion categories (i.e., basic emotions) [85, 87]. For example,



(a) *Anger*

(b) *Disgust*

(c) *Fear*

(d) *Happiness*

(e) *Sadness*

(f) *Surprise*

Figure 6.11: Accuracy detection in user-independent and user-dependent scenarios across all the subjects for the six emotions. Each axis of the radar graph represent the number of the subject while the y axis inside the graph reflects the accuracy.

according to the theory, the facial core configuration of anger is typically reflected by furrowing the brows, widening the eyes, and tightening the lips. Additionally, some variants may involve the opening of the mouth or include the narrowing of the eyes only [88]. According to that, it would be possible to read people's emotional states universally, and, more important, it would be possible to discriminate spontaneous from posed emotions basing on temporal (e.g., onset time of the expression), and morphological cues (e.g., reliable muscles) [1, 186]. As a consequence of that, the models used should have been able to generally discriminate spontaneous and posed basic emotions across all the subjects, without any concern about the significant potential variability in the emotional displays. However, the user-independent approach partially confirmed a common pattern between all the subjects. In fact, even though it is possible to find a slight similarity in the emotional deception for all the individuals included in the dataset, the intra-individual analysis (i.e., user-dependent approach) was revealed to be more accurate and precise than the general approach (e.g., user-independent approach). In other words, albeit some inter-subjects similarities found by ML models, our results yield significant differences between subjects. These results align with the recent theories of emotions that refuse the universality of emotional facial displays. In particular, other scientific frameworks suggest that the facial configurations of emotions may vary substantially across different people and situations [21]. In particular, the behavioral ecology view (BECV) proposes that facial expressions are flexible tools that mute over time for cultural or natural reasons and may cause diversity across people [63]. This could explain why the user-dependent approach outperformed the generic user-independent approach. However, it is fair to assert that these results may also depend on the simplicity of the models and features used that are not able to generalize to all people's emotional displays (i.e., user-independent) like in the user-dependent scenario.

Finally, the importance of the intra-individual variability is also relevant in relation to the use of the recent emotion recognition software that claim to be able to read emotions in people based on their facial expressions (e.g., Affectiva.com, 2018; Microsoft Azure, 2018). These API systems aim to generalize their predictions in the open world, neglecting both the intra-individual variability among the emotional displays and the discrepancy in the performance between spontaneous and posed facial emotions [79, 156]. Different bias already emerged in the performance of the machine learning algorithms used, such as the gender or the age of people [127, 149, 152, 57] The current results provide additional proof about the variety in the emotional displays, both in spontaneous and posed emotions, raising further doubts about the methodology used in emotional facial recognition. Additionally, the current research restates the necessity of a methodology based on the single user, emphasizing the significant differences among the individuals, and abandoning the idea of a collective and equal emotional facial display.

Nonetheless, the findings of this study have to be seen in the light of some limitations. The sample size used for the analysis is composed of only 56 subjects. Moreover, the features extracted are limited to the descriptive statistics of action unit movements and do not consider dynamic and temporal elements (e.g., onset, offset time, asymmetry, acceleration). Finally, only five models were used in the current research. Other models may be revealed more effective and accurate in the same task. Therefore, the empirical results reported herein should be interpreted and considered with caution. Future studies are needed to address the generalization of these results.

6.8 Summary

The automatic genuineness detection of emotional facial expressions is a topic still debated and controversial in the state of the art of lie detection. In the current chapter, ML models were used to predict the genuineness of emotional facial expressions in general and specifically for every single user. The framework used was revealed to be a promising approach to apply in future research, and highlighted how inter-individual variability could be a significant factor to consider. Finally, the related findings were discussed in light of the state of art of lie detection, psychology of emotions, and artificial intelligence.

Chapter 7

BLUFADE: Blurred Face Detection

To begin using any modern computing device (e.g., desktop, workstation, laptop, tablet, or smartphone), the user must be authenticated. During the authentication process, the user is typically asked to demonstrate possession or knowledge of one or more of: (1) a secret, such as a password or PIN, (2) a biometric, such as a face or fingerprint, and (3) a device, such as a secure dongle or smartphone. Massive investments were made over the years to create and support secure means of user authentication.

At a later time, when the user ends (or abandons) its current session on a logged-in device, so-called *de-authentication* must ideally take place. However, in contrast with authentication, de-authentication received substantially less attention since lack thereof is not perceived as necessary as lack of (or insufficient) authentication. This is unfortunate since an unattended active secure session triggers the very real danger of *Lunchtime Attacks* [81]. Such attacks can occur whenever an adversary gains physical access to the active session of another user who carelessly stepped away and left the logged-in device unattended.

This motivates the need for secure, privacy-preserving, and usable de-authentication techniques. However, prior results do not satisfy all these three requirements. For instance, the popular means of de-authentication via inactivity timeouts can be considered somewhat[1] privacy-preserving. However, if timeouts are too long, it offers poor security as the lunchtime attack time window grows. Whereas, if timeouts are too short, usability

[1] Timeouts are not very privacy-preserving since they monitor user's typing and/or mouse activity.

suffers since the user might need to re-authenticate needlessly [235]. Other methods continuously authenticate the user, and de-authentication occurs once the user's identity can no longer be verified. Common techniques rely on detecting physical presence of the user [175, 177, 51].

We believe that continuous face recognition is a promising means of de-authentication. It tracks and identifies previously authenticated user's face as long as it is visible from the webcam; once the user's face disappears from view (for a specific time interval), de-authentication occurs. This general approach offers several benefits. First, it is easy to implement and does not require extra equipment since most modern general-purpose computing devices are equipped with video cameras. Second, it is secure because current face detection algorithms are fast and highly accurate [181], making it resistant to *Lunchtime Attacks*. Third, it keeps the user authenticated and logged in, even if keyboard or mouse activity stops, as long as the user's face remains within line-of-sight of the webcam. This is in contrast with methods based on inactivity intervals, keystroke dynamics [19] or gaze–tracking [81], where users have to interact with the system continuously or frequently.

However, face recognition in de-authentication is hampered by significant **privacy concerns**. First, most users would not want to be video-recorded continuously. Even if the rules explicitly state that recordings are not stored anywhere, users might (rightfully) not trust such promises and refrain from (or attempt to circumvent) using such a method. Second, an attacker who gains access to the webcam or recordings could exploit this information for malicious purposes. Blackmailing a user recorded during private moments is just one of many possible threats.

Nonetheless, most modern devices are equipped with user-facing cameras, and despite the manufacturers' assurances that cameras only operate in tandem with some user-visible indicator (e.g., an LED light in, or next to, the camera), many users find the constant presence of the camera unnerving. In fact, on some computers with integrated cameras, it is possible to surreptitiously turn on the camera and record **without** triggering the obligatory indicator [37].

Due to privacy and safety concerns, many cautious users have been applying physical barriers (e.g., placing tape) on their webcams [174]. This practice was publicly supported by the ex-FBI director James Comey [119], and some manufacturers now deliver laptops with built-in sliders to cover webcams.

Motivated by the above discussion, we propose `BLUFADE`, a de-authentication system based on continuous face detection that provides user privacy, security, and usability. We apply a physical blurring material on the webcam that obfuscates users' facial traits, making them unrecognizable. Then, after demonstrating that state-of-the-art face detection models perform very poorly on blurred images, we implemented a deep neural network for this specific task. We tested our system with 30 subjects in different scenarios and activities, reaching over 95% detection accuracy.

Contributions. The main contributions of this chapter are:

- A novel secure, usable, and privacy preserving de-authentication method based on blurred face detection
- Its evaluation via extensive experiments, demonstrating that it outperforms state-of-the-art algorithms on blurred face detection tasks
- Publicly released two datasets of physically blurred faces: the first one consists of $20k$ images of celebrities and backgrounds, blurred with two different materials, and the second contains $1,080$ enrollment images and 600 videos of 30 subjects interacting with a laptop (both blurred).

7.1 Related Work

Related work stems from several areas, including de-authentication as well as face recognition and detection.

7.1.1 De-Authentication

In contrast with authentication techniques, which are extensively studied in the literature and are widely used in everyday life, there are no standard or broadly adopted user de-authentication methods. This reflects the fact that users are forced to authenticate at the beginning of a login session, while de-authentication is almost never mandatory. Locking the screen or logging out during a short break (e.g., coffee, bathroom, hallway chat, lunch) is widely perceived as being tedious or unnecessary (i.e., 25% of the users leave their computers unlocked when stepping away from their desk [50]). However, as mentioned earlier, failure to de-authenticate opens the door for lunchtime attacks, which are pretty common, as noted by Marques et al. [179]. Thus, the research community tried to come up with secure, usable, and privacy-preserving techniques for *automatic* user de-authentication.

The simplest de-authentication method is to log out the user after a fixed keyboard/mouse inactivity period. However, choosing the duration of this period is not trivial [235]. Recent techniques rely on Continuous Authentication (CAuth): the user is continuously monitored and authenticated while interacting with the system, and de-authentication happens once these interactions stop. CAuth usually relies on some form(s) of biometrics usually based on recognition of: face [223, 175], voice [202], motion [65, 233], keystroke and/or mouse dynamics [19], and even video-game playing style [52]. For an extensive list of these techniques, we refer to [11, 122].

Of the above, keystroke dynamics is popular and seemingly non-intrusive while requiring no special equipment, whereas others need a camera and/or a microphone, which must be turned on. Keystroke dynamics utilize the user's unique typing style (reflected in a profile created at enrollment time) for authentication. While easy to deploy, this approach is not secure since an attacker can reproduce the user's typing style [245]. Carrying around a unique token that communicates with the workstation is another option [55]. However, its prominent drawback is the requirement to always carry and protect this token. A similar approach is explored in ZEBRA [177]: the user is continuously authenticated using a personal bracelet as long as wrist movements and the computer actions match. Unfortunately, [128] showed that Zebra is insecure. More complex and exotic systems, e.g., based on gaze–tracking [81] and pulse–response [211] have been proposed. Since they require pricey specialized equipment, thus their applicability is quite limited.

All aforementioned techniques have a major common drawback: a user can be authenticated only when **interacting** with the device. Consider the following frequent everyday activities that involve no interaction (no keyboard, mouse, or touchscreen actions) while the user remains physically present:

- Reading something on-screen or printed
- Watching a video/movie
- Listening to music or podcast
- Making a phone-call
- Taking a seated nap
- Having an in-person conversation with someone

Any of such activity, once it exceeds the inactivity threshold, would cause automatic de-authentication, resulting in extra user burden or even DoS. To overcome this issue, several methods have been proposed. FADEWICH [51] instruments an office with position sensors to detect whether the users are sitting at their desks. *Assentication* [143] detects user presence through pressure sensors in the chair cushion. Whereas, [53] instruments a chair with BLE beacons to detect whether the user is currently sitting. Facial recognition can be used for CAuth by continuously monitoring faces that appear in front of the camera, while being user-transparent [64, 203, 223]. In this chapter, we focus on detection – rather than recognition – of faces, since most facial features would not be visible for privacy reasons. Since the user is already logged in, it is enough to trace the presence (detection) of their face.

7.1.2 Face Detection and Recognition

Face detection and face recognition are distinct Computer Vision tasks thoroughly studied in recent years. We consider face recognition a subclass of face detection, since the algorithms first start by detecting a face and then use its features to compare to a set of known faces to recognize the person. In early stages, face recognition was done by automatically extracting distinctive facial features, e.g., eyes, mouth, or nose. These features were used to transform the face into a vector, and using statistical pattern recognition techniques, faces were matched [38, 145]. With the rise of deep learning, especially Convolutional Neural Networks (CNN), computers reached (and surpassed) human performance in such tasks [204]. Deep-learning-based face recognition techniques can be divided into: (1) ones using single CNN [111, 146], (2) multi CNNs [172], and (3) variants of CNN [269]. For a comprehensive list of face recognition methods, refer to [113, 137].

Similar to face recognition, early face detection methods were based on developing discriminative hand-crafted features from faces and building robust learning algorithms [264, 274]. Nowadays, with the evolution of CNNs, detecting frontal faces is considered a solved task [181]. More efforts took place to detect faces under challenging conditions, such as partial faces [266] or faces captured by depth sensors [34]. Recently, TinaFace [279], by considering face detection as a particular object detection task, outperformed state-of-the-art methods on the set of most challenging face detection dataset WIDER FACE [265]. We refer to [270] for a complete treatment of this topic. Finally, [276] tested state-of-the-art face detection models on low-quality images with different levels of blurring, noise, and contrast, showing that both hand-crafted and deep-learning-based face detectors perform poorly on such images.

7.2 Model Overview

We now describe our system model and its real-world application scenarios.

7.2.1 System Model

The core idea is to use a webcam (built-in or external) to detect the user's face continuously. At the beginning of the session, the user authenticates by any canonical method, e.g., passwords or fingerprint recognition. Then, BLUFADE collects images at regular intervals from the webcam, keeping the user authenticated as long as a face is detected. Once the detection fails and a grace period passes, the user is automatically logged out. To preserve user privacy, the webcam view is *physically* blurred by a somewhat-transparent tape or a similar means. Thus, users can be sure that the images received by the webcam are already altered and cannot be used to recognize them. We note that BLUFADE's goal is to detect, and not to recognize, faces since the tape should blur the image enough to obscure facial traits.

Besides privacy, BLUFADE offers the usual benefits of face detection de-authentication mechanisms. First, is completely transparent for the user, since it does not interfere with normal user behavior, and prevents *Lunch Time Attacks*. Furthermore, it only requires a simple strip of tape as additional equipment, and allows the user to remain inactive without being de-authenticated, as long as they remain in the camera's view. The main implementation challenges are: (i) selecting an appropriate material that obscures users' facial traits, while still allowing face detection by automated algorithms, and (ii) developing an algorithm to detect faces from blurred images. (i) is analyzed in Section 7.4, and (ii) in Section 7.5.

7.2.2 Application Scenario

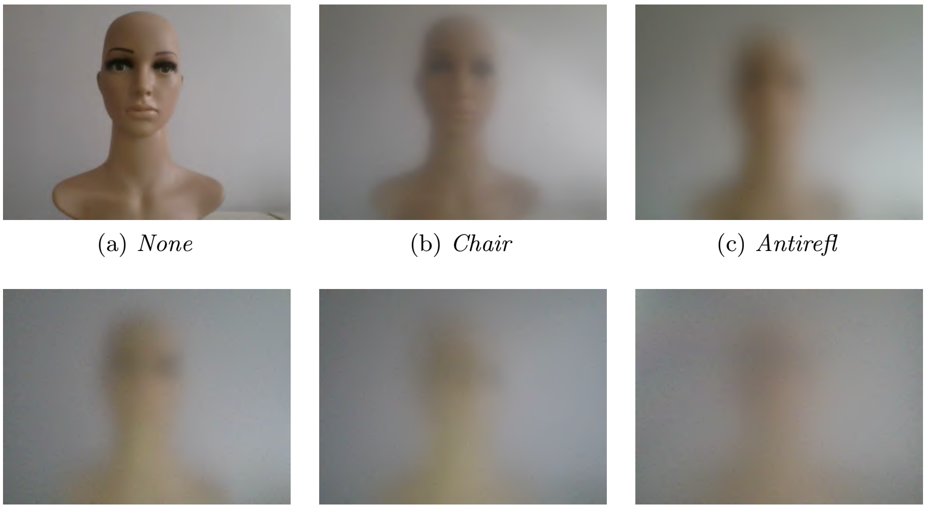
We start by distinguishing between shared and personal computers. We assume that the latter is always used by the same person; thus, the detection system can be tailored to their blurred face. The phase of training the software to recognize a face is called *enrollment*. In shared computer settings, the system is used by multiple users and should detect all of them. Thus, the enrollment is complicated and should be done to every new user, which is clearly not applicable. The second distinction concerns the place where the system is used. A computer can be stationary or portable, which defines the scene its webcam sees when no users are present (i.e., the "background"). If stationary, the background is fixed; otherwise, it will vary depending on the place. Based on that, we identify four scenarios:

- **Scenario 1 - Same person and fixed background:** represents workstations or desktops, located in an office/home and is always used by the same person. Enrollment is possible;
- **Scenario 2 - Different people and fixed background:** represents shared workstations in fixed places (e.g., offices). Enrollment is not applicable;

- **Scenario 3 - Same person and variable background:** represents personal computers, e.g., laptops or tablets, that owners can bring anywhere. Enrollment is possible;
- **Scenario 4 - Different people and variable background:** represents shared computers that are either portable and/or have variable backgrounds, e.g., public ATMs or wheeled workstations. Enrollment is not applicable.

7.3 Material Evaluation

One of the critical design elements for BLUFADE is how to choose the appropriate blurring material. In this section, we discuss the criteria for this selection (Section 7.3.1), and the experimental settings to determine the best candidates in terms of suitability for face detection (Section 7.3.2).


(a) *None* (b) *Chair* (c) *Antirefl*

(d) *Ruvid* (e) *RuvidX2* (f) *Scotch*

Figure 7.1: Effectiveness of blurring materials considered at a distance of 30 cm.

7.3.1 Selection Criteria

The ideal blurring material should satisfy three requirements: (i) blur enough to prevent face recognition, (ii) not blur too much to enable face detection, (iii) be inexpensive and readily available. Based on these requirements, we identify five possibilities[2]:

[2] Chair: https://bit.ly/3i9Vjm8, Antirefl: https://bit.ly/3CN14xS, Ruvid: https://bit.ly/3m3KZ0i, Scotch: https://bit.ly/3zMUOV8.

- ***Chair*** - Polimark Poliver Battisedia 280854. Semi-transparent rigid plastic material that is commonly used on floors to prevent chairs from scratching them;
- ***Antirefl*** - Polimark Poliver PL01322. Anti-reflective obfuscating film, commonly used on windows to block visibility from the outside but letting light to pass through;
- ***Ruvid*** - Ruvid Transparent Paper. Transparent rough paper used as book covers;
- ***RuvidX2*** - Double Ruvid Transparent Paper. Double layer of the previous item;
- ***Scotch*** - Magic Tape Scotch 3M. Common semi-transparent white adhesive tape;

7.3.2 Experimental Settings & Best Candidates

To find the best blurring material, we evaluated the quality of blurred images produced by a webcam when various materials were applied. To this extent, we used a mannequin called Dolores[3] as a fixed subject of our photos. For each material, we positioned Dolores in front of the webcam at several distances (from 30 cm to 90 cm, with 10 cm steps), simulating realistic usage scenarios. We used a white background in a light-controlled environment. At each distance, we took five snapshots, and used three samples of each material. Then, we assessed image quality (i.e., sharpness) using the algorithm presented in [70], and averaged the results. Figure 7.1 shows pictures of Dolores taken with different blurring materials, while Figure 7.2 shows the quality of images for all materials and steps. A lower Niqe value indicates the image has an higher sharpness. The plot shows that all blurring materials significantly lower image quality and that the distance from the webcam does not meaningfully influence the Niqe value. Ideally, the lower the image quality, the more challenging the face recognition by automatic systems. Thus, we selected two materials yielding highest quality images (*Chair* and *Antirefl*), which from visual inspection (examples are visible on Figure 7.1) could preserve users' privacy. The following section provides more evidence on their privacy features and discusses material selection.

[3] The name was chosen from an analog situation from the TV series Umbrella Academy.

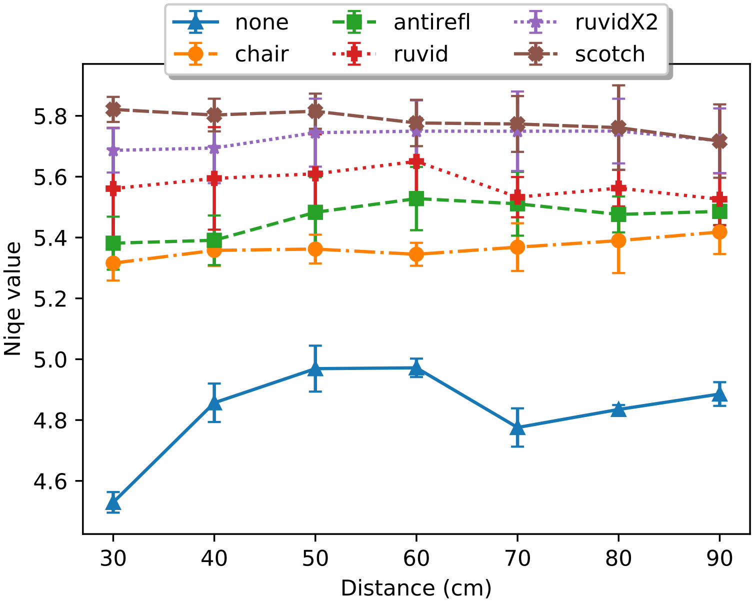


Figure 7.2: Averaged quality of images for each material and steps. Lower Niqe values are associated to sharper images.

7.4 Material Selection

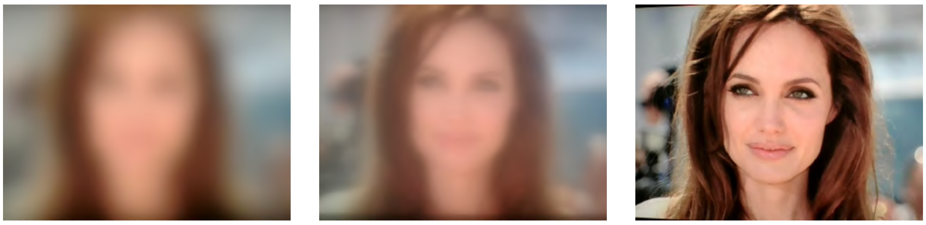
To select the best material among the two candidates from the previous section, we need to evaluate their privacy-preserving characteristics. To this extent, we first collected a dataset of blurred pictures of celebrities (Section 7.4.1), and we conducted a survey asking the participants to recognize some of them (Section 7.4.2). Last, we report the results and final decision (Section 7.4.3).

7.4.1 Celebrities Dataset

To the best of our knowledge, there are no physically blurred faces datasets publicly available. Furthermore, to carry on our experiments, we need images of both blurred backgrounds and faces with the materials we selected in Section 7.3.2. To create such a dataset, we exploit the CelebA dataset [168] and the SUN dataset [262]. In particular, we randomly selected 5000 images from CelebA (faces) and 5000 images from SUN (backgrounds). Then, applying the *Chair* and *Antirefl* filters to a laptop webcam, we recorded a slideshow of the 10K images displayed on a tablet. Finally, we picked a frame in correspondence of each image from the recording, creating two new datasets of 10K blurred images each. The dataset is available at the following link: https://spritz.math.unipd.it/projects/BLUFADE/

7.4.2 Celebrities Privacy Survey

We conducted an online survey asking participants to recognize celebrities from blurred images to test whether the blur level was enough to protect users' privacy. In particular, we selected ten images of well-known celebrities in a neutral context, and we asked participants to guess their names. For each image, first, we presented the *Antirefl* version, then the *Chair* version, and last the original image (i.e., from the less sharp image to the most). The participants were asked to provide a name at each step, without the possibility to go back and change the name after seeing a less blurred image. If the name provided at the last step was correct (we also accepted names with spelling errors), we could assume the participant knew the celebrity, and thus we checked at which blur stage the participant recognized them. If the participant did not know the celebrity, we discarded that sample. Figure 7.3 shows an example of a celebrity blurred with the two filters and the original photo.



(a) *Antirefl* (b) *Chair* (c) *None*

Figure 7.3: Angelina Jolie with different blur filters.

7.4.3 Survey Results and Material Decision

We collected answers from 70 participants (Age range: 22-45, 64.3% Male, 35.7% Female). 391 images were recognized correctly with no blur, 273 with *Chair* blur, and only 5 with *Antirefl*. In other words, participants recognized a celebrity they knew only in 1.28% of the cases through the *Antirefl* filter, and in 69.8% of the cases through *Chair*. Thus, we demonstrated that *Antirefl* successfully protects users' privacy, and we decided to use it for the rest of the experiments.

7.5 Experiments

We now present the experiments we conducted to evaluate BLUFADE. In Section 7.5.1, we illustrate the data we collected for the experiments. Section 7.5.2 evaluates the face detection state of the art models on our data. Last, we propose our model in Section 7.5.3.

7.5.1 Data Collection

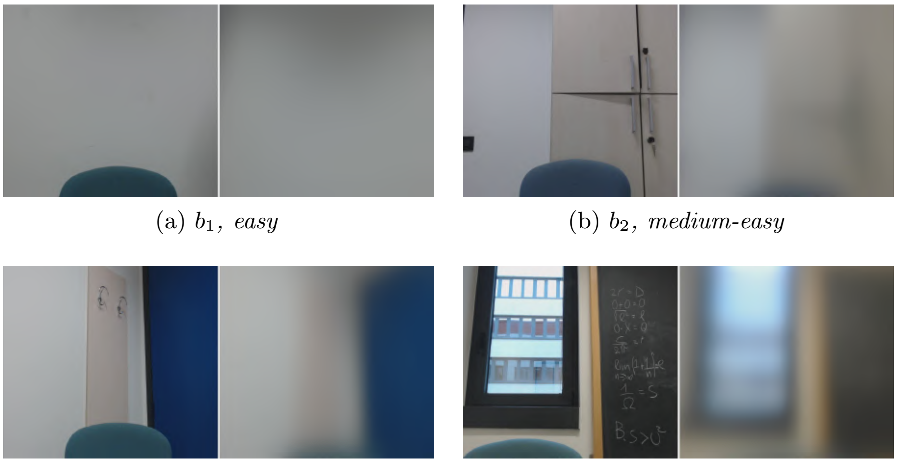
To conduct our experiments, we collected data from 30 people, 13 females and 17 males, aged 22-43. According to the scenarios presented in Section 7.2.2, we first asked participants to follow an enrollment procedure, and then we recorded them while performing common everyday actions. In detail, the enrollment procedure consisted in taking snapshots of the user in 9 different positions: in front of the webcam at close distance (i.e., less than 30 cm), mid-range distance (between 30 and 70 cm), and far (more than 70 cm); at mid-range translated to left and right (i.e., the face should be completely contained in the left or right half of the webcam view); at mid-range rotating the head by looking up, down, left, and right. Then we recorded users for 10 seconds while reading an email, writing sentences, looking at their phones, talking with a colleague, and leaving the workstation. Users repeated these steps on four different *backgrounds* $b_n \in \mathcal{B}, n = \{1, 2, 3, 4\}$ of increasing difficulty: a white wall (b_1 - easy), a white wall with a closet and a poster (b_2 - medium–easy), a white wall with a blue door (b_3 - medium–hard), a white wall with a written blackboard and a window (b_4 - hard). They are shown in Figure 7.4. We used a Logitech C922 Pro Stream Webcam (30 Frames Per Second) with *Antirefl* blur for the recordings. This dataset is available at the following link: https://spritz.math.unipd.it/projects/BLUFADE/

7.5.2 State of the Art Face Detection Algorithms

The performance of BLUFADE highly depends on the face detection algorithm behind it. Before implementing our neural network, we tested the state-of-the-art face detection systems on both our celebrities and enrollment blurred images. To this extent, we extracted 240 random celebrities and 240 random enrollment images and tested with Google Cloud Vision[4], Amazon Rekognition[5], Azure Cognitive Services with detection_01 and detection_03

[4] https://cloud.google.com/vision/docs/detecting-faces

[5] https://docs.aws.amazon.com/rekognition/latest/dg/faces.html



(a) b_1, *easy*

(b) b_2, *medium-easy*

(c) b_3, *medium-hard*

(d) b_4, *hard*

Figure 7.4: The four different backgrounds used in the experiments (left original, right blurred with *Antirefl*).

models[6], and TinaFace [279]. Results are reported in Table 7.1, and they show how any of the state-of-the-art models were not suitable for our task, given the high level of blur of our images. Even Azure v3, explicitly designed for blurred faces, with 72.08% of accuracy, was not good enough for `BLUFADE`.

Table 7.1: Comparison between accuracy of state-of-the-art face detection models on blurred samples from Celebrities and People datasets

	Google	***Amazon***	***Azure v1***	***Azure v3***	***TinaFace***
Celebrities	1.67%	43.75%	0.04%	45.83%	13.75%
People	3.33%	26.25%	0.00%	72.08%	18.75%

7.5.3 Proposed Model

The poor performances of state-of-the-art methods in detecting blurred faces suggest that a new approach is needed for this task. Since the high level of blur removes facial traits, we decided to shape our problem as an object detection task, as also suggested by Zhu et al. [279]. Rather than binary classification (i.e., face vs. no face), we opted for object detection also to possibly track the person, or detect two or more people in the same image for security purposes. For instance, if a person is logged in and using the

[6] https://docs.microsoft.com/en-us/azure/cognitive-services/face/face-api-how-to-topics/specify-detection-model

computer and another user walks behind the first user, the system should detect which person is keeping the session alive; otherwise, it might wrongly de-authenticate the user. Furthermore, [249, 276] demonstrated that CNNs do not cope well with blurred images, but fine-tuning them can help to improve the performances in object detection significantly. From these considerations, we decided to fine-tune the state-of-the-art object detection model RetinaNet [164], which uses ResNet and Feature Pyramid Network as back-bone for feature extraction. We followed an official procedure released by TensorFlow [244]. In particular, our fine-tuning procedure follows these steps: starting from ResNet pre-trained using the COCO dataset [165], we replace the classification head with a new randomly initialized classification head able to classify a single class (i.e., face), and we finally fine-tune the network using 150 batches of 32 samples each, with SGD optimizer (learning rate = 0.01, momentum = 0.9).

7.5.3.1 Four Scenarios

To represent the four scenarios from Section 7.2.2, we used the enrollment snapshots and the activity videos to create different training and test sets. In general, enrollment images are used in the training set, while activity videos are used for testing purposes. For each scenario, we test person by person and background by background, creating every time a training set that respects the requirements of the scenario to fine-tune the neural network. We remind that every person p of our dataset of people $\mathcal{P}$ has taken 9 enrollment snapshots for each background b of the 4 backgrounds $\mathcal{B}$ analyzed (from easy to hard). We refer to the 9 enrollment images of a person p in a background b as $e_{p,b}$. In more details, we use a leave-one-out test procedure, testing at each iteration the activity videos of a person $p \in \mathcal{P}$ in a background $b \in \mathcal{B}$, and setting the training (fine-tuning) set as specified in Table 7.2. In the table, we give formal and informal explanations on how we constructed the training set to understand the scenarios easily.

7.5.3.2 Regular People vs. Celebrities

To introduce more variance in the training set, we also run some experiments using the celebrities dataset. The first set of experiments was run using only people's snapshots as a training set. Then, we repeated the experiments adding in the training set $1,080$ celebrities' faces, and the last repetition was done fine-tuning the network using celebrities only. This way, we could see how the variance in the training set affects performance of network detection. The case of celebrities only was possible just in the fourth scenario, since

Table 7.2: Training set composition according to specific application scenario. $\mathcal{P}$ and $\mathcal{B}$ are sets of participants and backgrounds, respectively.

Scenario	***Formal Training Set***	***Explanation***
1) Same person and fixed background	with p, b fixed, $i \in \mathcal{P}$, $j \in \mathcal{B}$, $\bigcup_{\substack{\forall i \neq p \\ \forall j \neq b}} e_{i,j} \cup e_{p,b}$	All enrollment snapshots of people different from p in backgrounds different from b + enrollment of p in background b
2) Different people and fixed background	with p, b fixed, $i \in \mathcal{P}$, $j \in \mathcal{B}$, $\bigcup_{\substack{\forall i \neq p \\ \forall j}} e_{i,j}$	All enrollment snapshots of people different from p
3) Same person and variable background	with p, b fixed, $i \in \mathcal{P}$, $j, k \in \mathcal{B}$, $\bigcup_{\substack{\forall i \neq p \\ \forall j \neq b}} e_{i,j} \cup e_{p,k} \| k \neq j$	All enrollment snapshots of people different from p in two backgrounds (j, k) different from b + enrollment of p in the remaining background different from b, j, k
4) Different people and variable background	with p, b fixed, $i \in \mathcal{P}$, $j \in \mathcal{B}$, $\bigcup_{\substack{\forall i \neq p \\ \forall j \neq b}} e_{i,j}$	All enrollment snapshots of people different from p in backgrounds different from b

it was impossible to have their enrollment or more celebrities in the same background.

7.5.3.3 Confidence Threshold

RetinaNet returns the objects it detects along with their confidence scores. Based on a threshold, usually 0.80, the object is detected or ignored. Since our data is highly blurred and strongly differs from usual data, we had to find a proper threshold for the task. We used the more general celebrities in this case since it has thousands of faces and thousands of backgrounds without faces. Using the celebrities instead of the people dataset to find the threshold, we would have limited the possibility of overfitting. Thus, we fine-tuned the network with the same 1080 celebrities we used to augment the people training set, and we tested the network on the remaining celebrities and backgrounds of our dataset. Then, we tried different thresholds ranging in [0.100,0.125 0.150, . . . , 0.900], selecting the one which gave the best accuracy (i.e., threshold = 0.425). We used this threshold for the rest of the experiments.

7.6 Results and Discussion

We now present the results of our experiments. Section 7.6.1 shows the performance of the face detection task. In Section 7.6.2, we evaluate the performance of `BLUFADE` in de-authenticating people. Last, we discuss current limitation of our system in Section 7.6.3.

7.6.1 Face Detection

Table 7.3 reports the balanced accuracy of face detection on the frames of the activity videos divided by scenarios, backgrounds, training datasets, and tasks (T1 = read email, T2 = write sentence, T3 = look phone, T4 = talk with colleague, T5 = leave workstation). As expected, we reach the best performance on the easiest background *b*1, with around 98% accuracy on every scenario using the people scenario, 97% also using the celebrities, and 94% in the celebrities only case. Among the tasks, T1, T2, T3 scores the best, probably because are composed of frontal frames of the people. In T4, people were talking with a colleague on their left or right, showing the webcam their face profile. This has probably lead to some mistakes. Finally, T5 shows some errors during the transition period in which the user is leaving. In fact, we considered the user had completely left only when the face was not more visible, and the network struggled a bit with partial faces or with just

Table 7.3: Balanced accuracy of face detection of frames of activity videos divided by scenarios, backgrounds, training datasets, and tasks (T1=read email, T2=write sentence, T3=look at phone, T4=talk with colleague, T5=leave workstation).

		Scenario 1					**Scenario 2**					**Scenario 3**					**Scenario 4**				
Training	*Task*	b_1	b_2	b_3	b_4	***Avg***	b_1	b_2	b_3	b_4	***Avg***	b_1	b_2	b_3	b_4	***Avg***	b_1	b_2	b_3	b_4	***Avg***
People	T1	1.00	0.99	0.95	0.86	**0.95**	1.00	0.99	0.95	0.91	**0.96**	0.99	0.99	0.93	0.80	**0.93**	0.99	0.98	0.93	0.82	**0.93**
	T2	1.00	0.99	0.95	0.87	**0.95**	1.00	0.99	0.95	0.94	**0.97**	0.99	0.99	0.94	0.80	**0.93**	1.00	0.99	0.93	0.83	**0.94**
	T3	0.99	0.99	0.90	0.84	**0.93**	1.00	0.99	0.92	0.93	**0.96**	0.99	0.99	0.89	0.78	**0.91**	0.99	0.98	0.88	0.79	**0.91**
	T4	0.98	0.94	0.88	0.80	**0.90**	0.98	0.96	0.90	0.93	**0.94**	0.98	0.95	0.87	0.73	**0.88**	0.99	0.94	0.86	0.72	**0.88**
	T5	0.94	0.94	0.91	0.77	**0.89**	0.94	0.94	0.92	0.79	**0.89**	0.91	0.90	0.88	0.68	**0.84**	0.94	0.93	0.90	0.75	**0.88**
	Overall	**0.98**	**0.97**	**0.92**	**0.83**	**0.92**	**0.98**	**0.98**	**0.93**	**0.90**	**0.95**	**0.97**	**0.97**	**0.90**	**0.76**	**0.90**	**0.98**	**0.97**	**0.90**	**0.78**	**0.91**
People & Celeb	T1	0.99	0.98	0.94	0.74	**0.91**	0.99	0.99	0.96	0.90	**0.96**	0.99	0.99	0.93	0.72	**0.91**	0.99	0.99	0.94	0.78	**0.93**
	T2	0.99	0.98	0.95	0.80	**0.93**	0.99	0.99	0.96	0.93	**0.97**	0.99	0.99	0.94	0.80	**0.93**	0.99	0.98	0.96	0.83	**0.94**
	T3	0.99	0.98	0.87	0.72	**0.89**	0.99	0.99	0.90	0.88	**0.94**	0.99	0.98	0.84	0.68	**0.87**	0.99	0.98	0.87	0.74	**0.90**
	T4	0.94	0.89	0.84	0.62	**0.82**	0.95	0.91	0.84	0.82	**0.88**	0.94	0.90	0.79	0.61	**0.81**	0.95	0.90	0.82	0.66	**0.84**
	T5	0.93	0.92	0.90	0.74	**0.87**	0.94	0.92	0.91	0.78	**0.89**	0.94	0.92	0.88	0.73	**0.87**	0.93	0.91	0.89	0.73	**0.87**
	Overall	**0.97**	**0.95**	**0.90**	**0.72**	**0.89**	**0.97**	**0.96**	**0.91**	**0.86**	**0.93**	**0.97**	**0.96**	**0.88**	**0.71**	**0.88**	**0.97**	**0.95**	**0.90**	**0.75**	**0.89**
Celeb	T1	-	-	-	-	**-**	-	-	-	-	**-**	-	-	-	-	**-**	0.99	0.95	0.91	0.65	**0.87**
	T2	-	-	-	-	**-**	-	-	-	-	**-**	-	-	-	-	**-**	0.99	0.97	0.93	0.73	**0.91**
	T3	-	-	-	-	**-**	-	-	-	-	**-**	-	-	-	-	**-**	0.96	0.94	0.76	0.57	**0.81**
	T4	-	-	-	-	**-**	-	-	-	-	**-**	-	-	-	-	**-**	0.84	0.79	0.73	0.52	**0.72**
	T5	-	-	-	-	**-**	-	-	-	-	**-**	-	-	-	-	**-**	0.92	0.88	0.88	0.70	**0.84**
	Overall	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**0.94**	**0.91**	**0.84**	**0.63**	**0.83**

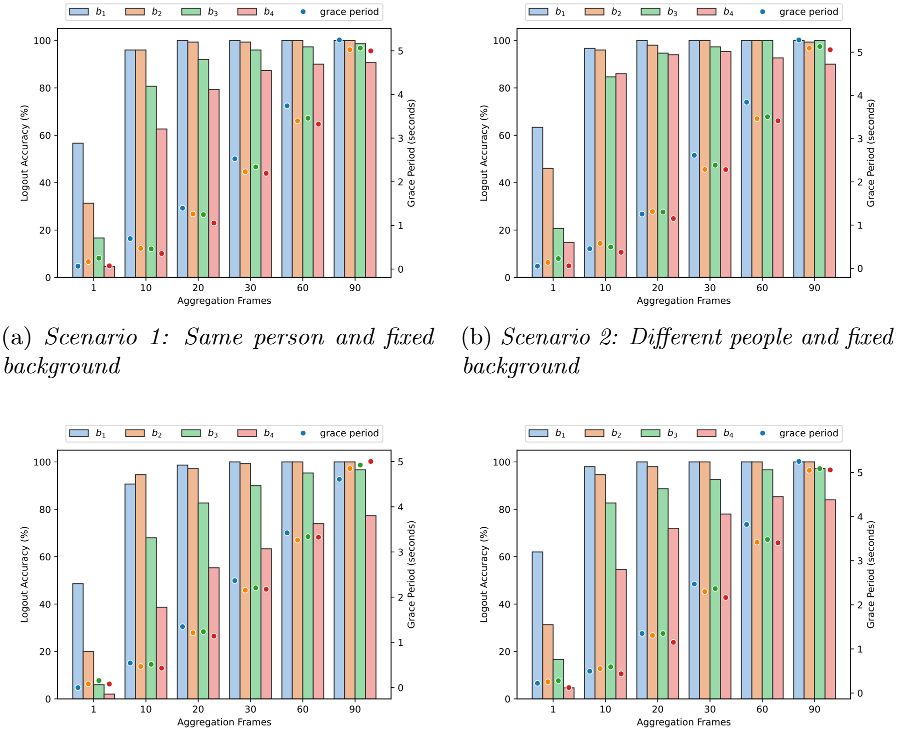
the body. However, when the user was fully present or absent, the network worked just fine as in the other tasks. In Section 7.6.2, we better analyze this task to implement the de-authentication system.

Looking at the scenarios, surprisingly, those without enrollment (i.e., Scenarios 2,4) show slightly better performances than the others. This could be explained by the lacking of real unique traits in the enrollment images. Having a wider variance in the training set helps the network in detecting people in different tasks. In fact, the great differences are again in T4, T5, thus a more general network can help in such difficult tasks. Finally, better performances are achieved when the training set is formed by people only. This is understandable since the training and test set are more similar. Adding the celebrities lowers the performances, but not significantly. We lose around 2% in each scenario, but still achieve 90% accuracy, which is a good result. We believe that adding more variance in the training set as in this case could help on a real-world situation with a lot of different people and backgrounds. Finally, using only celebrities to fine-tune the network leads to the worst accuracy, but still the average is above 80%, which is remarkable since training and test set are very different. Comparing our results with the one state of the art models (Table 7.1), we clearly outperform them. Against Azure v3, specifically built to detect blur faces, we score around 35% and 20% more on celebrities and people respectively.

7.6.2 BLUFADE Performance

Face detection is the heart of BLUFADE. By detecting the user's face frame by frame, we are able to understand when they leave and de-authenticate them accordingly. Even with results above 90%, which are generally good in the computer vision area, we still need an improvement to provide users a reliable de-authentication system. In fact, de-authenticate them every time the neural network fails the prediction is not desirable, and can negatively impact the users' experience. To improve BLUFADE, we can consider two crucial aspects: i) the neural network commits sparse, not sequential mistakes, and ii) the de-authentication has not to be instantaneous. In fact, the literature identifies a "grace period" in which the user might be still logged in even though they already left. Obviously, this period must be short enough to not allow lunchtime attacks, and is based on the fact that users, in that period, can notice if someone is trying to steal their active session. A good grace period is below six seconds [51].

Following these considerations, BLUFADE performs face detection and evaluates the results using a sliding window of aggregates frames. The de-authentication occurs once the face is no detect for N consecutive frames. N can be 1, which means that at the first frame the face is not detected, BLUFADE de-authenticates the user, or higher. In our experiments, we tested different values of N, to a maximum of 90, which means 3 seconds (the webcam recorded at 30 FPS). Figure 7.5 shows the logout accuracy (i.e., the times BLUFADE correctly logs out a user) per different level of N (aggregation frames) and the corresponding grace period needed to log-out the user. The four graphs represent the four scenarios, and each bar in the plot represents a background accuracy, while the dots indicates the grace period. These graphs refer to the experiments using the people dataset only, which achieved better scores than using People and Celebrities or Celebrities only. We discuss these two cases later in this section.



(a) *Scenario 1: Same person and fixed background*

(b) *Scenario 2: Different people and fixed background*

(c) *Scenario 3: Same person and variable background*

(d) *Scenario 4: Different people and variable background*

Figure 7.5: Average logout accuracy (bars) and average grace period (dots) for different aggregation frames and application scenarios.

In general, the de-authentication accuracy trends reflect the underlying face detection system. For all the application scenarios, the accuracy increases as the aggregation does. Considering an aggregation frame equal to one, BLUFADE would wrongly de-authenticate users too frequently (i.e., over 60% on average in all the scenarios and backgrounds), making our system not usable. On the other hand, considering an higher number of aggregation frames (i.e., 90 frames) the logout accuracy rate increases up to 100% for Scenario 1 (Figure 7.5a) and scenario 2 (Figure 7.5b) in b_1 and b_2, keeping the grace period under 5 seconds. Scenario 3 (Figure 7.5c) shows the lowest performance of BLUFADE, with an accuracy below 80% even with 90 aggregation frames in b_4. However, the other backgrounds show very high scores with a grace period under five seconds.

Considering all scenarios together, the difficulty of the backgrounds highly impacts the performances. More difficult is the background, less the accuracy. Starting from 30 aggregation frames, b_1 reaches 100% of accuracy in all the scenarios, keeping the grace period below 3 seconds. b_2 shows similar performance, reaching 100% of accuracy in less than 4 seconds in all the scenarios when the aggregation frames is equal to 60. b_3 shows more than 95% accuracy with 90 aggregated frames in about five seconds, while b_4 struggles a bit especially in the third scenario. These data reveals that BLUFADE can work incredibly well when the background is an empty wall or with simple decorations, like in a common work office, and struggles a bit with challenging backgrounds. However, when the background is fixed, BLUFADE always performs above 90%.

Figure 7.6 compares the averaged BLUFADE performances in all the scenarios and background, with respect to the different training sets we used to fine-tune the network (i.e., People, People & Celebrities, Celebrities only). The plot clearly shows how adding more variance to the training set does not help in the task. This is understandable since when using People only, the training and test set are more similar, which is preferable. In this case, BLUFADE achieves 96% accuracy in less than 4 seconds. On the other hand, when fine-tuning the network only using Celebrities, the training and test set are very different. Still, BLUFADE achieve almost 90% accuracy in less than 5 seconds, which is remarkable.

7.6.3 Limitations

Though BLUFADE achieves good performance, it has some limitations. First, our participants set include few ethnicity, and subjects were tested in just four backgrounds. We added more variance using the celebrities dataset, and the good results suggest BLUFADE would work even with different people. Still, more evaluations need to be conducted. Nonetheless, the four scenarios give us a good idea of how BLUFADE would work in the real world. Second,

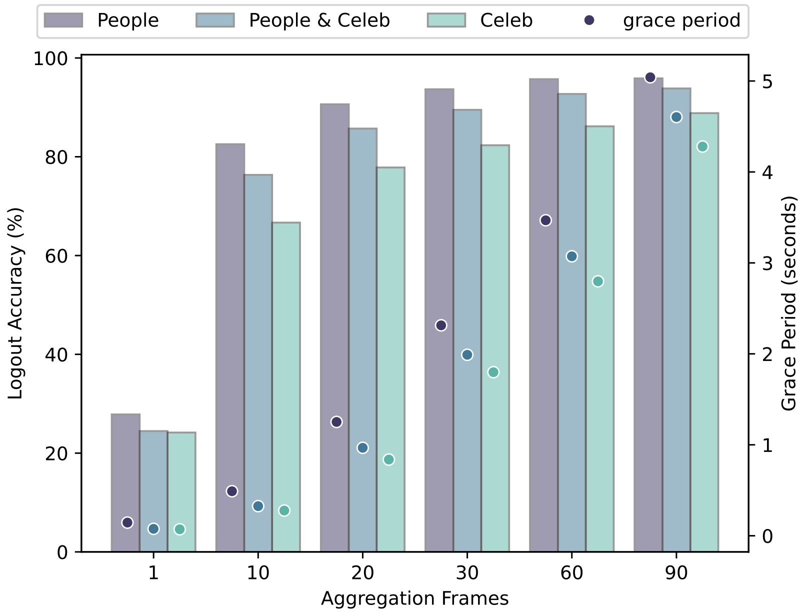


Figure 7.6: Logout accuracy (bars) and grace period (dots) for different aggregation frames and training sets. The accuracy and grace periods are averaged on all the background and scenarios.

participants performed their tasks for ten seconds each. Clearly, longer use of BLUFADE needs to be evaluated. Finally, our evaluation focused on frames containing one person. Since RetinaNet can detect multiple objects in a single image, we assume it can also cope with multiple faces. This needs to be assessed.

7.7 Summary

In this work, we presented BLUFADE, a de-authentication system based on blurred face detection deep learning algorithm. We conducted extensive experiments to select the physical blurring material for BLUFADE, to remove facial traits, ensuring privacy, while allowing face detection by deep learning algorithms. Users' privacy was evaluated through an online survey, demonstrating that a simple anti reflex tape applied to the webcam is sufficient to make a face unrecognizable. By continually detecting the users' blurred faces, BLUFADE automatically de-authenticates them with very high accuracy, i.e., up to 100% in under 3 seconds on simple backgrounds, or 96% within 4 seconds considering also difficult ones. We tested BLUFADE in four application scenarios that represent most of the real-world systems, ranging from laptops to ATMs, with 30 people conducting five different tasks. Our face detection neural network outperforms both commercial and literature state of the art algorithms, demonstrating that fine-tuning can help in the detection of highly blurred objects and faces.

As future work, we plan to better assess the security of the system, testing whether is possible to reconstruct facial traits from physical blurry images. Another possible direction to expand our work is to implement a tracking system instead of performing face detection frame by frame. This way, a higher security level could be achieved without hurting usability.

Chapter 8

Conclusion and Future Work

The expansion of digital technology has transformed human societies, and with it, criminal activity has increased exponentially, accelerating every day. In addition to the public sphere and institutions, these activities affect families and communities in the private sphere. Human actors are at the center of the digital transformation, but from a security perspective, they represent a critical point to be carefully considered to design robust cybersecurity solutions and predict new threats.

In this thesis, we investigated human interactions and cybersecurity, focusing on two main aspects: In Part I we developed new attacks, based on human interaction, against existing and consolidated authentication methods (i.e., PIN pads), and in Part II we proposed new methods leveraging human behavior in multiple contexts to enhance the security of users and organizations.

8.1 Summary of Contributions

In this section, we summarize the contributions of the works presented in this thesis.

8.1.1 In-Security Through Human Interaction Analysis: the PIN Pad Case

In Part I, we investigate several attacks against the security of PIN-based authentication systems, leveraging human interactions. Our goal is to emphasize vulnerabilities raised by the evolution of technology on a consolidated authentication system (i.e., the PIN pad) and propose possible solutions to improve the security.

- *Your PIN Sounds Good: Inter-keystroke Timing Based Attacks on PINs*: In Chapter 2, we investigated several novels inter-keystroke timing-based attacks on PINs. We demonstrated that it is possible to retrieve accurate inter-keystroke timing information from audio feedback. We showed how the user's behavior affects the adversary's ability to guess PINs. In particular, users who type PINs with one finger are more vulnerable to PIN guessing from inter-keystroke timings than users who enter their PIN using at least two fingers, leading to a 34-fold improvement over random guessing. Further, we combined inter-keystroke timing with other information an adversary could retrieve (e.g., thermal trace, knowledge of a key). Our experiments show that inter-keystroke timing significantly improves the performance of the attacks. For example, by combining inter-keystroke timing with a thermal attack, we were able to guess 15% of the PINs at the first attempt, reaching a four-fold improvement in performance compared to thermal attack only.

- *Hand me your PIN: Inferring PINs from Videos of Users Typing with a Covered Hand*: In Chapter 3, we proposed a novel attack aiming to reconstruct PINs entered by victims that cover the typing hand with the other hand. Using deep learning models, we developed a method that predicts what PIN is entered based on the position of the user's hand and their movements while pressing the keys. Our attack was evaluated through an extensive data collection of 58 participants who typed 5,800 5-digit PINs in a simulated ATM. The collected dataset has been made publicly available for the scientific community. We demonstrated that 30% of 5-digit PINs and 41% of 4-digit PINs could be reconstructed by our attack within three attempts, showing that hiding the PIN while typing is not enough to ensure adequate protection. Finally, we assess several possible countermeasures suggesting strategies to mitigate our attack.

- *$\mathcal{P}$inDrop*: *Acoustic Side-Channel Attacks on ATM PIN Pads*: In Chapter 4, we demonstrated how acoustic emanations produced by ATM users entering their PINs could be leveraged to perform a highly accurate side-channel attack (*$\mathcal{P}$inDrop*). Our experiments showed that *$\mathcal{P}$inDrop* could reconstruct up to 94% of 5-digit PINs and 96%

of 4-digit PINs within three attempts. We evaluated the robustness of the attack on two metal PIN pads by combining: the number of attackers, the size of the training set, the microphone's distance from the PIN pad, and the presence of different environmental noises. We showed that the threat posed by $\mathcal{P}inDrop$ is higher compared to the performance of state-of-the-art acoustic side-channel attacks on ATM PIN pads.

8.1.2 Securing the Interaction with Humans

In Part II, we explore how human factors can be used to create non-intrusive, holistic, and secure systems. We focused on three application areas where human interactions play an important role but are still not fully exploited in security contexts: bot detection in social networks, fake emotion detection, and de-authentication.

- *It's a Matter of Style: Detecting Social Bots through Writing Style Consistency*: In Chapter 5, we proposed a novel approach for bot detection leveraging the stylistic consistency of social network posts. More than 12,000 Twitter accounts, including human- and bot-operated ones, were characterized by their writing style. Based on statistical evidence, we identified a set of features capturing the stylistic consistency of posts that allow distinguishing when humans or bots create them. Finally, we evaluated the effectiveness of different ML algorithms based on stylistic consistency features in discerning between human-operated and bot-driven Twitter accounts. Our results showed that the ML models could achieve high performance in this task, achieving an F-measure value up to 98%.

- *Face the Truth: Detection of Spontaneous and Posed Emotional Facial Expressions*: In Chapter 6, we developed a novel framework for the automatic detection of spontaneous and posed emotional facial expressions from clips. For this purpose, we collected a novel dataset that includes a considerable amount of emotional clips (i.e., 1458) for both spontaneous and posed emotions. We validated the dataset through a survey asking 122 participants to rate each clip according to the emotion, genuineness, and intensity of the facial expression perceived. The dataset was publicly released to the research community. We showed that our framework achieves an average accuracy of 84.4% in detecting spontaneous and posed facial expressions in a user-dependent scenario, outperforming humans in the same task (i.e., average accuracy 62.5%).

- BLUFADE*Blurred Face Detection*: In Chapter 7, we proposed a novel secure, usable, and privacy-preserving de-authentication method (BLUFADE) based on blurred face detection. We assessed our approach through extensive experiments on two datasets. The first consists of 20k images of celebrities and backgrounds, blurred with two different materials. The second contains 1,080 enrollment images and 600 videos of 30 subjects interacting with a laptop (both blurred). Both datasets have been made publicly available. BLUFADE showed 95% accuracy in detecting blurred faces, outperforming state-of-the-art methods. Results demonstrate that BLUFADE can effectively de-authenticates users up to 100% accuracy in under 3 seconds while satisfying security, privacy, and usability requirements.

8.2 Future Work

In this section, we introduce future directions of the research presented in this thesis.

8.2.1 In-Security Through Human Interaction Analysis: the PIN Pad Case

Although the use of PINs and passwords are consolidated as authentication methods, several directions can still be taken from a security perspective. In particular, new sources of information could be investigated and used in the attack presented in Chapter 2 to increase its effectiveness. Nevertheless, it would be interesting to develop PIN pads that would allow to mitigate the effectiveness of an attack based on inter-keystroke timing and at the same time maintain high usability of the input devices. The study presented in Chapter 3 also suggests further work in terms of the attack's effectiveness and the study of new countermeasures. In particular, we believe that an interesting direction is to investigate if it is possible to extract the timestamp directly from the video rather than inferring it from the audio feedback. Further, the study could be extended to other PIN pads models (e.g., PoS). Moreover, it would be interesting to explore new strategies for user coverage and their effectiveness in mitigating the attack. Finally, we believe the work presented in Chapter 4 can be extended on the one hand by studying new solutions to increase the recording distance (e.g., parabolic microphones), on the other hand by evaluating the performance of the attachment on new PIN pads (e.g., touch screen PIN pads).

8.2.2 Securing Computer-Human Interaction

The variability and unpredictability of human behavior are characteristics that can also be exploited to increase the security of systems and the users themselves. In Chapter 5 we showed how writing style is effective in distinguishing human and bot accounts on Twitter. One future direction in this study is to extend the dataset to other social networks to validate the generalizability of the proposed approach. Moreover, it would be interesting to evaluate whether stylometry also effectively detects other profiles (e.g., trolls on social media). The human-human behavior distinction is a challenging topic, even for humans themselves. In Chapter 6, we showed how the task of distinguishing genuine from false facial expressions subtends processes that are not easily generalizable. In this context, it would be useful to extend our method as support, via interpretable models, for more insights into the strategies used for expressing fake emotions. In addition, expanding the dataset and developing deep learning models could increase the accuracy of the presented approach. The development of new methods that leverage human behavior to increase security cannot ignore two factors: usability and privacy preservation. Following these directions, in Chapter 7 we developed a novel method for de-authentication of physically blurred faces (BLUFADE). Several directions on the security side can be explored to advance the proposed work. First, assess the resistance of our filters to deblurring algorithms. Second, implementing a tracking system instead of performing face detection frame by frame. Finally, it would be interesting to add an additional layer that would allow recognizing if a lunchtime attack has happened in the grace period, analyzing if there has been a substitution of the user (e.g., different clothes, different haircuts).

Bibliography

[1] *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA, 1997.

[2] Yomna Abdelrahman, Mohamed Khamis, Stefan Schneegass, and Florian Alt. Stay cool! understanding thermal attacks on mobile-based user authentication. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 3751–3763, 2017.

[3] Joyce H Addae, Xu Sun, Dave Towey, and Milena Radenkovic. Exploring user behavioral data for adaptive cybersecurity. *User Modeling and User-Adapted Interaction*, 29(3):701–750, 2019.

[4] Faraz Ahmed and Muhammad Abulaish. A generic statistical approach for spam detection in online social networks. *Computer Communications*, 36(10-11):1120–1129, 2013.

[5] Zara Ambadar, Jonathan W Schooler, and Jeffrey F Cohn. Deciphering the enigmatic face: The importance of facial dynamics in interpreting subtle facial expressions. *Psychological science*, 16(5):403–410, 2005.

[6] S Abhishek Anand and Nitesh Saxena. Keyboard emanations in remote voice calls: Password leakage and noise (less) masking defenses. In *ACM CODASPY*, 2018.

[7] Richard John Andrew. Evolution of facial expression. *Science*, 142(3595):1034–1041, 1963.

[8] Shlomo Argamon, Casey Whitelaw, Paul Chase, Sobhan Raj Hota, Navendu Garg, and Shlomo Levitan. Stylistic text classification us-

ing functional lexical features. *Journal of the American Society for Information Science and Technology*, 58(6):802–822, 2007.

[9] Shaina Ashraf, Omer Javed, Muhammad Adeel, Haider Iqbal, and Rao Muhammad Adeel Nawab. Bots and gender prediction using language independent stylometry-based approach. In *Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 9-12, 2019*, 2019.

[10] Dmitri Asonov and Rakesh Agrawal. Keyboard acoustic emanations. In *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004*, pages 3–11. IEEE, 2004.

[11] S Ayeswarya and Jasmine Norman. A survey on different continuous authentication systems. *International Journal of Biometrics*, 11(1):67–99, 2019.

[12] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.

[13] Erica R Bailey, Sandra C Matz, Wu Youyou, and Sheena S Iyengar. Authentic self-expression on social media is associated with greater subjective well-being. *Nature communications*, 11(1):1–9, 2020.

[14] Andrzej Bakowski, Leszek Radziszewski, Vladimir Dekỳš, and Paweł Świetlik. Frequency analysis of urban traffic noise. In *2019 20th International Carpathian Control Conference (ICCC)*, pages 1–6. IEEE, 2019.

[15] Kiran Balagani, Matteo Cardaioli, Mauro Conti, Paolo Gasti, Martin Georgiev, Tristan Gurtler, Daniele Lain, Charissa Miller, Kendall Molas, Nikita Samarin, et al. Pilot: Password and pin information leakage from obfuscated typing videos. *Journal of Computer Security*, 27(4):405–425, 2019.

[16] Kiran S Balagani, Mauro Conti, Paolo Gasti, Martin Georgiev, Tristan Gurtler, Daniele Lain, Charissa Miller, Kendall Molas, Nikita Samarin, Eugen Saraci, et al. Silk-tv: Secret information leakage from keystroke timing videos. In *European Symposium on Research in Computer Security*, pages 263–280. Springer, 2018.

[17] Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency. Openface: an open source facial behavior analysis toolkit. In *2016 IEEE*

Winter Conference on Applications of Computer Vision (WACV), pages 1–10. IEEE, 2016.

[18] Davide Balzarotti, Marco Cova, and Giovanni Vigna. Clearshot: Eavesdropping on keyboard input from video. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 170–183. IEEE, 2008.

[19] Salil P Banerjee and Damon L Woodard. Biometric authentication and identification using keystroke dynamics: A survey. *Journal of Pattern Recognition Research*, 7(1):116–139, 2012.

[20] Ioana-Alexandra Bara, Carol J Fung, and Thang Dinh. Enhancing twitter spam accounts discovery using cross-account pattern mining. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 491–496. IEEE, 2015.

[21] Lisa Feldman Barrett, Ralph Adolphs, Stacy Marsella, Aleix M Martinez, and Seth D Pollak. Emotional expressions reconsidered: Challenges to inferring emotion from human facial movements. *Psychological science in the public interest*, 20(1):1–68, 2019.

[22] Marian Stewart Bartlett, Gwen Littlewort, Mark Frank, Claudia Lainscsek, Ian Fasel, and Javier Movellan. Recognizing facial expression: machine learning and application to spontaneous behavior. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 568–573. IEEE, 2005.

[23] Marian Stewart Bartlett, Gwen Littlewort, Mark G Frank, Claudia Lainscsek, Ian R Fasel, Javier R Movellan, et al. Automatic recognition of facial actions in spontaneous expressions. *J. Multim.*, 1(6):22–35, 2006.

[24] Marian Stewart Bartlett, Gwen C Littlewort, Mark G Frank, and Kang Lee. Automatic decoding of facial movements reveals deceptive pain expressions. *Current Biology*, 24(7):738–743, 2014.

[25] Bernardo Bátiz-Lazo and Robert Reid. The development of cash-dispensing technology in the uk. *IEEE Annals of the History of Computing*, 33(3):32–45, 2011.

[26] Yigael Berger, Avishai Wool, and Arie Yeredor. Dictionary attacks using keyboard acoustic emanations. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 245–254, 2006.

[27] David M Beskow and Kathleen M Carley. Bot-hunter: A tiered approach to detecting & characterizing automated activity on twitter. In *SBP-BRiMS: International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*, volume 8, 2018.

[28] Alessandro Bessi and Emilio Ferrara. Social bots distort the 2016 us presidential election online discussion. *First Monday*, 21(11-7), 2016.

[29] Shivam Bhasin, Anupam Chattopadhyay, Annelie Heuser, Dirmanto Jap, Stjepan Picek, and Ritu Ranjan Shrivastwa. Mind the portability: A warriors guide through realistic profiled side-channel analysis. In *27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, California, USA, February 23-26, 2020*. The Internet Society, 2020.

[30] Farid Binbeshr, ML Mat Kiah, Lip Yee Por, and Aws Alaa Zaidan. A systematic review of pin-entry methods resistant to shoulder-surfing attacks. *computers & security*, page 102116, 2020.

[31] Chris M. Bishop. Training with noise is equivalent to tikhonov regularization. *Neural Comput.*, 7(1):108–116, January 1995.

[32] Mike Bond, Omar Choudary, Steven J Murdoch, Sergei Skorobogatov, and Ross Anderson. Chip and skim: cloning emv cards with the preplay attack. In *2014 IEEE Symposium on Security and Privacy*, pages 49–64. IEEE, 2014.

[33] Joseph Bonneau, Sören Preibusch, and Ross Anderson. A birthday present every eleven wallets? the security of customer-chosen banking pins. In *International Conference on Financial Cryptography and Data Security*, pages 25–40. Springer, 2012.

[34] Guido Borghi, Marco Venturelli, Roberto Vezzani, and Rita Cucchiara. Poseidon: Face-from-depth for driver pose estimation. In *Proceedings of the IEEE CVPR*, pages 4661–4670, 2017.

[35] Antoine Bouveret. *Cyber risk for the financial sector: A framework for quantitative assessment.* International Monetary Fund, 2018.

[36] Vasiliy Boychuk, Kirill Sukharev, Daniil Voloshin, and Vladislav Karbovskii. An exploratory sentiment and facial expressions analysis of data from photo-sharing on social media: The case of football violence. *Procedia computer science*, 80:398–406, 2016.

[37] Matthew Brocker and Stephen Checkoway. iseeyou: Disabling the macbook webcam indicator led. In *23rd USENIX*, pages 337–352, 2014.

[38] Roberto Brunelli and Tomaso Poggio. Face recognition: Features versus templates. *IEEE transactions on pattern analysis and machine intelligence*, 15(10):1042–1052, 1993.

[39] Stephen Butterworth. On the theory of filter amplifiers. *Wireless Engineer*, 7(6):536–541, 1930.

[40] Qiang Cao, Michael Sirivianos, Xiaowei Yang, and Tiago Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *Presented as part of the 9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12)*, pages 197–210, 2012.

[41] Matteo Cardaioli, Stefano Cecconello, Mauro Conti, Simone Milani, Stjepan Pjcek, and Eugen Saraci. Hand me your PIN! inferring ATM PINs of users typing with a covered hand. In *31st USENIX Security Symposium (USENIX Security 22)*, Boston, MA, August 2022. USENIX Association.

[42] Matteo Cardaioli, Mauro Conti, Kiran Balagani, and Paolo Gasti. Your pin sounds good! augmentation of pin guessing strategies via audio leakage. In *European Symposium on Research in Computer Security*, pages 720–735. Springer, 2020.

[43] Matteo Cardaioli, Mauro Conti, Andrea Di Sorbo, Enrico Fabrizio, Sonia Laudanna, and Corrado A Visaggio. It'sa matter of style: Detecting social bots through writing style consistency. In *2021 International Conference on Computer Communications and Networks (ICCCN)*, pages 1–9. IEEE, 2021.

[44] Matteo Cardaioli, Mauro Conti, Gene Tsudik, and Pier Paolo Tricomi. Privacy-friendly de-authentication with blufade: Blurred face detection. In *2 20th International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2022.

[45] Sandra Carvalho, Jorge Leite, Santiago Galdo-Álvarez, and Oscar F Gonçalves. The emotional movie database (emdb): A self-report and psychophysiological study. *Applied psychophysiology and biofeedback*, 37(4):279–294, 2012.

[46] Stefano Cecconello, Alberto Compagno, Mauro Conti, Daniele Lain, and Gene Tsudik. Skype & type: Keyboard eavesdropping in voice-over-ip. *ACM Transactions on Privacy and Security (TOPS)*, 22(4):1–34, 2019.

[47] Zi Chu, Steven Gianvecchio, Haining Wang, and Sushil Jajodia. Who is tweeting on twitter: human, bot, or cyborg? In *Proceedings of the 26th annual computer security applications conference*, pages 21–30, 2010.

[48] Zi Chu, Steven Gianvecchio, Haining Wang, and Sushil Jajodia. Detecting automation of twitter accounts: Are you a human, bot, or cyborg? *IEEE Transactions on Dependable and Secure Computing*, 9(6):811–824, 2012.

[49] Norman Cliff. Answering ordinal questions with ordinal data using ordinal statistics. *Multivariate Behavioral Research*, 31(3):331–350, 1996.

[50] HP Development Company. Hp presence aware. https://tinyurl.com/HPunattended, 2020. Accessed: October, 2021.

[51] Mauro Conti, Giulio Lovisotto, Ivan Martinovic, and Gene Tsudik. Fadewich: fast deauthentication over the wireless channel. In *2017 IEEE 37th ICDCS*, pages 2294–2301. IEEE, 2017.

[52] Mauro Conti and Pier Paolo Tricomi. Pvp: Profiling versus player! exploiting gaming data for player recognition. In *International Conference on Information Security*, pages 393–408. Springer, 2020.

[53] Mauro Conti, Pier Paolo Tricomi, and Gene Tsudik. De-auth of the blue! transparent de-authentication using bluetooth low energy beacon. In *ESORICS*, pages 277–294. Springer, 2020.

[54] Daniel T Cordaro, Rui Sun, Dacher Keltner, Shanmukh Kamble, Niranjan Huddar, and Galen McNeil. Universals and cultural variations in 22 emotional expressions across five cultures. *Emotion*, 18(1):75, 2018.

[55] Mark D Corner and Brian D Noble. Zero-interaction authentication. In *Proceedings of the 8th annual international conference on Mobile computing and networking*, pages 1–11. ACM, 2002.

[56] National Cash Register (NCR Corporation). The rise of emv and what it means for the magnetic stripe. https://www.ncr.com/blogs/payments/emv-magnetic-stripe, March 2021. [Online; accessed 7-June-2021].

[57] Crawford. Artificial Intelligence's White Guy Problem, 2016.

[58] Stefano Cresci. A decade of social bot detection. *Communications of the ACM*, 63(10):72–83, 2020.

[59] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. Fame for sale: Efficient detection of fake twitter followers. *Decision Support Systems*, 80:56–71, 2015.

[60] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. Dna-inspired online behavioral modeling and its application to spambot detection. *IEEE Intelligent Systems*, 31(5):58–64, 2016.

[61] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race. In *Proceedings of the 26th international conference on world wide web companion*, pages 963–972, 2017.

[62] Stefano Cresci, Marinella Petrocchi, Angelo Spognardi, and Stefano Tognazzi. Better safe than sorry: An adversarial approach to improve social bot detection. In *Proceedings of the 10th ACM Conference on Web Science*, pages 47–56, 2019.

[63] Carlos Crivelli and Alan J Fridlund. Inside-out: From basic emotions theory to the behavioral ecology view. *Journal of Nonverbal Behavior*, 43(2):161–194, 2019.

[64] David Crouse, Hu Han, Deepak Chandra, Brandon Barbello, and Anil K. Jain. Continuous authentication of mobile user: Fusion of face image and inertial measurement unit data. In *2015 ICB*, pages 135–142, 2015.

[65] Robertas Damaševičius, Rytis Maskeliūnas, Algimantas Venčkauskas, and Marcin Woźniak. Smartphone user identity verification using gait characteristics. *Symmetry*, 8(10):100, 2016.

[66] George Danezis and Prateek Mittal. Sybilinfer: Detecting sybil nodes using social networks. In *Ndss*, pages 1–15. San Diego, CA, 2009.

[67] Charles Darwin. The expression of emotions in animals and man. *London: Murray*, 11:1872, 1872.

[68] Clayton Allen Davis, Onur Varol, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. Botornot: A system to evaluate social bots. In *Proceedings of the 25th international conference companion on world wide web*, pages 273–274, 2016.

[69] Amy Dawel, Luke Wright, Jessica Irons, Rachael Dumbleton, Romina Palermo, Richard O'Kearney, and Elinor McKone. Perceived emotion genuineness: normative ratings for popular facial expression stimuli and the development of perceived-as-genuine and perceived-as-fake sets. *Behavior Research Methods*, 49(4):1539–1562, 2017.

[70] Kanjar De and V Masilamani. Image sharpness measure for blurred images in frequency domain. *Procedia Engineering*, 64:149–158, 2013.

[71] Munmun De Choudhury and Scott Counts. The nature of emotional expression in social media: measurement, inference and utility. *Human Computer Interaction Consortium (HCIC)*, 2012.

[72] Gerson de Souza Faria and Hae Yong Kim. Differential audio analysis: a new side-channel attack on pin pads. *International Journal of Information Security*, 18(1):73–84, 2019.

[73] Belinda L Del Gaudio, Claudio Porzio, Gabriele Sampagnaro, and Vincenzo Verdoliva. How do mobile, internet and ict diffusion affect the banking industry? an empirical analysis. *European Management Journal*, 2020.

[74] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.

[75] Bella M DePaulo, Deborah A Kashy, Susan E Kirkendol, Melissa M Wyer, and Jennifer A Epstein. Lying in everyday life. *Journal of personality and social psychology*, 70(5):979, 1996.

[76] John P Dickerson, Vadim Kagan, and VS Subrahmanian. Using sentiment to detect bots on twitter: Are humans more opinionated than bots? In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, pages 620–627. IEEE, 2014.

[77] Lijun Ding and Ardeshir Goshtasby. On the canny edge detector. *Pattern Recognition*, 34(3):721–725, 2001.

[78] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[79] Damien Dupré, Eva G Krumhuber, Dennis Küster, and Gary J McKeown. A performance comparison of eight commercially available automatic classifiers for facial affect recognition. *Plos one*, 15(4):e0231968, 2020.

[80] Juan I Durán, Rainer Reisenzein, and José-Miguel Fernández-Dols. Coherence between emotions and facial expressions. *The science of facial expression*, pages 107–129, 2017.

[81] Simon Eberz, K Rasmussen, Vincent Lenders, and Ivan Martinovic. Preventing lunchtime attacks: Fighting insider threats with eye movement biometrics. In *22th NDSS 2015*. Internet Society, 2015.

[82] Malin Eiband, Mohamed Khamis, Emanuel Von Zezschwitz, Heinrich Hussmann, and Florian Alt. Understanding shoulder surfing in the wild: Stories from users and observers. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 4254–4265, 2017.

[83] Paul Ekman. Universals and cultural differences in facial expressions of emotion. 1971. *URL: https://www. paulekman. com/wp-content/uploads/2013/07/Universals-And-Cultural-Differences-In-Facial-Expressions-Of. pdf (2015-07-15)*, 1972.

[84] Paul Ekman. Are there basic emotions? 1992.

[85] Paul Ekman. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200, 1992.

[86] Paul Ekman. Darwin, deception, and facial expression. *Annals of the new York Academy of sciences*, 1000(1):205–221, 2003.

[87] Paul Ekman and Daniel Cordaro. What is meant by calling emotions basic. *Emotion review*, 3(4):364–370, 2011.

[88] Paul Ekman, Wallace Friesen, and Joseph Hager. Facs investigator's guide.(2002). 2002.

[89] Paul Ekman and Wallace V Friesen. A new pan-cultural facial expression of emotion. *Motivation and emotion*, 10(2):159–168, 1986.

[90] Paul Ekman and Wallace V Friesen. *Unmasking the face: A guide to recognizing emotions from facial clues.* Ishk, 2003.

[91] Paul Ekman, Wallace V Friesen, and JC Hager. Facial action coding system: manual. palo alto, 1978.

[92] Paul Ekman and Dacher Keltner. Universal facial expressions of emotion. *Segerstrale U, P. Molnar P, eds. Nonverbal communication: Where nature meets culture*, 27:46, 1997.

[93] Paul Ekman and Erika Rosenberg. *What the Face Reveals: Basic and Applied Studies of Spontaneous Expression Using the Facial Action Encoding System (FACS).* Oxford University Press, 2005.

[94] Paul Ekman, E Richard Sorenson, and Wallace V Friesen. Pan-cultural elements in facial displays of emotion. *Science*, 164(3875):86–88, 1969.

[95] Adam EM Eltorai, Syed S Naqvi, Soha Ghanian, Craig P Eberson, Arnold-Peter C Weiss, Christopher T Born, and Alan H Daniels. Readability of invasive procedure consent forms. *Clinical and translational science*, 8(6):830–833, 2015.

[96] Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. The rise of social bots. *Communications of the ACM*, 59(7):96–104, 2016.

[97] FINECO. Le carte fineco, 2019.

[98] Dinei Florencio and Cormac Herley. A large-scale study of web password habits. In *ACM WWW*, 2007.

[99] Mara Fölster, Ursula Hess, and Katja Werheid. Facial age affects emotional expression decoding. *Frontiers in psychology*, 5:30, 2014.

[100] Mark G Frank and Janine Stennett. The forced-choice paradigm and the perception of facial expressions of emotion. *Journal of personality and social psychology*, 80(1):75, 2001.

[101] Lex Fridman, Steven Weber, Rachel Greenstadt, and Moshe Kam. Active authentication on mobile devices via stylometry, application usage, web browsing, and gps location. *IEEE Systems Journal*, 11(2):513–521, 2016.

[102] Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, 32(200):675–701, 1937.

[103] Edward Fry. A readability formula that saves time. *Journal of reading*, 11(7):513–578, 1968.

[104] Michael Gamon. Linguistic correlates of style: authorship classification with deep linguistic analysis features. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 611–617, 2004.

[105] Daniel Genkin, Adi Shamir, and Eran Tromer. Acoustic cryptanalysis. *J. Cryptol.*, 30(2):392–443, April 2017.

[106] Yulia Golland, Adam Hakim, Tali Aloni, Stacey Schaefer, and Nava Levit-Binnun. Affect dynamics of facial emg during continuous emotional experiences. *Biological psychology*, 139:47–58, 2018.

[107] Jack William Grieve. *Quantitative authorship attribution: A history and an evaluation of techniques.* PhD thesis, Department of Linguistics-Simon Fraser University, 2005.

[108] Christian Grimme, Mike Preuss, Lena Adam, and Heike Trautmann. Social bots: Human-like by means of human control? *Big data*, 5(4):279–293, 2017.

[109] Robert J. Grissom and John J. Kim. *Effect sizes for research: A broad practical approach.* Lawrence Earlbaum Associates, 2nd edition edition, 2005.

[110] Charline Grossard, Laurence Chaby, Stéphanie Hun, Hugues Pellerin, Jérémy Bourgeois, Arnaud Dapogny, Huaxiong Ding, Sylvie Serret, Pierre Foulon, Mohamed Chetouani, et al. Children facial expression production: influence of age, gender, emotion subtype, elicitation condition and culture. *Frontiers in psychology*, 9:446, 2018.

[111] Ivan Gruber, Miroslav Hlaváč, Miloš Železnỳ, and Alexey Karpov. Facing face recognition with resnet: Round one. In *International Conference on Interactive Collaborative Robotics*, pages 67–74. Springer, 2017.

[112] Robert Gunning. The fog index after twenty years. *Journal of Business Communication*, 6(2):3–13, 1969.

[113] Guodong Guo and Na Zhang. A survey on deep learning based face recognition. *Computer vision and image understanding*, 189:102805, 2019.

[114] Hui Guo, Xiao-Hui Zhang, Jun Liang, and Wen-Jing Yan. The dynamic features of lip corners in genuine and posed smiles. *Frontiers in psychology*, 9:202, 2018.

[115] Brij B Gupta and Shaifali Narayan. A survey on contactless smart cards and payment system: Technologies, policies, attacks and countermeasures. *Journal of Global Information Management (JGIM)*, 28(4):135–159, 2020.

[116] Tzipora Halevi and Nitesh Saxena. A closer look at keyboard acoustic emanations: random passwords, typing styles and decoding techniques. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, pages 89–90, 2012.

[117] Tzipora Halevi and Nitesh Saxena. Keyboard acoustic side channel attacks: exploring realistic and security-sensitive scenarios. *International Journal of Information Security*, 14(5):443–456, 2015.

[118] SL Happy, Priyadarshi Patnaik, Aurobinda Routray, and Rajlakshmi Guha. The indian spontaneous expression database for emotion recognition. *IEEE Transactions on Affective Computing*, 8(1):131–142, 2015.

[119] Julian Hattem. Fbi director: Cover up your webcam. https://thehill.com/policy/national-security/295933-fbi-director-cover-up-your-webcam, 2016. Accessed: September, 2021.

[120] Catherine J Hayes, Richard J Stevenson, and Max Coltheart. The processing of emotion in patients with huntington's disease: variability and differential deficits in disgust. *Cognitive and behavioral neurology*, 22(4):249–257, 2009.

[121] Catherine J Hayes, Richard J Stevenson, and Max Coltheart. Production of spontaneous and posed facial expressions in patients with huntington's disease: Impaired communication of disgust. *Cognition and Emotion*, 23(1):118–134, 2009.

[122] Luis Hernández-Álvarez, José María de Fuentes, Lorena González-Manzano, and Luis Hernández Encinas. Privacy-preserving sensor-based continuous authentication and user profiling: a review. *Sensors*, 21(1):92, 2021.

[123] Jonathan Herz and Abdelghani Bellaachia. The authorship of audacity: Data mining and stylometric analysis of barack obama speeches. In *Proceedings of the International Conference on Data Science (ICDATA)*, page 1. The Steering Committee of The World Congress in Computer Science, Computer . . . , 2014.

[124] Ursula Hess and Robert E Kleck. Differentiating emotion elicited and deliberate emotional facial expressions. *European Journal of Social Psychology*, 20(5):369–385, 1990.

[125] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[126] Christof Holberg, Cathrin Maier, Stefanie Steinhäuser, and Ingrid Rudzki-Janson. Inter-individual variability of the facial morphology during conscious smiling. *Journal of Orofacial Orthopedics/Fortschritte der Kieferorthopädie*, 67(4):234–243, 2006.

[127] Ayanna Howard, Cha Zhang, and Eric Horvitz. Addressing bias in machine learning algorithms: A pilot study on emotion recognition for intelligent systems. In *2017 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*, pages 1–7. IEEE, 2017.

[128] Otto Huhta, Prakash Shrestha, Swapnil Udar, Mika Juuti, Nitesh Saxena, and N Asokan. Pitfalls in designing zero-effort deauthentication: Opportunistic human observation attacks. In *NDSS*, 02 2016.

[129] Xuan-Phung Huynh and Yong-Guk Kim. Discrimination between genuine versus fake emotion using long-short term memory with parametric bias and facial landmarks. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[130] Nautilus Hyosung. cmax7600ta installation manual. http://www.tetralink.com/core/media/media.nl/id.46617/c.4970910/.f?h=d919934a85943438b8fe, 2015. [Online; accessed 30-December-2020].

[131] Rodrigo Augusto Igawa, Sylvio Barbon Jr, Kátia Cristina Silva Paulo, Guilherme Sakaji Kido, Rodrigo Capobianco Guido, Mario Lemes Proença Júnior, and Ivan Nunes da Silva. Account classification in online social networks with lbca and wavelets. *Information Sciences*, 332:72–83, 2016.

[132] M. Islam, M. Hossain, R. ul Islam, and K. Andersson. Static hand gesture recognition using convolutional neural network with data augmentation. In *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, pages 324–329, jun 2019.

[133] ISO. Financial services – personal identification number (pin) management and security – part 1: Basic principles and requirements for pins in card-based systems, 2017.

[134] Rachael E Jack, Oliver GB Garrod, Hui Yu, Roberto Caldara, and Philippe G Schyns. Facial expressions of emotion are not culturally universal. *Proceedings of the National Academy of Sciences*, 109(19):7241–7244, 2012.

[135] Rachael E Jack and Philippe G Schyns. The human face as a dynamic tool for social communication. *Current Biology*, 25(14):R621–R634, 2015.

[136] Rachael E Jack, Wei Sun, Ioannis Delis, Oliver GB Garrod, and Philippe G Schyns. Four not six: Revealing culturally common facial expressions of emotion. *Journal of Experimental Psychology: General*, 145(6):708, 2016.

[137] Rabia Jafri and Hamid R Arabnia. A survey of face recognition techniques. *journal of information processing systems*, 5(2):41–68, 2009.

[138] Shan Jia, Shuo Wang, Chuanbo Hu, Paula Jo Webster, and Xin Li. Detection of genuine and posed facial expressions of emotion: Databases and methods. *Frontiers in Psychology*, 11:3818, 2020.

[139] Fredrik Johansson. Supervised classification of twitter accounts based on textual content of tweets. In *CLEF (Working Notes)*, 2019.

[140] Lucy Johnston, Lynden Miles, and C Neil Macrae. Why are you smiling at me? social functions of enjoyment and non-enjoyment smiles. *British Journal of Social Psychology*, 49(1):107–127, 2010.

[141] Louise Marie Jupe and David Adam Keatley. Airport artificial intelligence can detect deception: or am i lying? *Security Journal*, 33(4):622–635, 2020.

[142] Abdullah Talha Kabakus and Resul Kara. A survey of spam detection methods on twitter. *International Journal of Advanced Computer Science and Applications*, 8(3):29–38, 2017.

[143] Tyler Kaczmarek, Ercan Ozturk, and Gene Tsudik. Assentication: User de-authentication and lunchtime attack mitigation with seated posture biometric. In *International Conference on Applied Cryptography and Network Security*, pages 616–633. Springer, 2018.

[144] Tyler Kaczmarek, Ercan Ozturk, and Gene Tsudik. Thermanator: Thermal residue-based post factum attacks on keyboard data entry. In *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, pages 586–593, 2019.

[145] Takeo Kanade. Picture processing system by computer complex and recognition of human faces. 1974.

[146] Bong-Nam Kang, Yonghyun Kim, and Daijin Kim. Pairwise relational networks for face recognition. In *Proceedings of ECCV*, pages 628–645, 2018.

[147] Nikhil Ketkar. Introduction to keras. In *Deep learning with Python*, pages 97–111. Springer, 2017.

[148] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *European Conference on Computer Vision*, pages 158–171. Springer, 2012.

[149] Eugenia Kim, De'Aira Bryant, Deepak Srikanth, and Ayanna Howard. Age bias in emotion detection: An analysis of facial emotion recognition performance on young, middle-aged, and older adults. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 638–644, 2021.

[150] Jaehun Kim, Stjepan Picek, Annelie Heuser, Shivam Bhasin, and Alan Hanjalic. Make some noise. unleashing the power of convolutional neural networks for profiled side-channel analysis. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 148–179, 2019.

[151] Jaewoo Kim, Yui Ha, Seungche Kang, Hongjun Lim, and Meeyoung Cha. Detecting multiclass emotions from labeled movie scripts. In *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 590–594. IEEE, 2018.

[152] Brendan F Klare, Mark J Burge, Joshua C Klontz, Richard W Vorder Bruegge, and Anil K Jain. Face recognition performance: Role of demographic information. *IEEE Transactions on Information Forensics and Security*, 7(6):1789–1801, 2012.

[153] Paul C Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Annual International Cryptology Conference*, pages 104–113. Springer, 1996.

[154] Eva Krumhuber, Antony SR Manstead, Darren Cosker, Dave Marshall, and Paul L Rosin. Effects of dynamic attributes of smiles in human and synthetic faces: A simulated job interview setting. *Journal of Nonverbal Behavior*, 33(1):1–15, 2009.

[155] Eva Krumhuber, Antony SR Manstead, and Arvid Kappas. Temporal aspects of facial displays in person and expression perception: The effects of smile dynamics, head-tilt, and gender. *Journal of Nonverbal Behavior*, 31(1):39–56, 2007.

[156] Eva G Krumhuber, Dennis Küster, Shushi Namba, Datin Shah, and Manuel G Calvo. Emotion recognition from posed and spontaneous dynamic expressions: Human observers versus machine analysis. *Emotion*, 2019.

[157] Eva G Krumhuber and Antony SR Manstead. Can duchenne smiles be feigned? new evidence on felt and false smiles. *Emotion*, 9(6):807, 2009.

[158] Manu Kumar, Tal Garfinkel, Dan Boneh, and Terry Winograd. Reducing shoulder-surfing by using gaze-based password entry. In *Proceedings of the 3rd symposium on Usable privacy and security*, pages 13–19. ACM, 2007.

[159] Antu Mary Kuruvilla and Saira Varghese. A detection system to counter identity deception in social media applications. In *2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]*, pages 1–5. IEEE, 2015.

[160] Taekyoung Kwon and Jin Hong. Analysis and improvement of a pin-entry method resilient to shoulder-surfing and recording attacks. *Ieee transactions on information forensics and security*, 10(2):278–292, 2015.

[161] Rachid Ait Maalem Lahcen, Bruce Caulkins, Ram Mohapatra, and Manish Kumar. Review and insight on the behavioral aspects of cybersecurity. *Cybersecurity*, 3(1):1–18, 2020.

[162] K. Lai and S. N. Yanushkevich. Cnn+rnn depth and skeleton based dynamic hand gesture recognition. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 3451–3456, 2018.

[163] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[164] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[165] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 2014.

[166] Jian Liu, Yan Wang, Gorkem Kar, Yingying Chen, Jie Yang, and Marco Gruteser. Snooping keystrokes with mm-level audio ranging on a single phone. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 142–154, 2015.

[167] Ximing Liu, Yingjiu Li, Robert H Deng, Bing Chang, and Shujun Li. When human cognitive modeling meets pins: User-independent inter-keystroke timing attacks. *Computers & Security*, 80:90–107, 2019.

[168] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of ICCV*, December 2015.

[169] Beth Logan et al. Mel frequency cepstral coefficients for music modeling. In *Ismir*, volume 270, pages 1–11, 2000.

[170] Rocío López-Anguita, Arturo Montejo-Ráez, and Manuel Carlos Díaz-Galiano. Complexity measures and POS n-grams for author identification in several languages: SINAI at pan@clef 2018. In *Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018*, 2018.

[171] Wenze Lu, Cindy Sing Bik Ngai, and Lu Yang. The importance of genuineness in public engagement—an exploratory study of pediatric communication on social media in china. *International Journal of Environmental Research and Public Health*, 17(19):7078, 2020.

[172] Xiaojun Lu, Yue Yang, Weilin Zhang, Qi Wang, and Yang Wang. Face verification with multi-task and multi-scale feature fusion. *entropy*, 19(5):228, 2017.

[173] Timothy J Luke. Lessons from pinocchio: Cues to deception may be highly exaggerated. *Perspectives on Psychological Science*, 14(4):646–671, 2019.

[174] Dominique Machuletz, Henrik Sendt, Stefan Laube, and Rainer Böhme. Users protect their privacy if they can: Determinants of webcam covering behavior. In *Proceedings of EuroSEC'16*, 2016.

[175] Upal Mahbub, Vishal M Patel, Deepak Chandra, Brandon Barbello, and Rama Chellappa. Partial face detection for continuous authentication. In *2016 IEEE ICIP*, pages 2991–2995. IEEE, 2016.

[176] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer, December 2006. ISBN 0-387-30857-1, http://www.dpabook.org/.

[177] Shrirang Mare, Andrés Molina Markham, Cory Cornelius, Ronald Peterson, and David Kotz. Zebra: Zero-effort bilateral recurring authentication. In *2014 IEEE Symposium on Security and Privacy*, pages 705–720. IEEE, 2014.

[178] Philip Marquardt, Arunabh Verma, Henry Carter, and Patrick Traynor. (sp) iphone: Decoding vibrations from nearby keyboards using mobile phone accelerometers. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 551–562. ACM, 2011.

[179] Diogo Marques, Ildar Muslukhov, Tiago Guerreiro, Luís Carriço, and Konstantin Beznosov. Snooping on mobile phones: Prevalence and trends. In *Twelfth SOUPS 2016)*, 2016.

[180] Zdenek Martinasek, Vlastimil Clupek, and Krisztina Trasy. Acoustic attack on keyboard using spectrogram and neural network. In *2015 38th International Conference on Telecommunications and Signal Processing (TSP)*, pages 637–641. IEEE, 2015.

[181] Iacopo Masi, Yue Wu, Tal Hassner, and Prem Natarajan. Deep face recognition: A survey. In *2018 31st SIBGRAPI*. IEEE, 2018.

[182] G Harry Mc Laughlin. Smog grading-a new readability formula. *Journal of reading*, 12(8):639–646, 1969.

[183] Douglas R McCallum and James L Peterson. Computer-based readability indexes. In *Proceedings of the ACM'82 Conference*, pages 44–48, 1982.

[184] Mary L McHugh. The chi-square test of independence. *Biochemia medica*, 23(2):143–149, 2013.

[185] Patrick E McKnight and Julius Najab. Mann-whitney u test. *The Corsini encyclopedia of psychology*, pages 1–1, 2010.

[186] Marc Mehu, Marcello Mortillaro, Tanja Bänziger, and Klaus R Scherer. Reliable facial muscle activation enhances recognizability and credibility of emotional expression. *Emotion*, 12(4):701, 2012.

[187] Gabriele H Meiselwitz. *Social computing and social media*. Springer, 2016.

[188] Alessio Miolla, Merylin Monaro, and Cristina. Scarpazza. Choose your face: a comprehensive review of dataset of emotions conveyed by faces. *submitted*, 2021.

[189] Abbas Moallem. *Human-Computer Interaction and cybersecurity handbook*. CRC Press, 2018.

[190] John Monaco. Sok: Keylogging side channels. In *IEEE S&P*, 2018.

[191] Robert Morris and Ken Thompson. Password security: A case history. *Communications of the ACM*, 22(11):594–597, 1979.

[192] Michael T Motley and Carl T Camden. Facial expression of emotion: A comparison of posed expressions versus spontaneous expressions in an interpersonal communication setting. *Western Journal of Communication (includes Communication Reports)*, 52(1):1–22, 1988.

[193] Keaton Mowery, Sarah Meiklejohn, and Stefan Savage. Heat of the moment: Characterizing the efficacy of thermal camera-based attacks. In *Proceedings of the 5th USENIX conference on Offensive technologies*, pages 6–6, 2011.

[194] Hiroko Nakayama. Changes in the affect of infants before and after episodes of crying. *Infant Behavior and Development*, 36(4):507–512, 2013.

[195] NationalCash Systems. ATM Statistics.

[196] Tempestt Neal, Kalaivani Sundararajan, Aneez Fatima, Yiming Yan, Yingfei Xiang, and Damon Woodard. Surveying stylometry techniques and applications. *ACM Computing Surveys (CSUR)*, 50(6):1–36, 2017.

[197] Peter Nemenyi. Distribution-free multiple comparisons. In *Biometrics*, volume 18, page 263. International Biometric Soc 1441 I ST, NW, SUITE 700, WASHINGTON, DC 20005-2210, 1962.

[198] Mariam Orabi, Djedjiga Mouheb, Zaher Al Aghbari, and Ibrahim Kamel. Detection of bots in social media: a systematic review. *Information Processing & Management*, 57(4):102250, 2020.

[199] Farhan Nurdiatama Pakaya, Muhammad Okky Ibrohim, and Indra Budi. Malicious account detection on twitter based on tweet account features using machine learning. In *2019 Fourth International Conference on Informatics and Computing (ICIC)*, pages 1–5. IEEE, 2019.

[200] Sourav Panda, Yuanzhen Liu, Gerhard Petrus Hancke, and Umair Mujtaba Qureshi. Behavioral acoustic emanations: Attack and verification of pin entry using keypress sounds. *Sensors*, 20(11):3015, 2020.

[201] Marietta Papadatou-Pastou, Eleni Ntolka, Judith Schmitz, Maryanne Martin, Marcus R Munafò, Sebastian Ocklenburg, and Silvia Paracchini. Human handedness: A meta-analysis. *Psychological Bulletin*, April 2020.

[202] Ge Peng, Gang Zhou, David T Nguyen, Xin Qi, Qing Yang, and Shuangquan Wang. Continuous authentication with touch behavioral biometrics and voice on wearable glasses. *IEEE transactions on human-machine systems*, 47(3):404–416, 2016.

[203] Pramuditha Perera and Vishal M Patel. Face-based multiple user active authentication on mobile devices. *IEEE Transactions on Information Forensics and Security*, 14(5):1240–1250, 2018.

[204] P Jonathon Phillips and Alice J O'toole. Comparison of human and computer performance across face recognition experiments. *Image and Vision Computing*, 32(1):74–85, 2014.

[205] Stephen Porter and Leanne ten Brinke. The truth about lies: What works in detecting high-stakes deception? *Legal and criminological Psychology*, 15(1):57–75, 2010.

[206] Stephen Porter, Leanne Ten Brinke, and Brendan Wallace. Secrets and lies: Involuntary leakage in deceptive facial expressions as a function of emotional intensity. *Journal of Nonverbal Behavior*, 36(1):23–37, 2012.

[207] Kari Pulli, Anatoly Baksheev, Kirill Kornyakov, and Victor Eruhimov. Real-time computer vision with opencv. *Communications of the ACM*, 55(6):61–69, 2012.

[208] Roshan G. Ragel, P. Herath, and Upul Senanayake. Authorship detection of sms messages using unigrams. *2013 IEEE 8th International Conference on Industrial and Information Systems*, pages 387–392, 2013.

[209] Francisco Rangel and Paolo Rosso. Overview of the 7th author profiling task at pan 2019: Bots and gender profiling in twitter. In *Proceedings of the CEUR Workshop, Lugano, Switzerland*, pages 1–36, 2019.

[210] Francisco Rangel, Paolo Rosso, Moshe Koppel, Efstathios Stamatatos, and Giacomo Inches. Overview of the author profiling task at pan 2013. In *CLEF Conference on Multilingual and Multimodal Information Access Evaluation*, pages 352–365. CELCT, 2013.

[211] Kasper Bonne Rasmussen, Marc Roeschlin, Ivan Martinovic, and Gene Tsudik. Authentication using pulse- response biometrics. In *NDSS*, 2014.

[212] Lawrence Ian Reed and Peter DeScioli. The communicative function of sad facial expressions. *Evolutionary Psychology*, 15(1):1474704917700418, 2017.

[213] Bernard Rimé. Interpersonal emotion regulation. *Handbook of emotion regulation*, 1:466–468, 2007.

[214] Bernard Rimé, Pierre Bouchat, Louise Paquot, and Laura Giglio. Intrapersonal, interpersonal, and social outcomes of the social sharing of emotion. *Current opinion in psychology*, 31:127–134, 2020.

[215] Judith L Rochat and D Reiter. Highway traffic noise. *Acoust. Today*, 12(4):38, 2016.

[216] Brendan Rooney, Ciarán Benson, and Eilis Hennessy. The apparent reality of movies and emotional arousal: A study using physiological and self-report measures. *Poetics*, 40(5):405–422, 2012.

[217] Björn Ross, Laura Pilz, Benjamin Cabrera, Florian Brachten, German Neubaum, and Stefan Stieglitz. Are social bots a real threat? an agent-based model of the spiral of silence to analyse the impact of manipulative actors in social networks. *European Journal of Information Systems*, 28(4):394–412, 2019.

[218] Volker Roth, Kai Richter, and Rene Freidinger. A pin-entry method resilient against shoulder surfing. In *Proceedings of the 11th ACM conference on Computer and communications security*, pages 236–245. ACM, 2004.

[219] Y Roth and D Harvey. How twitter is fighting spam and malicious automation, 2018.

[220] J Rottenberg, RD Ray, JJ Gross, JA Coan, and JJB Allen. The handbook of emotion elicitation and assessment. *JJB Allen & JA Coan (Eds.)*, pages 9–28, 2007.

[221] Annie Roy-Charland, Melanie Perron, Olivia Beaudry, and Kaylee Eady. Confusion of fear and surprise: A test of the perceptual-attentional limitation hypothesis with eye movement monitoring. *Cognition and Emotion*, 28(7):1214–1222, 2014.

[222] Hamid Sadeghi, Abolghasem-A Raie, and Mohammad-Reza Mohammadi. Facial expression recognition using geometric normalization and appearance representation. In *2013 8th Iranian Conference On Machine Vision and Image Processing (MVIP)*, pages 159–163. IEEE, 2013.

[223] Pouya Samangouei, Vishal M Patel, and Rama Chellappa. Facial attributes for active authentication on mobile devices. *Image and Vision Computing*, 58:181–192, 2017.

[224] Enver Sangineto, Gloria Zen, Elisa Ricci, and Nicu Sebe. We are not all equal: Personalizing models for facial expression analysis with transductive parameter transfer. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 357–366, 2014.

[225] Allen Sarkisyan, Ryan Debbiny, and Ani Nahapetian. Wristsnoop: Smartphone pins prediction using smartwatch motion sensors. In *2015*

IEEE international workshop on information forensics and security (WIFS), pages 1–6. IEEE, 2015.

[226] Frerk Saxen, Philipp Werner, and Ayoub Al-Hamadi. Real vs. fake emotion challenge: Learning to rank authenticity from facial activity descriptors. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[227] Nolen Scaife, Christian Peeters, and Patrick Traynor. Fear the reaper: Characterization and fast detection of card skimmers. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 1–14, 2018.

[228] Karen L Schmidt, Zara Ambadar, Jeffrey F Cohn, and L Ian Reed. Movement differences between deliberate and spontaneous facial expressions: Zygomaticus major action in smiling. *Journal of nonverbal behavior*, 30(1):37–52, 2006.

[229] Karen L Schmidt, Sharika Bhattacharya, and Rachel Denlinger. Comparison of deliberate and spontaneous facial movement in smiles and eyebrow raises. *Journal of nonverbal behavior*, 33(1):35–45, 2009.

[230] Sean Kelly. Cell Phone Cameras Hidden Inside ATMs Cause Rise In Fraud, 2018.

[231] Nicu Sebe, Michael S Lew, Yafei Sun, Ira Cohen, Theo Gevers, and Thomas S Huang. Authentic facial expression analysis. *Image and Vision Computing*, 25(12):1856–1863, 2007.

[232] RJ Senter and Edgar A Smith. Automated readability index. Technical report, CINCINNATI UNIV OH, 1967.

[233] Chao Shen, Tianwen Yu, Sheng Yuan, Yunpeng Li, and Xiaohong Guan. Performance analysis of motion-sensor behavior for user authentication on smartphones. *Sensors*, 16(3):345, 2016.

[234] Diksha Shukla, Rajesh Kumar, Abdul Serwadda, and Vir V Phoha. Beware, your hands reveal your secrets! In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 904–917, 2014.

[235] Sara Sinclair and Sean W Smith. Preventative directions for insider threat mitigation via access control. In *Insider Attack and Cyber Security*, pages 165–194. Springer, 2008.

[236] Ian Sneddon, Margaret McRorie, Gary McKeown, and Jennifer Hanratty. The belfast induced natural emotion database. *IEEE Transactions on Affective Computing*, 3(1):32–41, 2011.

[237] Kyung-Ah Sohn, Tae-Sun Chung, et al. A graph model based author attribution technique for single-class e-mail classification. In *2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS)*, pages 191–196. IEEE, 2015.

[238] Marina Solnyshkina, Radif Zamaletdinov, Ludmila Gorodetskaya, and Azat Gabitov. Evaluating text complexity and flesch-kincaid grade level. *Journal of Social Studies Education Research*, 8(3):238–248, 2017.

[239] Dawn Xiaodong Song, David Wagner, and Xuqing Tian. Timing analysis of keystrokes and timing attacks on ssh. In *USENIX Security Symposium*, volume 2001, 2001.

[240] Sound and Video Understanding teams pursing Machine Perception research at Google. AudioSet: Traffic noise, roadway noise.

[241] Eugene H Spafford. Observing reusable password choices. 1992.

[242] Jingchao Sun, Xiaocong Jin, Yimin Chen, Jinxue Zhang, Yanchao Zhang, and Rui Zhang. Visible: Video-assisted keystroke inference from tablet backside motion. In *NDSS*, 2016.

[243] Anna Tcherkassof, Damien Dupré, Brigitte Meillon, Nadine Mandran, Michel Dubois, and Jean-Michel Adam. Dynemo: A video database of natural facial expressions of emotions. *The International Journal of Multimedia & Its Applications*, 5(5):61–80, 2013.

[244] TensorFlow. Eager few shot object detection colab. https://tinyurl.com/FineTuningTF, 2020. Accessed: January, 2021.

[245] Chee Meng TEY, Payas GUPTA, and Debin GAO. I can be you: Questioning the use of keystroke dynamics as biometrics.(2013). In *20th NDSS 2013*, pages 1–16, 2013.

[246] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Iccv*, volume 98, page 2, 1998.

[247] United States Attorney's Office, District of Massachussets. Bulgarian National Pleads Guilty to ATM Skimming, 2021.

[248] Onur Varol, Emilio Ferrara, Clayton B Davis, Filippo Menczer, and Alessandro Flammini. Online human-bot interactions: Detection, estimation, and characterization. In *Proceedings of the Eleventh International AAAI Conference on Web and Social Media (ICWSM 2017)*, pages 280–289, 2017.

[249] Igor Vasiljevic, Ayan Chakrabarti, and Gregory Shakhnarovich. Examining the impact of blur on recognition by convolutional networks. *arXiv preprint arXiv:1611.05760*, 2016.

[250] Anne Vermeulen, Heidi Vandebosch, and Wannes Heirman. # smiling,# venting, or both? adolescents' social sharing of emotions on social media. *Computers in Human Behavior*, 84:211–219, 2018.

[251] Aldert Vrij. *Detecting lies and deceit: Pitfalls and opportunities.* John Wiley & Sons, 2008.

[252] Martin Vuagnoux and Sylvain Pasini. Compromising electromagnetic emanations of wired and wireless keyboards. In *USENIX security symposium*, pages 1–16, 2009.

[253] Jun Wan, Sergio Escalera, Gholamreza Anbarjafari, Hugo Jair Escalante, Xavier Baró, Isabelle Guyon, Meysam Madadi, Juri Allik, Jelena Gorbova, Chi Lin, et al. Results and analysis of chalearn lap multi-modal isolated and continuous gesture recognition, and real versus fake expressed emotions challenges. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3189–3197, 2017.

[254] Chen Wang, Xiaonan Guo, Yan Wang, Yingying Chen, and Bo Liu. Friend or foe?: Your wearable devices reveal your personal pin. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, pages 189–200. ACM, 2016.

[255] Ding Wang, Qianchen Gu, Xinyi Huang, and Ping Wang. Understanding human-chosen pins: characteristics, distribution and security. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 372–385. ACM, 2017.

[256] Junjue Wang, Kaichen Zhao, Xinyu Zhang, and Chunyi Peng. Ubiquitous keyboard for small mobile devices: harnessing multipath fading for fine-grained keystroke localization. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, pages 14–27, 2014.

[257] Shangfei Wang, Zhuangqiang Zheng, Shi Yin, Jiajia Yang, and Qiang Ji. A novel dynamic model capturing spatial and temporal patterns for facial expression analysis. *IEEE transactions on pattern analysis and machine intelligence*, 42(9):2082–2095, 2019.

[258] Yichen Wang and Aditya Pal. Detecting emotions in social media: A constrained optimization approach. In *Twenty-fourth international joint conference on artificial intelligence*, 2015.

[259] Sophie F Waterloo, Susanne E Baumgartner, Jochen Peter, and Patti M Valkenburg. Norms of online expressions of emotion: Comparing facebook, twitter, instagram, and whatsapp. *new media & society*, 20(5):1813–1831, 2018.

[260] Thomas Wehrle and Susanne Kaiser. Emotion and facial expression. In *International Workshop on Affective Interactions*, pages 49–63. Springer, 1999.

[261] Wojciech Wodo and Lucjan Hanzlik. Thermal imaging attacks on keypad security systems. In *ICETE*, 2016.

[262] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, 2010.

[263] Chao Yang, Robert Harkreader, and Guofei Gu. Empirical evaluation and new design for fighting evolving twitter spammers. *IEEE Transactions on Information Forensics and Security*, 8(8):1280–1293, 2013.

[264] Ming-Hsuan Yang, David J Kriegman, and Narendra Ahuja. Detecting faces in images: A survey. *IEEE Transactions on pattern analysis and machine intelligence*, 24(1):34–58, 2002.

[265] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5525–5533, 2016.

[266] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Faceness-net: Face detection through deep facial part responses. *IEEE transactions on pattern analysis and machine intelligence*, 40(8):1845–1859, 2017.

[267] Guixin Ye, Zhanyong Tang, Dingyi Fang, Xiaojiang Chen, Kwang In Kim, Ben Taylor, and Zheng Wang. Cracking android pattern lock in five attempts. In *Proceedings of the 2017 Network and Distributed System Security Symposium 2017 (NDSS 17)*. Internet Society, 2017.

[268] Haifeng Yu, Michael Kaminsky, Phillip B Gibbons, and Abraham D Flaxman. Sybilguard: defending against sybil attacks via social networks. *IEEE/ACM Transactions on networking*, 16(3):576–589, 2008.

[269] Umara Zafar, Mubeen Ghafoor, Tehseen Zia, Ghufran Ahmed, Ahsan Latif, Kaleem Razzaq Malik, and Abdullahi Mohamud Sharif. Face recognition with bayesian convolutional networks for robust surveillance systems. *EURASIP Journal on Image and Video Processing*, 2019(1):1–10, 2019.

[270] Stefanos Zafeiriou, Cha Zhang, and Zhengyou Zhang. A survey on face detection in the wild: past, present and future. *Computer Vision and Image Understanding*, 138:1–24, 2015.

[271] Michael Zalewski. Cracking safes with thermal imaging. *ser. http://lcamtuf. coredump. cx/tsafe*, 2005.

[272] Zhihong Zeng, Maja Pantic, Glenn I Roisman, and Thomas S Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE transactions on pattern analysis and machine intelligence*, 31(1):39–58, 2008.

[273] Zheng Zhang, Jeff M Girard, Yue Wu, Xing Zhang, Peng Liu, Umur Ciftci, Shaun Canavan, Michael Reale, Andy Horowitz, Huiyuan Yang, et al. Multimodal spontaneous emotion corpus for human behavior analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3438–3446, 2016.

[274] Wenyi Zhao, Rama Chellappa, P Jonathon Phillips, and Azriel Rosenfeld. Face recognition: A literature survey. *ACM CSUR*, 35(4), 2003.

[275] Rong Zheng, Jiexun Li, Hsinchun Chen, and Zan Huang. A framework for authorship identification of online messages: Writing-style features and classification techniques. *Journal of the American society for information science and technology*, 57(3):378–393, 2006.

[276] Yuqian Zhou, Ding Liu, and Thomas Huang. Survey of face detection on low-quality images. In *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, pages 769–773, 2018.

[277] Tong Zhu, Qiang Ma, Shanfeng Zhang, and Yunhao Liu. Context-free attacks using keyboard acoustic emanations. In *ACM CCS*, pages 453–464, 2014.

[278] Tong Zhu, Qiang Ma, Shanfeng Zhang, and Yunhao Liu. Context-free attacks using keyboard acoustic emanations. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 453–464, 2014.

[279] Yanjia Zhu, Hongxiang Cai, Shuhan Zhang, Chenhao Wang, and Yichao Xiong. Tinaface: Strong but simple baseline for face detection. *arXiv:2011.13183*, 2020.

[280] Li Zhuang, Feng Zhou, and J Doug Tygar. Keyboard acoustic emanations revisited. *ACM Transactions on Information and System Security (TISSEC)*, 13(1):1–26, 2009.

[281] Mircea Zloteanu, Eva G Krumhuber, and Daniel C Richardson. Detecting genuine and deliberate displays of surprise in static and dynamic faces. *Frontiers in Psychology*, 9:1184, 2018.

Appendices

Chapter A

Your PIN Sounds Good

A.1 Additional Details on Data Collection

We recorded subjects entering 4-digit PINs on a simulated ATM, shown in Figure A.1. Our dataset was based on experiments with 22 participants; 19 subjects completed three data collection sessions, while 4 subjects completed only one session, resulting in a total of 61 sessions. At the beginning of each session, the subject was given 45 seconds to get accustomed with the keypad of the ATM simulator. During this time, they were free to type as they pleased. Next, a subject was shown a PIN on the screen for ten seconds (Figure A.2a), and, once it disappeared from the screen, asked to enter it four times (Figure A.2b). Subjects were advised not to read the PINs out loud. This process was repeated for 15 consecutive PINs. During each session, subjects were presented with the same 15-PIN sequence 3 times. Subjects were given a 30-second break at the end of each sequence.

Specific 4-digit PINs were selected to test whether: inter-keypress time is proportional to Euclidean Distance between keys on the keypad; and the *direction of movement* (up, down, left, or right) between consecutive keys in a keypair impacts the corresponding inter-key time. We show an example of these two situations on the ATM keypad in Figure A.3. We chose a set of PINs that allowed collection of a significant number of key combinations appropriate for testing both hypotheses. For instance, PIN `3179` tested horizontal and vertical distance two, while `1112` tested distance 0 and horizontal distance 1.

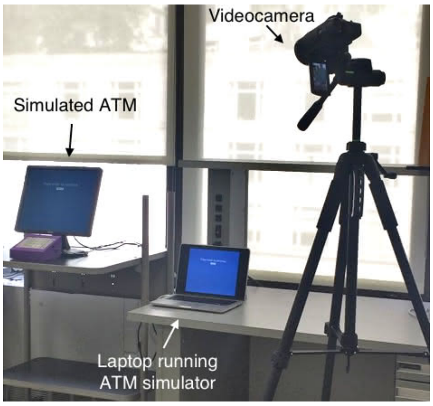
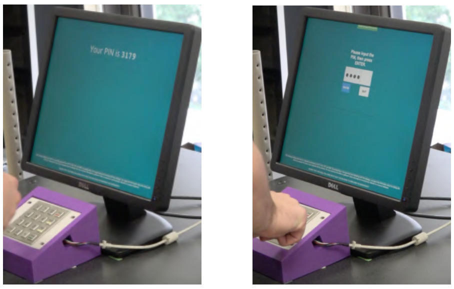


Figure A.1: Setup used in PIN inference experiments.



(a) (b)

Figure A.2: ATM Simulator during a data collection session. (a) The simulator displays the next PIN. (b) A subject types the PIN from memory.

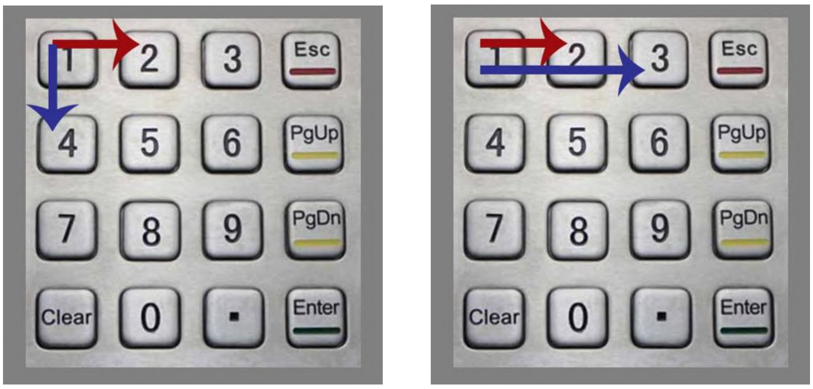
Sessions were recorded using a Sony FDR-AX53 camera, at a pixel resolution of 1,920×1,080 pixels, and 120 frames per second. At the same time, ATM simulation software collected millisecond-accurate inter-key distance ground truth by logging each keypress. PIN feedback was shown on a DELL 17" LCD screen with a refresh rate of 60 Hz, which resulted in each frame being shown for 16.7 ms.

A.1.1 Timing Extraction from Video

We developed software that analyzes video recordings to automatically detect the appearance of masking symbols and log corresponding timestamps. This software uses OpenCV [207] to infer the number of symbols present in each image. All frames are first converted to grayscale, and then processed through a bilateral filter [246] to reduce noise arising from the camera's sensor. The resulting images are analyzed using Canny Edge detection [77] to capture the edges of the masking symbol. External contours are compared with the expected shape of the masking symbol. When a masking symbol is detected, software logs the corresponding frame number.

A.2 Inter-keystroke Timings and Key Distance

We analyzed the relationship between inter-keystroke timings and Euclidean Distance between consecutive keys, and between inter-keystroke timings and direction of movement on the keypad.



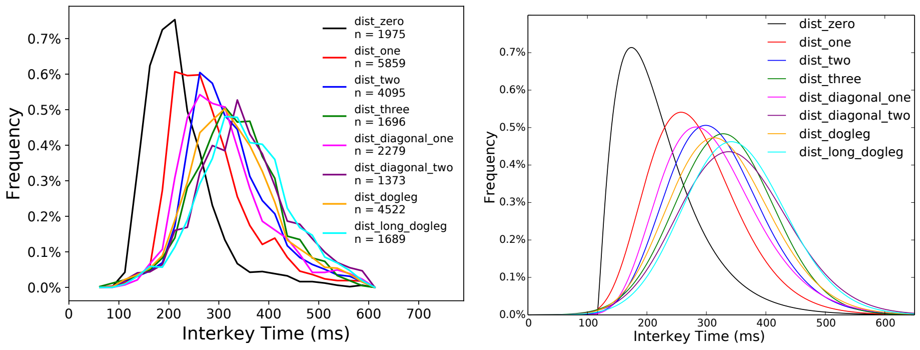
(a)

(b)

Figure A.3: ATM keypad in our experiments. (a) To type keypairs 1-2 and 1-4, the typing finger travels the same distance in different directions. (b) Keypairs 1-2 and 1-3 require the typing finger to travel different distances in the same direction.

Distance. Across all subjects, we observed that distributions of inter-keystroke latencies were distinct in all cases (for p-value $< 5 \cdot 10^{-6}$), with the following exceptions: (1) latencies for distance 2 (e.g., keypair 1-3) were close to latencies for distance 3 (keypair 2-0); (2) latencies for distance 2 were close to latencies for diagonal 1×1 (e.g., keypair 4-8); latencies for distance 3 were close to latencies for 2×1 diagonal (i.e. "2" to "9", "1" to "6", etc.), and diagonal 2×2 (e.g., keypair 7-3), and diagonal 3×2 (e.g., keypair 3-0). Figure A.4a shows the various probability distributions, while Figure A.4b models these different probability distribution functions as gamma distributions. In Figure A.4a, dist_zero indicate keypairs composed of the same two digits. dist_one, dist_two, and dist_three shows timings distributions for keypairs with horizontal or vertical distance one (e.g., keypair 2-5), two (e.g., 2-8), and three (2-0), respectively. dist_diagonal_one and dist_diagonal_two indicates keypairs with diagonal distance one (e.g., 2-4) and distance two (e.g., 1-9), respectively. dist_dogleg and dist_long_dogleg show timing distributions of keypairs such as 1-8 and 0-3. In Figure A.4b, dist_one_horizontal and dist_one_vertical indicate Euclidean Distance right in the left/right directions, and up/down directions, respectively, while dist_one_up, dist_one_down, dist_one_left, and dist_one_right indicate distances one in the up, down, left, and right directions.

Direction. The relative orientation of keypairs characterized by the same Euclidean distance (e.g., 2-3 vs. 2-5) has a negligible impact on the



(a) From raw data. (b) Modeled as gamma distributions.

Figure A.4: Inter-keystroke timings of all possible distances for ATM keypad typing.

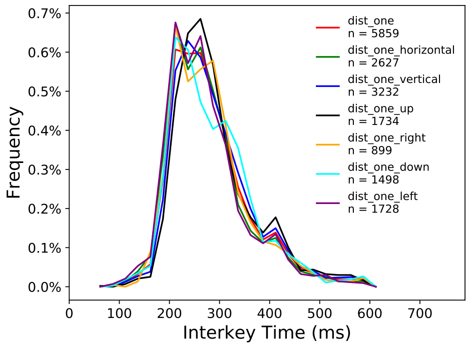


Figure A.5: Frequency of inter-keystroke timings for Euclidean Distance of one. dist_one indicates latency distribution for distance one in any direction.

corresponding inter-key latency. We observed that the distributions of keypress latencies observed from each possible direction between keys were not significantly different (for p-value $< 10^{-4}$). Figure A.5 shows different probability distributions relative to various directions for Euclidean distance 1.

A.3 PIN Guessing Algorithm

We are not aware of any publicly-available PIN timing datasets that can be used to train our algorithm (PILOT). To compute the attack baseline, we considered all PINs to be equally likely, i.e., we are modeling PINs as random four-digit strings. This is consistent with how many European banks assign PINs to bank cards [97], and with the work of Bonneau et al. [33], which showed that users are reluctant to change the random PIN provided by their bank.

Using the data we collected, we mapped the distribution of inter-keypress latencies, and used the resulting probabilities to test the effectiveness of PINs prediction from inter-key latencies. Our PIN guessing algorithm is composed of two parts: (1) an algorithm that estimates distances from keystroke timings; and (2) an algorithm that ranks PINs based on the estimated distances. The source code of the algorithm can be found at https://spritz.math.unipd.it/datasets/PILOT/ The core idea is to consider the PIN pad as a weighed multigraph. The graph nodes represent the keys, and are labeled 0-9. Keys are connected by weighted edges. The weight of an edge corresponds to the Euclidean distance between the corresponding keys, using the distance between two adjacent keys (e.g., 1 and 2) as unit. We identified 8 possible distances: zero distance (e.g., key 3 followed by key 3, $weight = 0$); horizontal or vertical distance one (e.g. keys 1-2, $weight = 1$); horizontal or vertical distance two (e.g., keys 1-3, $weight = 2$); vertical distance three (e.g., keys 2-0, $weight = 3$); diagonal distance one (e.g., keys 1-5, $weight = \sqrt{1^2 + 1^2}$); diagonal distance two (e.g., keys 1-9, $weight = \sqrt{2^2 + 2^2}$); short diagonal distance (e.g., keys 1-8, $weight = \sqrt{1^2 + 2^2}$) and long diagonal distance (e.g. keys 1-0, $weight = \sqrt{1^2 + 3^2}$).

For each PIN, we created three sets a subgraphs, indicated as S_1, S_2, and S_3, composed only of the nodes connected by edges with the same weight as the estimated distance. Specifically, S_i contains all the two-nodes subgraphs such that their edges have weight equal to the estimated distance between the keys in the i-th PIN digraph.

We combined the subgraphs in these sets by ensuring that, for $i = 1$ and $i = 2$, the second node of a graph from S_i is the same as the first node of a graph from S_{i+1}. For instance, given estimated distances 3, 0, and $\sqrt{2}$, our algorithm extracts the subgraphs shown in Figure A.6. It then refines these choices by removing all subgraphs from S_2 which do not have nodes 2 and 0 as their first node. The same rule is applied to S_3. The two resulting graphs are shown in Figure A.7.

Not all estimated distances correspond to possible PINs. For instance, estimated distances 3, 0, and $\sqrt{8}$ do not match any PIN that can be typed on the pad used for the experiments: distance 3 indicates that the second PIN digit must be either 0 or 2; as a consequence, distance 0 associated to the second PIN digraph restricts the third PIN digit to 0 or 2; however, the set of keypairs with a relative distance of $\sqrt{8}$ (i.e., $\{(1,9),(7,3),(9,1),(3,7)\}$) does not include keys 0 or 2. Therefore, estimated distances 3, 0, and $\sqrt{8}$ do not lead to any valid PIN. Figure A.8 shows a visual representation of this example.

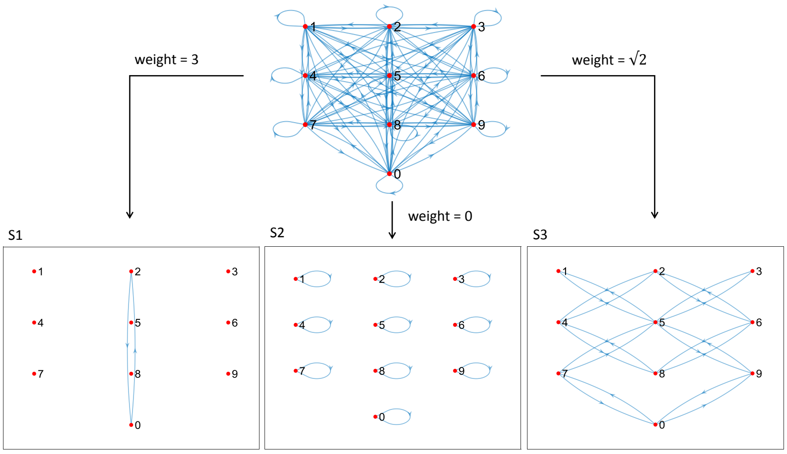
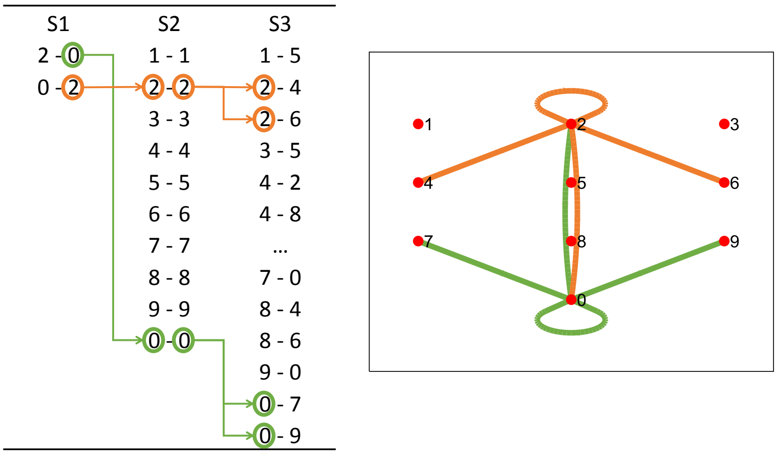


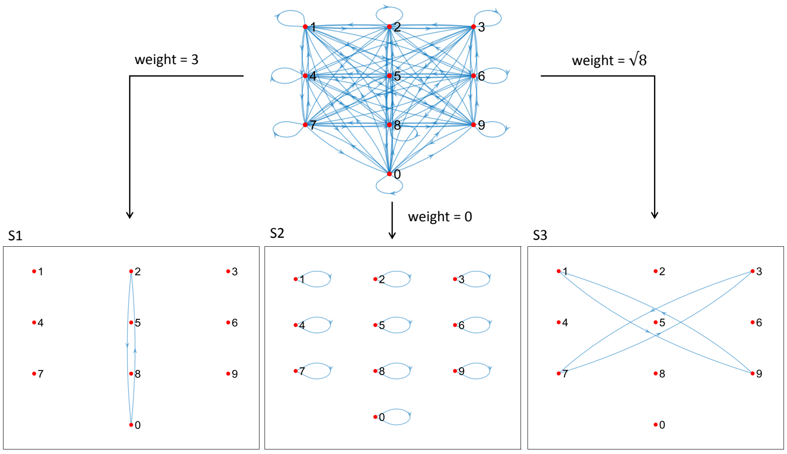
Figure A.6: Full graph, and subgraphs $h_1 \in S_1$ ($weight = 3$), $h_2 \in S_2$ ($weight = 0$), and $h_3 \in S_3$ ($weight = \sqrt{2}$).



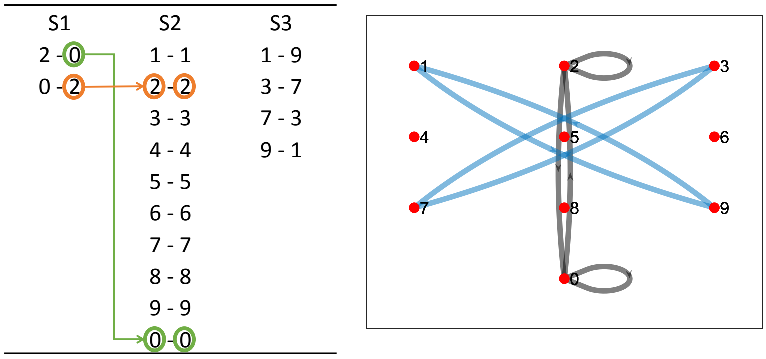
S1	S2	S3
2 - 0	1 - 1	1 - 5
0 - 2	2 - 2	2 - 4
	3 - 3	2 - 6
	4 - 4	3 - 5
	5 - 5	4 - 2
	6 - 6	4 - 8
	7 - 7	…
	8 - 8	7 - 0
	9 - 9	8 - 4
	0 - 0	8 - 6
		9 - 0
		0 - 7
		0 - 9

Figure A.7: Nodes sequences for input distances 3, 0, $\sqrt{2}$.

For each triplet of estimated distances, the number of associated PINs may differ. For example, 58 triplets have no associated PIN (e.g., distances 3, 0, and $\sqrt{8}$). The remaining 454 combinations vary from a minimum of 2 associated PINs (57 combinations; e.g., distances 3, 3, and 3 correspond only to PINs 2020 and 0202) to a maximum of 216 PINs (distances 1, 1, and 1). If the adversary is able to reconstruct the distances between digraphs without errors, this process drastically reduces the number of attempts needed to guess the PIN compared to a random guessing. Figure A.9 shows the benefit of this approach in terms of percentage of PINs guessed within a fixed number of attempts. However, due to the overlapping between timing distributions shown in figures A.4 and A.5, the adversary cannot always estimate distances correctly.



(a)



S1	S2	S3
2 - 0	1 - 1	1 - 9
0 - 2	2 - 2	3 - 7
	3 - 3	7 - 3
	4 - 4	9 - 1
	5 - 5	
	6 - 6	
	7 - 7	
	8 - 8	
	9 - 9	
	0 - 0	

(b)

Figure A.8: (a) Subgraphs corresponding to distances 3, 0, and $\sqrt{2^2 + 2^2}$. (b) Nodes connected by vertices weights 3, 0, and $\sqrt{2^2 + 2^2}$. No common nodes are in S_2 and S_3, and therefore this combination of distances does not correspond to any PIN.

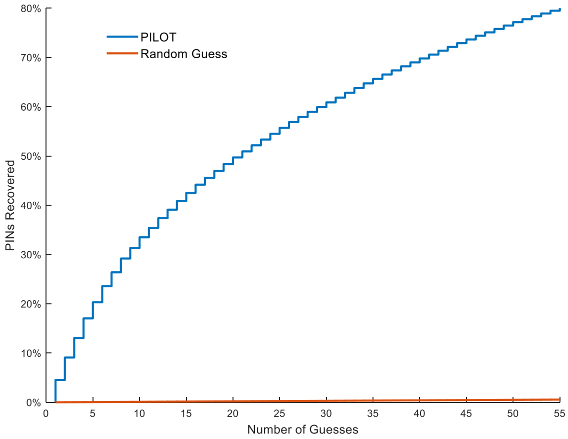


Figure A.9: CDF showing the number of PINs recovered under the assumption that distances are recovered without error.

Chapter B

Hand me Your PIN

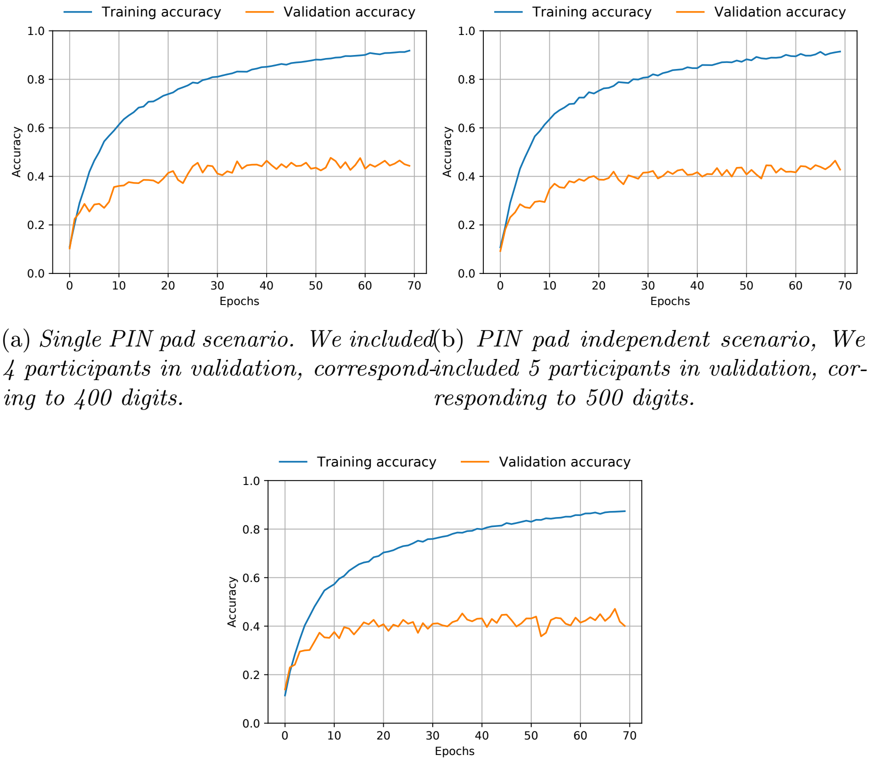
B.1 Neural Networks Additional Info

In Figure B.1, we show the training and validation accuracy for the three models selected after the random grid search. In the *Mixed* scenario, the validation accuracy grows faster than in *PIN pad independent* scenario and *Single PIN pad* scenario, reaching faster the plateau. Indeed, in the *Mixed* scenario, the validation accuracy stabilizes after 20 epochs, while we require more than 35 epochs for the other scenarios. This difference can be linked to a larger training size and a higher variance in the samples since the *Mixed* scenario is the only one to include videos from both PIN pads in the training phase.

Next, we report some statistics about the training execution times for the three scenarios we consider.

- **Single PIN pad** scenario: the training set is composed of 32 participants, corresponding to 16 000 samples of 11 frames each. Our model takes 1 577 seconds to complete an epoch (i.e., approximately 34 hours to complete the entire training phase).
- **PIN pad independent** scenario: the training set is composed of 35 participants, corresponding to 17 500 samples of 11 frames each. Our model takes 1 598 seconds to complete an epoch (i.e., approximately 34 hours to complete the entire training phase).
- **Mixed** scenario: the training set is composed of 46 participants, corresponding to 23 000 samples of 11 frames each. Our model takes

2 240 seconds to complete an epoch (i.e., approximately 46 hours to complete the entire training phase).



(a) *Single PIN pad scenario. We included 4 participants in validation, corresponding to 400 digits.*

(b) *PIN pad independent scenario, We included 5 participants in validation, corresponding to 500 digits.*

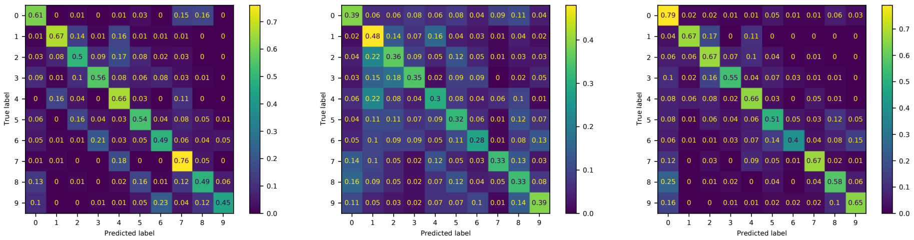
(c) *Mixed scenario. We included 6 participants in validation, corresponding to 600 digits.*

Figure B.1: Training and validation accuracy for our three scenarios.

B.2 Key Accuracy Analysis

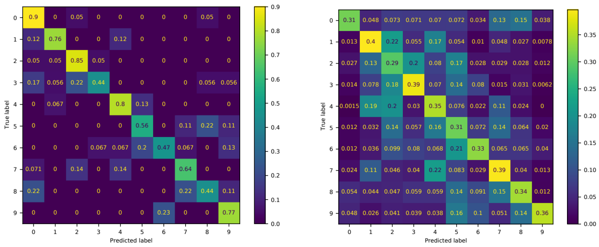
In this section, we provide further analysis on the key accuracy for our attack. Figure B.2 highlights that the accuracy on a single key is worse in the *PIN pad independent* scenario. Although the performance is considerably lower than the other two scenarios in Top-1 accuracy, it is interesting that the error dispersion affects the keys topologically close to the target one.

In Figure B.3, we compare our model and human performance on the key classification task. The misclassification error and the dispersion result are significantly lower for our algorithm. Moreover, it can be noticed how the four keys on which humans perform the best match those in the corners of our keypad (i.e., 1, 3, 7, and 9).



(a) *Single PIN pad* scenario. (b) *PIN pad independent* scenario. (c) *Mixed* scenario.

Figure B.2: Confusion matrices of key predictions (predicted labels) vs. true values (true labels) for our three scenarios.



(a) *Recalculated confusion matrix for our algorithm (Mixed scenario).* (b) *Confusion matrix for humans.*

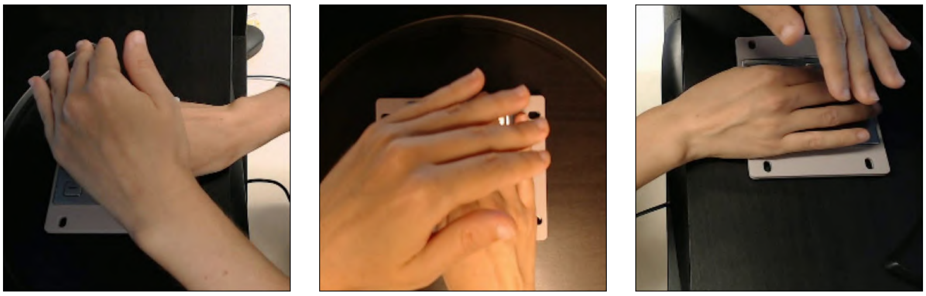
Figure B.3: Confusion matrix comparison between our algorithm and humans.

B.3 Additional Experiments

To gain further insight into how coverage can affect the attack performance, we grouped the tested users by the coverage strategy:

- `Side`: The non-typing hand rests on the side of the palm and is angled to cover the keys of the PIN pad (40% of users applied this covering strategy).
- `Over`: The non-typing hand is raised completely off the surface, covering the PIN pad both with the entire back of the hand and the fingers (43% of users applied this covering strategy).

- `Top`: The fingers of the non-typing hand rest on the top of the PIN pad, and the back of the hand is used for the coverage (17% of users applied this covering strategy).



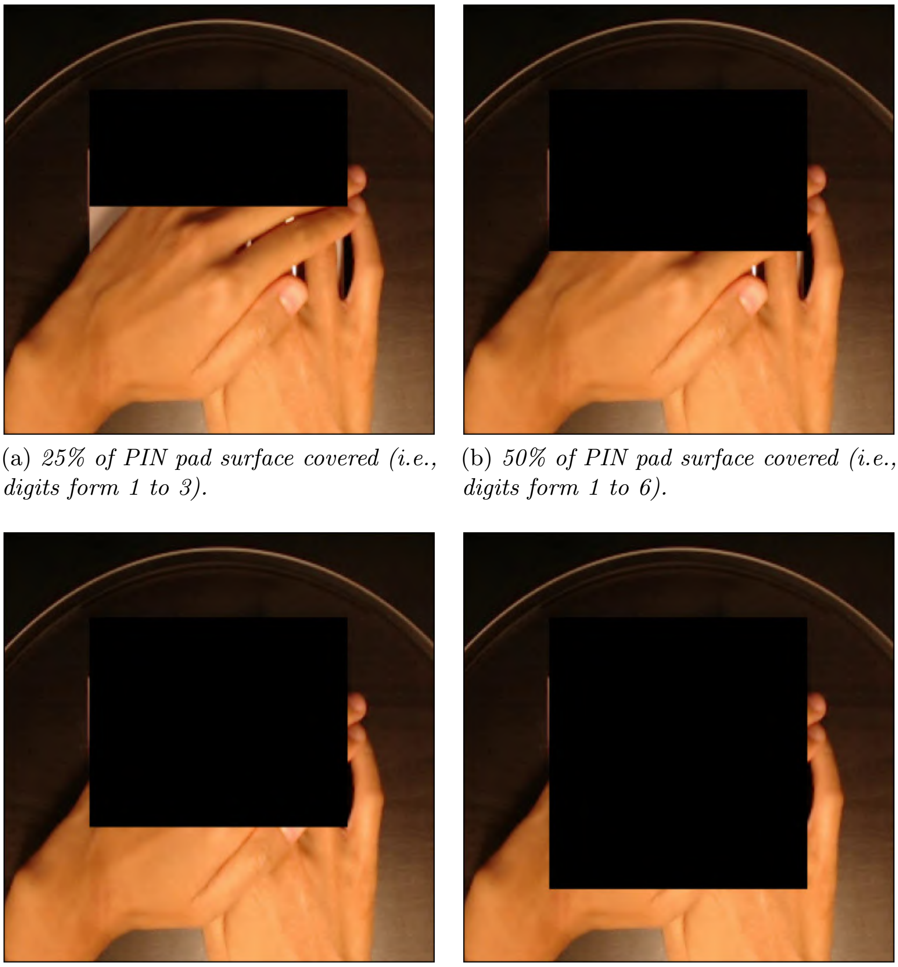
(a) *Left-corner camera.* (b) *Center camera.* (c) *Right-corner camera.*

Figure B.4: Same video frame recorded by three cameras.

In Table B.1, we report key and PIN TOP-3 accuracies for our approach. Clearly, `Side` covering strategy provides the least protection and should be avoided. At the same time, the `Over` and `Top` covering strategies provide much better protection. Interestingly, we see that with the `Over` covering strategy, the *Mixed* scenario reaches lower accuracy than the *Single PIN pad* scenario. We postulate this happens as this covering strategy makes it less "natural" for the user to type, deceiving the deep learning algorithm. Further attack improvements could be made with datasets having examples of one covering strategy only. For the `Top` covering strategy, there were no data for two out of three scenarios (denoted NA in Table B.1).

For the PIN shield countermeasure, we depict various levels of hiding in Figure B.5. There, 25% denotes that the first row of the PIN pad is covered (simulated with a black patch), 50% first two rows, 75% first three rows, and finally, 100% all four rows of the PIN pad are covered. Note that we do not include the covering with the other hand into these percentages.

Table B.2 provides results for several additional attack configurations. First, we performed two experiments simulating a lower camera quality or a larger camera distance from the PIN pad. For this purpose, we reduced the model input resolution from 250 x 250 to 125 x 125 and to 64 x 64. Results show that our model maintains an accuracy higher than 20%, even when halving the input resolution (i.e., doubling the camera distance). However, this is not to be considered as a physical limitation for our attack since if the attacker places a camera outside the ATM chassis, it is possible to use an optical zoom. Further, many pinhole cameras can record with a resolution



(a) *25% of PIN pad surface covered (i.e., digits form 1 to 3).*

(b) *50% of PIN pad surface covered (i.e., digits form 1 to 6).*

(c) *75% of PIN pad surface covered (i.e., digits form 1 to 9).*

(d) *100% of PIN pad surface covered (i.e., no digit is visible).*

Figure B.5: PIN pad shield configurations.

Covering strategy	**Scenario**	**Key accuracy**	**PIN TOP-3 accuracy**
`Side`	Single	0.64	0.30
	Independent	0.42	0.12
	Mixed	0.77	0.53
`Over`	Single	0.52	0.12
	Independent	0.31	0.10
	Mixed	0.46	0.07
`Top`	Single	NA	NA
	Independent	0.41	0.13
	Mixed	NA	NA

Table B.1: Performance of our attack for different covering strategies in *Single PIN pad*, *PIN pad independent*, and *Mixed* scenarios. `Top` covering participants were present in the *PIN pad independent* scenario only, as for the others, no data were available (NA).

up to 1 080p [1], which is higher than the resolution we used to collect our dataset (720p).

Next, we investigated the accuracy of our attack leveraging different camera positions. In particular, we performed two experiments training and testing our model with the left-corner and the right-corner cameras, respectively. Figure B.4 shows the camera views used in our experiments. The results give a significant difference in performance if the camera is on the right or the left. This is because the participants in our experiment were right-handed, and therefore filming from the right had worse coverage of the PIN pad and typing hand. In contrast, the typing hand and the PIN pad were almost completely covered using shots from the left, significantly reducing the model's performance. We also evaluated whether using video from all three cameras in training (the experiment "multi-camera training" in Table B.2) could improve the accuracy of our model when compared with videos recorded from the center camera only. The results show a drop in performance, which we attribute to the higher variance in the data provided as input to the model.

Finally, we report the results of our model without data augmentation and without including the blacklisted users in the training set. In both configurations, the performance of our model drops, showing that reducing the training size is penalized heavily. Note that even in the worst case of a camera placed on the left corner (i.e., the one with less visibility), *our model still performs better than an average human.*

[1] https://www.dsecctv.com/Prod_telecamere_spioncino_porta_AHD.htm

Experiment	**Key accuracy**	**PIN TOP-3 accuracy**
Input resolution 125 x 125	0.55	0.23
Input resolution 64 x 64	0.47	0.15
Left-corner camera	0.46	0.10
Right-corner camera	0.62	0.31
Multi-camera training	0.53	0.22
No data augmentation	0.44	0.11
Blacklisted excluded in training	0.54	0.18

Table B.2: Additional attack configurations and results in the *Mixed* scenario.

In this paper, we used the feedback sound emitted by the PIN pad as a detection system for the frames containing a keystroke. To evaluate the impact of other frame detection systems, we conducted an experiment varying the frame extraction precision. We simulated the detection error by adding Gaussian noise with mean zero to the ground truth (i.e., the frame position in the video). In Table B.3, we report the single key and the PIN TOP-3 accuracies for the *Mixed* scenario, simulating five levels of the frame detection error. Compared to the results obtained using the audio feedback (key accuracy 0.61, 5-digits PIN Top-3 accuracy 0.30), we see that our model works well even with small/medium levels of frame detection error (i.e., less than five frames). In particular, for a frame error confidence of three (i.e., when the frames are detected through the appearance on the screen of the masked symbols [42]), the performance drops only 1% both for key and TOP-3 PIN accuracies. Contrarily, when the detection error becomes high (i.e., more than 15 frames), the performance of our model decreases significantly. This happens since the frames considered by the model do not contain information related to the target key, as they are too temporally shifted. Naturally, if the attacker recognizes a situation like this, it would be possible to mitigate the effect of detection error by not using the feedback sound but observing the appearance of "*" symbols on the screen.

Frame error confidence ($p < 0.01$)	**Key accuracy**	**PIN TOP-3 accuracy**
3	0.60	0.29
5	0.59	0.26
10	0.54	0.16
15	0.49	0.12
20	0.12	0.06

Table B.3: Performance of our attack in the *Mixed* scenario assuming different levels of frame detection error.