PROTECTING SENSITIVE INFORMATION AND ASSETS

SYLLABUS

Authentication Abuse prevention-Account reputation scoring. User Authentication with Keystroke Recognition- Keystroke Dynamics, Anomaly Detection with keystroke Dynamics. Biometric Authentication with Facial Recognition-Pros and Cons, Eigen faces facial recognition, Dimensionality Reduction with PCA (Principal Component Analysis).

AUTHENTICATION ABUSE PREVENTION

- Authentication abuse prevention is considered a vital aspect of cybersecurity, particularly in the current environment characterised by increasing disintermediation of traditional services into digital forms, such as e-commerce and home banking.
- It focuses on identifying and preventing potential threats that target a user's digital identity, primarily to prevent identity theft.
- The rapid spread of the Internet of Things (IoT) has increased the possibility of unauthorised access through counterfeit or stolen credentials.
- The attack surface of cyberspace is growing exponentially due to the vast number of connections between humans and machines, and between machines themselves, making information leakage more likely.
- Protecting user accounts is crucial not only for data integrity but also due to the reputational risk for organisations and potential legal responsibilities, especially with the rise of automated services and regulations like General Data Privacy Regulation (GDPR).
- Preventing abuse involves monitoring suspicious activities, including attempts to compromise passwords.

Are Passwords Obsolete?

- While passwords have historically been the primary tool for protecting user accounts, they have "shown their limits".
- This is largely due to the increasing number of online services and platforms, which requires users to memorise a growing number of passwords.
- The difficulty in managing many unique, strong passwords leads users to often reuse the same password across multiple accounts and services.
- This practice significantly increases the attack surface and the risk of compromise.

- If an attacker compromises one account, such as a personal email, it becomes very likely they can violate other accounts and steal the victim's digital identity.
- Identity theft is highlighted as a major threat, enabling attackers to conduct illegal activities like money laundering through the creation of fake bank accounts using the victim's identity, often without their knowledge.
- Therefore, while passwords are still in use, their inherent weaknesses and common user habits mean they are not sufficient as a standalone security measure in today's threat landscape.
- Security measures that integrate password authentication with monitoring and authorisation procedures have been adopted over time.
- There is a move towards supplementing or potentially replacing passwords with other methods like biometric credentials.
- The protection of a user's account is not limited to the simple verification of the correctness of the entered password and the correspondence of the password to the user account, but also the various account activities that are recorded, such as simultaneous access from IP addresses belonging to different geographical areas, or the use of different devices, such as PCs, and smartphones, browsers, and operating systems that are unusual or have never been used before.
- The objective of this monitoring is obviously to detect possible credential theft by the attackers, who attempt to access user accounts by exploiting previously violated passwords.
- To achieve this level of contextual security awareness, it is necessary to integrate monitoring activities that occur on user accounts with anomaly detection procedures that make use of automated learning algorithms, thus learning to discriminate between different suspicious activities based on the habits and behavior of the users themselves.
- It is also possible to replace the same passwords with authentication procedures that make use of a user's biometric credentials, such as their iris, voice, fingerprints, or face.

Common Authentication Practices

Traditional authentication primarily relies on **password** verification. Over time, various verification forms have been introduced to ensure credentials belong to the legitimate owner.

Some of these are based on adopting **second authentication factors**. Examples include temporary passwords transmitted as **OTP (One-Time Password) codes** via SMS to a user's phone number or via email.

Two-factor authentication has also been introduced by financial institutions to prevent abuse of payment instruments.

However, the reliability of these second authentication factors depends on the integrity of the support and channels used for transmission and management. If a user's email account is hacked or malware intercepts SMS messages on a smartphone, these factors become ineffective. Their effectiveness assumes diversification of the supports used, following a risk management best practice of not keeping all sensitive information on a single support. If this diversification assumption is not met, the reliability of these procedures diminishes.

More recently, there has been increasing reliance on new forms of authentication and detection of suspicious user accounts using **biometric recognition**. This is benefiting from the diffusion of neural networks and embedded hardware like cameras. Biometric recognition can supplement or replace traditional password-based authentication. Distinctive physical elements reliably traceable to a user include the **iris, face, fingerprints, and voice**.

Biometric behaviours and habits can also form patterns associated with an individual user; **keystroke typing** (keystroke dynamics) is given as an example, which, like freehand writing, can reliably identify different subjects. Integrating traditional methods with monitoring tasks and anomaly detection procedures using automated learning algorithms is also presented as a key practice.

How to Spot Fake Logins

Identifying potential credential theft by attackers who exploit previously violated passwords is a key objective of monitoring user account activities. The use of security tokens like passwords, SMS, and OTPs should be integrated with **automated anomaly detection procedures**.

Anomalies related to user account management that should be monitored include,:

- **Brute force access attempts** aimed at identifying a user's password by entering different passwords within a limited period.
- **Simultaneous access from IP addresses belonging to different geographical areas**
- The **use of devices, software, and operating systems that are uncommon for the user**
- **Frequency and typing speed that is incompatible with human operators**. This relates to keystroke dynamics, where different typing patterns can identify impostors using stolen passwords.

The list of events to monitor can vary depending on the analysis context. Automated detection of anomalies is important once a historical basis of representative events has been fed into the system.

Fake Login Management

There are different strategies for managing detected suspicious access.

- **Reactive management** involves configuring alarm systems that trigger an event, such as automatically suspending or blocking a user account, once possible unauthorised access is identified. While simple to implement, this strategy has significant drawbacks. It can lead to **Denial of Service (DoS) attacks** targeting legitimate users, where an attacker simulates unauthorised access attempts to trigger the automatic blocking of accounts, damaging the organisation's reputation and causing disruption. Reactive systems often use default, global triggers that do not learn to recognise individual user behaviours,. This approach reads reality through the "rear-view mirror," assuming the future is the same as the past and failing to automatically adapt to rapid changes in context. It is also ineffective against **stealth-mode attacks** that do not cause anomalous peaks exceeding the alarm threshold, allowing attackers to remain hidden and perform information gathering or abusive operations undetected. A large-scale user account violation at Yahoo! was conducted in stealth mode and took years to discover.

- **Predictive management**, on the other hand, requires adopting a predictive approach to anomaly detection. This strategy starts by analysing past data to reveal latent patterns and extrapolate future user behaviours, identifying potential attempts at compromise or fraud in a timely manner. It must consider changes in the context and scenario affecting both user and attacker behaviour. Automated learning algorithms and anomaly detection procedures are necessary to discriminate between suspicious activities based on individual user habits and behaviour. **Unsupervised learning and clustering algorithms** are particularly indicated for exploring latent and unknown patterns in suspicious user behaviour.

Predicting the Unpredictable

- The task of predictive analytics is to reveal hidden patterns, identifying latent trends within the data.
- To this end, it is necessary to combine various data mining and machine learning (ML) methodologies in order to exploit sets of structured and unstructured data from the various heterogeneous information sources available to the organization.
- This way, it is possible to translate the raw data into actionable predictive responses, applying different automated learning algorithms to the data.
- Different algorithms will obviously provide different results in terms of predictive accuracy.
- Classification algorithms are particularly suitable when we have to manage discrete answers (spam or ham), while if we need continuous outputs (that is,

output values characterized by greater granularity), the use of regression algorithms is our preferred choice.
- Similarly, to manage large-scale classification tasks, we can consider the use of linear support vector machines (SVMs) and the use of decision trees and random forests, which usually provide the best results when we need to categorize data.
- A special mention must then be given to unsupervised learning and clustering algorithms, which are particularly indicated in the exploration of latent and unknown patterns within the data, to carry out tasks such as the anomaly detection of suspicious user behavior.

Choosing the Right Features

For a predictive approach to detecting potential user account violations, selecting the correct features to monitor is crucial. The relevant features vary based on the threats considered most likely.

Examples of features mentioned include:

- For brute forcing attempts on user credentials (user ID and password): monitoring the **number of failed access attempts (logins)**, their growth rate, and variations over time,.
- Other relevant indicators can include the **frequency of password changes, failed logins, and password recovery attempts**.
- For detecting stealth-mode attacks by attackers with stolen passwords or session hijacking: monitoring the **IP addresses associated with user logins** to check for simultaneous access from geographically distant areas or overly frequent access using devices and software uncommon for that specific user.
- For estimating **account reputation scoring**: features like the **number and frequency of user posts**, access via **proxy, VPN, or other IP anonymisation systems**, the use of **uncommon user agents** (such as scripts) to log in, and the user's **speed in typing text on the keyboard** can be considered.
- For **keystroke dynamics**: timing features such as the time between consecutive key presses (**keydown-keydown**), the time between releasing one key and pressing the next (**keyup-keydown**), and the time between pressing and releasing a key (**hold**) are used.

Choosing features that lack a high correlation with potential anomalous behaviours leads to high error rates (false positives), invalidating their usefulness. **Feature engineering** practices are important for transforming raw data into suitable features for models. Reducing the dimensionality of datasets (the number and type of features) can also significantly improve algorithm performance, particularly for heavy datasets like images.

Preventing Fake Account Creation

Preventing the spread of fake profiles within platforms is an activity that requires monitoring. **Monitoring** should encompass both the **request to activate new accounts** and the **identification of fake profiles among already existing accounts** that need blocking or cancellation due to misconduct.

- A possible indicator for the anomalous creation of new accounts (which are likely fake profiles) is the **activation of numerous new accounts from the same IP address within a short period of time** (e.g., less than an hour).
- For existing accounts, an anomaly indicator that can reliably suggest a fake profile is a **large number of user posts delivered within a short period**, which might indicate the presence of a bot spreading spam.

Monitoring user account activity for both new and existing accounts is crucial for preventing malicious activities. Associating a measure of **account reputation scoring** based on user behaviour can help identify attacks conducted in stealth mode, avoiding reliance solely on detecting anomalous and noisy peaks of activity,,. Various features, as mentioned previously, can be used to dynamically estimate this reputation score.

ACCOUNT REPUTATION SCORING

Authentication abuse prevention is considered vital in the current digital landscape, which is marked by the increasing move of traditional services to digital forms like e-commerce and home banking. A primary focus is to identify and prevent threats targeting a user's digital identity, particularly to prevent identity theft. The expanding attack surface of cyberspace due to numerous connections increases the likelihood of information leakage. Protecting user accounts is crucial not only for data integrity but also due to reputational risk and potential legal liabilities, especially with the rise of automated services and regulations like GDPR. Preventing abuse involves monitoring suspicious activities, including attempts to compromise passwords.

Account Reputation Scoring

Protecting user accounts encompasses monitoring activities on both newly created and existing accounts to prevent malicious actions. Identity theft is a significant threat, enabling attackers to engage in illegal activities such as money laundering through creating fake bank accounts using the victim's identity. Security measures have evolved to integrate password authentication with monitoring and authorisation procedures, aiming to enhance contextual awareness of user activities.

It is advisable to associate a measure of **reputation (reputation scoring)** that is estimated based on the behaviour of the user associated with the account. This reputation scoring serves a crucial purpose in identifying attacks conducted in **stealth mode**. By leveraging reputation scoring, organisations can avoid relying solely on detecting anomalous and noisy peaks of activity to spot threats.

Various features can be considered when estimating the reputation score for each user account. These include:

- The number and frequency of user posts published over a period of time.
- Access to the user account via proxy, VPN, or other IP anonymisation systems.
- The use of uncommon user agents, such as scripts, for logging in.
- The user's speed in typing text on the keyboard (keystroke typing).

These and other features can be effectively taken into consideration to train algorithms for the dynamic estimation of an individual user's reputation score.

Classifying Suspicious User Activity

Identifying attempts at fraud or the compromising of applications by malicious users as they happen is highlighted as an emerging area of application for Deep Learning. To achieve contextual security awareness, monitoring activities on user accounts need to be integrated with automated anomaly detection procedures that utilise automated learning algorithms. This allows systems to learn to discriminate between different suspicious activities based on the specific habits and behaviour of individual users.

The sources suggest that the use of authentication procedures based on security tokens, such as passwords, SMS, and OTPs, should at least be integrated with automated anomaly detection procedures. Anomalies related to the management of user accounts that can be monitored include:

- Brute force access attempts, which aim to identify a user's password by entering different passwords within a limited timeframe.
- Simultaneous access from IP addresses belonging to different geographical areas.
- The use of devices, software, and operating systems that are uncommon for the user.
- Frequency and typing speed that is incompatible with human operators.

The list of events to monitor can be expanded and adjusted depending on the specific analysis context. However, automated detection of anomalies is crucial once a historical basis of representative events has been fed into the system.

Managing detected suspicious access can follow different strategies, including reactive and predictive approaches. A predictive approach to anomaly detection is recommended, which analyses past data to reveal latent patterns and extrapolate future user behaviours, allowing for the timely identification of potential compromise or fraud attempts. This approach must account for changes in the context and scenario affecting both user and attacker behaviour. Automated learning algorithms and anomaly detection procedures are necessary for this approach to discriminate suspicious activities based on individual user habits and behaviour.

When classifying suspicious user activity, a spontaneously adopted strategy is **supervised learning**. This approach leverages existing information and previous classifications of suspicious accounts, such as those already placed on blacklists or identified by rule-based detection systems. Features associated with suspended or blacklisted accounts can serve as examples for positive training, while features from accounts that are still enabled can be used as examples for negative training. After selecting a suitable supervised learning algorithm, the training phase proceeds using these identified labels.

Another potential approach for classifying suspicious activities on user accounts is **clustering**. Clustering involves grouping user accounts into homogeneous groups based on their activity patterns (such as frequency of user posts, time spent on the platform, and frequency of user logins). This can help identify suspicious activities that may involve multiple user accounts compromised by the same attacker, perhaps for coordinated spamming or publishing unwanted posts. Clustering can detect similarities, including hidden ones, within various user groups. Once accounts are grouped into clusters, analysts need to determine which clusters represent suspicious activity and which accounts within those clusters are involved in potential fraud.

Supervised Learning Pros and Cons

Supervised learning is a type of machine learning where algorithm training is conducted using an input dataset for which the desired output is already known. The algorithms are trained to identify relationships between training variables, optimising learning parameters based on known target variables (or labels). Examples include classification algorithms used for spam detection and regression algorithms, as well as k-Nearest Neighbors (k-NNs), Support Vector Machines (SVMs), Decision Trees, Random Forests, and Neural Networks (NNs). This approach is suitable when normal and anomalous behaviours can be reliably distinguished, such as in credit card fraud detection where future attempts might follow predefined schemes. Classification algorithms are particularly well-suited for tasks requiring discrete answers like spam or ham. In the context of suspicious activity classification, supervised learning can leverage existing blacklists or rule-based systems for training data.

However, adopting a supervised learning strategy carries methodological risks.

- **Cons:** A main problem is that algorithms trained this way may have difficulty recognising *new* cases of suspicious activity. This is because they are conditioned by previous classification labels, which might contain systematic errors. Retraining models to detect new forms requires introducing new classification rules, which risks amplifying previously introduced systematic errors. For instance, mistakenly including users based on geographical location in blacklists would introduce false positives that are then systematically fed into the models. To mitigate the distortive effect of false positives, appropriate weighing of samples is needed during subsequent training phases. Additionally, issues like unbalanced or mislabeled datasets, observed in data like credit card fraud, can reduce the effectiveness of supervised algorithms. While human operator feedback can address mislabeled datasets, this solution is often burdensome in terms of time and resources. Supervised learning can also suffer from exclusion bias, where representative samples of the population are left out of the datasets. Limiting the scope of algorithmic detectors to specific threats is one effective strategy to prevent dataset bias.

Clustering Pros and Cons

Clustering algorithms are a type of unsupervised learning where the algorithms aim to classify data independently without prior classification from an analyst. In cybersecurity, unsupervised learning algorithms, including clustering, are important for identifying *new* forms of malware attacks, frauds, and email spamming campaigns that haven't been previously detected. Clustering allows for the automatic identification of classes when class information is not known in advance and is considered fundamental in malware analysis and forensic analysis. Algorithms that utilise the concept of similarity, such as clustering algorithms, are well-suited for implementing anomaly detection solutions.

Clustering is particularly indicated for exploring latent and unknown patterns within data. When used for classifying suspicious activities on user accounts, clustering groups accounts based on their activity to detect similarities, even hidden ones, within user groups. This can help identify activities across multiple compromised accounts used by the same attacker. K-Means is mentioned as a clustering algorithm that is simple to use and offers high scalability, making it preferable for large datasets.

However, there are disadvantages and challenges associated with clustering.

- **Cons:** Care is needed in selecting the metrics used to define similarity between normal and anomalous traffic. Also, scoring systems and thresholds used in anomaly detection based on clustering require consideration of data ordering and distribution. It is necessary to carefully choose the specific clustering algorithm to use, as not all are effective for detecting suspicious activity. K-means, for example, requires the correct determination of the number of clusters (defining the parameter 'k') in advance. This requirement is often not well-suited for detecting suspicious user activities in practice because the correct number of groups is usually unknown. Furthermore, K-means algorithms do not work with features expressed in categorical or binary classification values.

USER AUTHENTICATION WITH KEYSTROKE RECOGNITION

User Authentication with Keystroke Dynamics

User authentication using keystroke dynamics is a method employed in cybersecurity to enhance the protection of user accounts and credentials. It is considered a form of **biometric recognition based on behaviour**. Unlike physical biometrics such as fingerprints or facial features, keystroke dynamics relies on patterns of behaviour that can be reliably associated with an individual user. This behavioural biometric is likened to the uniqueness found in freehand writing.

The foundation of keystroke dynamics lies in the **cadence and rhythm of keypress events**, which are considered unique to each individual. However, raw keypress data can be affected by external factors like interruptions, error corrections, or the use of special keys. To address this, the raw data from user typing is transformed into a dataset of features that accurately represent the user's keyboard dynamics, after cleaning away these random disturbances.

Key features used to define keystroke dynamics include various timing measurements extracted from the typing process:

- **Keydown-keydown**: The time interval between pressing one key and pressing the very next key.
- **Keyup-keydown**: The duration from releasing one key to pressing the subsequent key.
- **Hold**: The length of time a specific key is held down.

These extracted timing features are then used as input for algorithms designed to detect specific users or identify anomalous typing patterns.

Keystroke dynamics is particularly applicable to **anomaly detection**. The concept is that individuals attempting to authenticate with stolen or compromised passwords would exhibit different typing patterns compared to the legitimate user. This difference in keystroke dynamics can be used to identify and potentially block such impostors. Monitoring features like typing speed that are incompatible with human operators can indicate suspicious activity.

A scientific study titled "Comparing Anomaly-Detection Algorithms for Keystroke Dynamics" by Killourhy and Maxion explored the use of keystroke dynamics for anomaly detection. The study involved collecting keystroke data from 51 subjects typing 400 passwords each, which was then submitted to 14 different algorithms to evaluate their performance in user detection. The goal was to reliably identify impostors based on their distinct typing patterns.

Various AI and machine learning algorithms can be applied to classify and analyse keystroke dynamics data. These include algorithms like k-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Multilayer Perceptrons (MLP) [previous conversation]. The Multilayer Perceptron (MLP) is an Artificial Neural Network (ANN) and has shown high prediction accuracy (over 90%) in related biometric tasks such as facial recognition. MLP classifiers can be used for user detection with keystroke dynamics, showing "considerably better results" in prediction accuracy.

Coursera Signature Track

Coursera Signature Track stands out as one of the first concrete examples of user authentication leveraging keystroke dynamics. Introduced by Coursera some years ago, its primary purpose was to **verify the identity of students** taking tests to earn Statements of Accomplishment. This technology aimed to link a student's coursework to their real identity, enabling them to receive a verified certificate with a unique verification code for third-party validation.

Signature Track was designed to operate at the **large scale** characteristic of Coursera courses, which often enrol tens of thousands of students. This necessitated authentication and verification procedures that were both efficient and did not require significant intervention from instructors or staff.

A specific challenge Coursera faced was the temptation for students to share login credentials for homework completion. To counter this, Coursera implemented a **dual biometric and photographic authentication approach**. This combined **facial recognition** with the analysis of **typing patterns** associated with individual students. During the enrollment process, students were required to provide a webcam photo, a copy of an ID document, and type a short sentence on their keyboard. This process served to establish their unique biometric keystroke profile based on the cadence and rhythm of their typing. As with general keystroke dynamics analysis, this relied on extracting feature sets from the raw typing data to be used by machine learning algorithms.

KEYSTROKE DYNAMICS

Keystroke dynamics, also known as keystroke typing or keystroke recognition, is a method used in cybersecurity for user authentication and the protection of sensitive information and credentials. It is considered a form of **biometric recognition based on behaviour**, distinguishing it from physical biometrics like fingerprints or facial features. Instead, keystroke dynamics relies on behaviour patterns that can be reliably associated with an individual user, similar to the uniqueness found in freehand writing.

The basis of keystroke dynamics is the unique **cadence and rhythm of keypress events** for each student or user. However, the raw data from keypresses is not directly usable by machine learning algorithms because it can be affected by random external factors such as interruptions, error corrections, or the use of special function keys like Shift or Caps Lock. Therefore, the raw data representing user typing must be transformed into a dataset of features that accurately represent the user's keyboard dynamics, cleaning it from these disturbing factors.

Specific timing features extracted from the raw data are used to determine keystroke dynamics and fed to user detection algorithms:

- **Keydown-keydown (DD)**: The time that elapses between pressing one key and pressing the next consecutive key.
- **Keyup-keydown (UD)**: The time that elapses between releasing one key and pressing the subsequent key.
- **Hold (H)**: The time that elapses between the press and release of each key.

Keystroke dynamics can be applied to **anomaly detection**. The core idea is to detect suspicious user behaviour or identify impostors attempting to authenticate with compromised passwords. Impostors would exhibit different typing patterns compared to the legitimate users, allowing them to be identified and blocked. Monitoring features such as frequency and typing speed that are incompatible with human operators can indicate suspicious activity.

ANOMALY DETECTION WITH KEYSTROKE DYNAMICS

Keystroke dynamics, also known as keystroke typing or keystroke recognition, is a method used in cybersecurity for **user authentication** and the protection of sensitive information and credentials. It is categorised as a form of **biometric recognition based on behaviour**. Unlike physical biometrics (e.g., fingerprints, facial features), it relies on patterns of behaviour that can be reliably associated with an individual user, likened to the uniqueness of freehand writing.

The fundamental principle of keystroke dynamics is based on the unique **cadence and rhythm of keypress events** for each user. However, the raw data generated by user typing cannot be directly used by machine learning (ML) algorithms. This is because raw data can be affected by external factors such as interruptions, error corrections, or the use of special keys (e.g., Shift, Caps Lock). To overcome this, the raw data is transformed into a dataset of **features** that accurately represent the user's keyboard dynamics, after cleaning it from these disturbances.

Key timing features extracted from the raw data to define keystroke dynamics include:

- **Keydown-keydown (DD):** The time interval between pressing one key and pressing the next consecutive key.
- **Keyup-keydown (UD):** The time interval between releasing one key and pressing the subsequent key.
- **Hold (H):** The duration for which a specific key is held down.This is the time that elapses between the press and release of each key

These extracted timing features are then used as input for algorithms designed for user detection or identifying anomalous typing patterns.

Keystroke dynamics is particularly well-suited for **anomaly detection**. The core idea is that someone attempting to authenticate using compromised passwords, but who is not the legitimate user (an impostor), would exhibit typing patterns different from the genuine user. This difference in keystroke dynamics allows for the identification and potential blocking of such impostors. Monitoring features like typing speed and frequency that are incompatible with human operators can also indicate suspicious activity.

User Detection with Multilayer Perceptron (MLP)

Various machine learning and AI algorithms can be used for user detection based on keystroke dynamics data. Examples include KNN Classifier, Support Vector Machines (SVM), and Multilayer Perceptrons (MLP).

In the context of user detection using keystroke dynamics, the **Multilayer Perceptron (MLP)** classifier has shown **considerably better results in prediction accuracy**. In the example code provided, which uses the dataset from the Killourhy and Maxion study, applying different classifiers (KNN, linear SVM, MLP) to keystroke data showed that the MLP classifier achieved the highest accuracy, accounting for **over 90% prediction accuracy**.

The reason for MLP's superior performance in this task is that it represents an **Artificial Neural Network (ANN)**. ANNs are the fundamental elements of deep learning and contribute to the high potential of deep learning algorithms, enabling tasks like classifying large datasets, performing face and speech recognition, or even beating a world chess champion.

MLP overcomes the limitations of a single perceptron. While a single perceptron is essentially a binary linear classifier that provides accurate results only if the data is linearly separable, which is not the case in most real-world scenarios, an MLP is designed to handle non-linearly separable data.

An MLP is composed of **multiple layers of artificial neurons**, with each neuron typically implemented as a perceptron. An MLP can consist of three or more layers of fully connected artificial neurons, forming a feedforward network. Critically, an MLP has the capacity to approximate any continuous mathematical function. Its overall predictive power can be amplified by adding an arbitrary number of hidden layers.

MLP classifiers are also utilised in other biometric authentication methods, such as facial recognition. In the Eigenfaces facial recognition technique, for instance, an MLP classifier is applied after the dimensionality of the image data is reduced using Principal Component Analysis (PCA) to identify the most relevant features (Eigenfaces).

In summary, keystroke dynamics provides a behavioural biometric for user authentication and anomaly detection based on typing patterns. MLP classifiers, being ANNs, are particularly effective for user detection using keystroke dynamics data, demonstrating high prediction accuracy due to their ability to process complex, non-linear data patterns inherent in human typing rhythm.

BIOMETRIC AUTHENTICATION WITH FACIAL RECOGNITION

- Facial recognition is a method of **biometric authentication** used in cybersecurity, leveraging distinctive physical elements of a user's face
- This type of biometric procedure is increasingly common for protecting sensitive user information and credentials
- The growing use of facial recognition is facilitated by the spread of neural networks (NNs) and the availability of hardware like cameras on devices such as smartphones, tablets, and PCs
- While the concept of facial recognition isn't new, its application has increased significantly due to the rise in cyber threats and the need for more stringent identity verification beyond traditional methods like username/password
- An early practical example combining facial recognition with another biometric was **Coursera's Signature Track** technology, used to verify students' identities for online course tests. This system involved providing a webcam photo and ID during enrollment, along with typing a sentence to capture keystroke patterns.

- However, facial recognition faces several challenges and disadvantages that can impact its reliability.
- Environmental factors such as reflections, shadows, and incident light, as well as the angle of the face, can distort images and reduce reliability. These issues are particularly noticeable in uncontrolled environments like crowds, leading to a high number of false positives. Reliability is generally better in controlled settings.
- A fundamental assumption of biometric methods is uniqueness, meaning the evidence can be traced exclusively to one person; this is not always the case with facial recognition, partly due to lookalikes.
- Furthermore, faces can change over time due to factors like disease, stress, accidents, or aging, which can affect recognition accuracy.
- As datasets grow larger, the possibility of spurious correlations increases, requiring a disproportionately large amount of data to train algorithms reliably, which can hinder real-time implementation. Exhaustively comparing every new image against an entire archive is not feasible.
- Technically, facial recognition is approached as a **classification problem**, aiming to associate names with face images, distinct from facial detection which just identifies the presence of a face.
- Performing facial recognition requires comparing new images with those stored in an archive.
- A direct comparison of image feature vectors is impractical due to the high dimensionality and noise in image data; therefore, **dimensionality reduction** techniques are needed to isolate the relevant information for recognition .
- A common technique used is **Eigenfaces**, which relies on **Principal Component Analysis (PCA)** .
- **PCA** is an unsupervised dimensionality reduction algorithm. It identifies the most representative variables, called principal components, along which the data exhibits the greatest variance.
- This process involves calculating the covariance matrix of the data and identifying the largest Eigenvectors, which represent the axes of these principal components. Eigenvectors define direction in linear space, while Eigenvalues measure intensity. Multiplying an Eigenvector by a matrix (like the covariance matrix) scales its intensity but not its direction.
- The principal components derived from PCA are known as Eigenfaces, and each face image can be seen as a combination of these Eigenfaces. PCA is considered a simple technical method for reducing the dimensionality of large datasets, including images.
- For the classification part of the process, algorithms such as the **Multilayer Perceptron (MLP) classifier** can be employed.
- An MLP is a type of Artificial Neural Network (ANN) and a fundamental element of deep learning. It uses multiple layers of artificial neurons to overcome the limitations of a single perceptron and can approximate any

continuous function, with increased predictive power achieved by adding hidden layers.

- An example script demonstrates applying PCA for dimensionality reduction and then using an MLP classifier for facial recognition with the Labeled Faces in the Wild (LFW) dataset, providing metrics like precision, recall, and f1-score.
- Cloud platforms like IBM Cloud offer relevant services, such as their Visual recognition service, which allows apps to find faces in images/videos and be trained on specific datasets
- In summary, biometric authentication via facial recognition is a promising method leveraging AI techniques like ANNs and dimensionality reduction (e.g., PCA in Eigenfaces) for user authentication, but its practical implementation faces challenges regarding image quality, environmental conditions, the integrity of uniqueness, and the computational cost of training reliable models

PROS AND CONS OF BIOMETRIC AUTHENTICATION WITH FACIAL RECOGNITION

Biometric authentication with facial recognition is a growing area in cybersecurity. It is used to protect users' sensitive information and credentials. Facial recognition is considered a distinctive physical element that can be reliably traced back to a specific human user, alongside methods like iris scans, fingerprints, or voice recognition. These procedures are becoming increasingly common, partly due to the widespread availability of hardware peripherals like embedded cameras on devices such as smartphones, tablets, and PCs, and the diffusion of neural networks (NNs). The need for more stringent personal identity verification beyond traditional methods like IP addresses or username/password has contributed to this increase.

Pros:

- **Increasingly Common and Important:** Facial recognition is assuming an increasingly important role in cybersecurity for user authentication and protecting sensitive information.
- **Supplement or Replace Passwords:** It can be used as an authentication procedure to supplement or even replace traditional password-based authentication methods.
- **Leverages Available Technology:** Its growth benefits from the increasing diffusion of neural networks and the availability of camera hardware on common devices like smartphones, tablets, and PCs.

- **Seems Logical and Practical:** Given the pervasive diffusion of devices equipped with high-definition cameras, facial recognition appears to be a logical and practical solution for identity verification.
- **Provides Stringent Verification:** It offers a more stringent form of personal identity verification, which is needed when traditional checks based on IP addresses or username/password credentials are insufficient.
- **Supported by Cloud Services:** Cloud platforms like IBM Watson offer facial recognition services.
- **Broad Cybersecurity Application:** Biometric data, including face, are widely used in cybersecurity for authentication, authorization, and detection purposes.

Cons:

- **Reliability Challenges:** For reliable identification, images must not be distorted by environmental elements such as reflections, shadows, or incident light. The angle of the face also affects reliability.
- **Ineffectiveness in Uncontrolled Contexts:** These distortion problems are particularly evident when attempting facial recognition on images extracted from crowds, often resulting in a high number of false positives that make the method ineffective. Reliability is greater in controlled contexts where distortive factors can be minimised, and smaller when used "in the wild".
- **Uniqueness Assumption Limitations:** The underlying assumption of biometric procedures is uniqueness, but this is not always verified with facial recognition. This is due to factors like lookalikes and natural changes in an individual's face over time (due to disease, stress, accidents, or aging).
- **Scalability Issues with Growing Data:** As populations increase, the possibility of spurious correlations increases, and the amount of data required to improve recognition reliability grows disproportionately with the dataset size.
- **Difficulty in Training and Real-time Use:** The challenges mentioned above make it particularly difficult to reliably train facial recognition algorithms, hindering real-time use. Exhaustive verification comparing all possible combinations between archived evidence and new images is not feasible.
- **High Rate of False Positives:** Despite its apparent practicality, facial recognition procedures are prone to generating a high number of false positives.
- **Risks of Incorrect Profiling:** Aggregating personal data, including biometric data like face, can lead to profiling. Incorrect profiling can result in negative consequences, including reputational damage and legal sanctions (potentially under the GDPR's accountability principle) if decisions (like denying a financial contract or flagging a transaction as suspicious) are made based on this incorrect data. Serious consequences can occur if special categories of data like biometrics are processed incorrectly.

- **Experimental Research Area:** The application of AI to cybersecurity, including facial recognition, is an experimental research area that is not without problems.
- **Skepticism and Conservative Attitudes:** Insiders sometimes react with ambivalence, alternating between skepticism and conservative attitudes.

Technically, facial recognition is treated as a **classification problem**, distinguishing it from facial detection. It involves comparing new images with an archive. Due to the high dimensionality and noise in image data, direct comparison of feature vectors is impractical, necessitating **dimensionality reduction** techniques. The Eigenfaces technique, for instance, uses **Principal Component Analysis (PCA)** to identify the most relevant features (Eigenfaces). For classification, algorithms like the **Multilayer Perceptron (MLP) classifier**, a type of Artificial Neural Network (ANN), can be used, offering increased predictive power compared to simpler models. Despite these technical approaches, the practical challenges related to image quality, the uniqueness assumption, and training data size, particularly in uncontrolled environments, remain significant factors influencing the reliability of facial recognition for authentication.

EIGEN FACES FACIAL RECOGNITION

- Among the common techniques for facial recognition is one known as **Eigenfaces**.
- The name "Eigenfaces" comes from the procedures used in its implementation, which involve linear algebra.
- Technically, facial recognition is considered a **classification problem**. It aims to combine the names of faces with their corresponding images.
- This is distinct from facial detection, which simply identifies the presence of a face within an image.
- Facial recognition requires comparing new images of individuals with an archive of images representative of faces to which names are matched.
- A direct method of comparison, such as reducing images to feature vectors and calculating differences, is impractical due to the high number of comparisons needed in near real-time.
- Images naturally have a high number of dimensions (features, like pixels), which can contain a lot of irrelevant information or "white noise" for recognition purposes.
- To make reliable comparisons, it is necessary to **reduce the number of dimensions** to only those strictly relevant for recognition.
- The Eigenfaces technology is based on an **unsupervised dimensionality reduction algorithm** called **Principal Component Analysis (PCA)**.

- PCA makes it possible to identify the **representative variables**, also called principal components, of a dataset by selecting those along which the data exhibits the greatest variance.
- Identifying this axis of maximum variance involves calculating the covariance matrix of the data and identifying the largest **Eigenvectors** within this matrix. These Eigenvectors correspond to the axes associated with the main components, allowing for dimensionality reduction.
- The concept of Eigenvectors (and the associated concept of Eigenvalues) comes from linear algebra. Eigenvectors define direction in linear space, while Eigenvalues measure intensity. Multiplying an Eigenvector by a matrix (like the covariance matrix) scales its intensity but does not change its direction.
- To find the principal components within a covariance matrix, one looks for the Eigenvectors that correspond to the higher Eigenvalue values. Libraries like NumPy can be used for these calculations.
- The principal components derived from PCA are known as **Eigenfaces**.
- Each image in the dataset can be interpreted as a combination of these Eigenfaces.
- The process involves reducing the dimensionality of an image (from its numerous pixel-based features) to the main components (Eigenfaces), which are the features most relevant for recognition.
- For the classification step, after dimensionality reduction using PCA, an **MLP classifier** (Multilayer Perceptron, a type of Artificial Neural Network) can be used for image classification.
- An example script demonstrates applying PCA for dimensionality reduction and then using an MLP classifier for facial recognition on the Labeled Faces in the Wild (LFW) dataset, showing classification metrics like precision, recall, and F1 score.

DIMENSIONALITY REDUCTION WITH PCA (PRINCIPAL COMPONENT ANALYSIS)

- Images, by their nature, contain a **high number of dimensions**, often representing the pixel data. This can include a lot of irrelevant information, or "white noise," for tasks like facial recognition.
- Directly comparing images by considering all these dimensions and calculating differences is **impractical**, especially when dealing with a large number of images that need to be processed quickly.
- Furthermore, working with a high number of dimensions without a corresponding increase in relevant information can lead to what's known as the **"curse of dimensionality."** This phenomenon disperses the data in the increased space, weighing down computational effort and potentially reducing the reliability of predictive models.

- To make reliable comparisons for tasks like facial recognition, it is crucial to **reduce the number of dimensions** to include only those that are most relevant.
- **Principal Component Analysis (PCA)** is an **unsupervised dimensionality reduction algorithm** that is central to the Eigenfaces technology for facial recognition.
- PCA makes it possible to identify the representative variables (also called principal components) of a dataset, selecting those along which the data is more spread out.
- PCA works by identifying the **representative variables**, also called **principal components**, within a dataset.
- It achieves this by finding the dimensions or axes along which the data exhibits the **greatest variance**, meaning where the data is most spread out.
- Mathematically, identifying these axes involves calculating the **covariance matrix** of the dataset. The covariance matrix helps understand how different dimensions in the data vary together.
- Within the covariance matrix, the algorithm looks for the largest **Eigenvectors**. These Eigenvectors represent the directions corresponding to the axes of maximum variance, which are the principal components.
- The concepts of Eigenvectors and Eigenvalues originate from linear algebra. An Eigenvector defines a direction in a linear space, and when multiplied by a matrix (like the covariance matrix), its direction remains unchanged, but its magnitude is scaled by a scalar value called the **Eigenvalue**. Larger Eigenvalues correspond to the Eigenvectors that capture the most variance in the data.
- By selecting the Eigenvectors with the largest corresponding Eigenvalues, PCA allows for the **reduction of the dataset's dimensionality**.
- In the specific context of the Eigenfaces technique for facial recognition, the **principal components** derived through PCA are known as **Eigenfaces**.
- Essentially, this process transforms an image from its original high-dimensional representation (based on many pixels) into a lower-dimensional representation based on these Eigenfaces, which capture the most significant features for differentiating faces.
- Reducing dataset dimensionality using PCA can significantly **improve the performance** of algorithms used for subsequent tasks, such as classification.

Variance: This measures the degree of dispersion existing within the data and is represented by the average of the deviations of the data with respect to their average, as follows:

$$\sigma^2 = \frac{\Sigma(x_i - \mu_x)^2}{(N-1)}$$

Covariance: This measures the degree of linear correlation between two variables; it is represented mathematically as follows:

$$cov(X, Y) = \frac{\Sigma(x_i - \mu_x)(y_i - \mu_y)}{(N-1)}$$

Covariance matrix: This is the matrix that contains the covariances calculated on each ordered pair of data belonging to a dataset.