MULTIDISCIPLINARY SCIENCE JOURNAL

TECHINICAL NOTE
Published Online: July 14, 2023
https://doi.org/10.31893/multiscience.2024009

Integrated MLOps and EEG techniques for enhanced crime detection and prevention

Akash Kathole[a] | **Sagar Shinde**[b] | **Lalitkumar Wadhwa**[b]

[a]Department of Artificial Intelligence, Nutan College of Engineering & Research, India.
[b]Department of Electronics & Telecommunication, Dr. D. Y. Patil Institute of Technology, India.

Abstract The use of machine learning and artificial intelligence in crime detection has gained significant attention in recent years. This research paper explores the potential of MLOps techniques in identifying criminal behavior through the analysis of vehicle plate numbers (Tripathi et al 2021), person detection, and object behavior detection. This paper presents a literature review of studies that have investigated the use of MLOps in crime detection and highlight the potential of these techniques to be used in criminal investigations. Specifically, the use of machine learning models for vehicle plate number recognition, person detection through image and video analysis, and object behavior detection through image and video analysis has been discussed. The paper presents a framework for integrating MLOps techniques into criminal investigations, which involves a combination of data acquisition, data preprocessing, model development, model training and testing, and deployment. Additionally, the discussion includes the ethical implications of using MLOps techniques in criminal investigations and highlights the need for transparency and fairness in model development and deployment. The detection and prevention of criminal behavior is a critical issue for society. In recent years, there has been growing interest in the use of electroencephalogram (EEG) techniques to detect and predict criminal behavior. This research paper explores the potential of EEG techniques as a means of detecting crime by analyzing brainwave activity. Specifically, this paper examines the use of alpha and beta waves in identifying deceptive or abnormal behavior (Xie et al 2022). The paper presents a literature review of studies that have investigated the relationship between EEG signals and criminal behavior and highlight the potential of these techniques to be used in criminal investigations (Xie et al 2022). Furthermore, a framework is proposed for integrating EEG techniques into criminal investigations, which involves a combination of data acquisition, analysis, and interpretation. Finally, the ethical implications of using EEG techniques in criminal investigations and the need for further research in this area have been discussed.

Keywords: machine learning, artificial intelligence, brainwave, criminal

1. Introduction

Detecting crime thinking or deception is a critical task for law enforcement agencies. Various methods have been developed for this purpose, including traditional interrogation techniques, polygraph tests, and more recently, neuro imaging methods such as electroencephalography (EEG). This research paper explores the use of EEG and machine learning operations (MLOps) for crime detection and provide an overview of the current state of research in this field (Hamdy et al 2015). Crime is a significant social and economic problem although the incidence of certain categories of crime is important, the overall crime situation in the country. It has improved over the last decade. Many older studies from other countries have shown positive or negative correlations between crime and economic growth. At the rural level, the relationship between different classifications of crime and economic growth using the irregular modeling technique of chartered add-on models. Such a modeling approach helps to understand how different categories of crime affect GDP.

EEG is a non-invasive technique that measures the electrical activity of the brain using sensors attached to the scalp. The technique has been used to study a wide range of brain functions, including attention, perception, memory and decision-making. In recent years, researchers have started to explore the use of EEG for detecting deception and crime thinking. MLOps is a set of practices and tools for managing and deploying machine learning models. It involves a combination of software engineering, data engineering, and machine learning expertise to create scalable and reliable machine learning systems.

Computer vision and machine learning it is now possible to develop systems that can assist in the identification and tracking of suspects and vehicles. This research paper explores the use of object detection, vehicle plate recognition, and face recognition for crime detection (Shah et al 2021) and provide an overview of the current state of research in this field. An electroencephalogram is a method used to measure the electrical activity of the brain. In EEG, electrodes are usually placed on the patient's scalp and are used primarily to detect the electrical activity of neurons, to detect the signal. Groups of neurons

 Received: April 13, 2023 | Accepted: June 17, 2023

are active. It calculates changes in the postsynaptic potential, which is restored by the release of neurotransmitters into receptors on the postsynaptic membrane. It is used to calculate brain activity during events such as crime. A measure of the brain's involuntary activity in the absence of what they're thinking - or the task of a particular event. It is the specific electrical activity combined with the event. This is used to analyze the patient's behavior and give some clues about what they plan to do about various other brain disorders. The time taken to measure brain activity by EEG is milliseconds. This can be done by recognizing emotions. Using EEG, EEG-based models help doctors detect emotions in critical patients. It records brain activity in real time. Doctors use EEG to evaluate a suspected case. It is used to monitor suspected pregnancy, what is the feeling. MLOps is a set of methods for developing and deploying machine learning operations and machine learning models. Enables version management of both data as well as model artifacts, CI CD executes the development and release cycle of ML models in the ecosystem, helping to monitor the performance of models in production to maintain effectiveness.

Crime detection by examining EEG signals and using MLOps approach as well as some small techniques. Emotion detection based on brain signals is one of the best methods to detect human emotions and stress, giving accurate results (Mane and Shinde 2022). Brain wave or signal based system along with EEG signal based system can help in detecting various disorders and disabilities. It can help detect human mental stress and emotions with sentiment analysis (Mane and Shinde 2022). Perform various classifications to identify emotions and behaviors using EEG datasets. It is a crime forecast that uses information from the past and, in turn, predicts future crime by area and time. A crime prediction framework uses recorded information and examines the information using some structural strategy and then anticipates different types of crime (Table 1).

Table 1 Categorization of crime with example.

Category	Examples
Misinformation Detection	Web Deals, Protection Misinformation, MasterCard Coercion
Atrocious Crime	Murder, Assault (Mahmud et al 2016; Yang et al 2011)
Traffic Violation	Violating laws governing movement of vehicles on the road (Prabakaran and Mitra 2018)
Sexual Assault (Mahmud et al 2016)	Unwanted physical contact that is associated with danger or shock

The question arises that how to combine machine learning, MLOps, EEG, computer vision Legal enforcement speed Organizations or the powers that have the ability to identify, avert, and resolve criminal activity with greater precision and speed (Shah et al 2021).

2. Background and Literature Review

Criminal activities have been a major problem in the world for a long time. Law enforcement agencies have been using various techniques and technologies to detect and prevent crimes, but the increasing complexity and frequency of criminal activities have made it difficult for them to keep up. The emergence of artificial intelligence (AI) has provided new opportunities for law enforcement agencies to improve their crime detection capabilities.

AI has the potential to improve the accuracy and speed of crime detection, as well as the ability to identify patterns and predict future criminal activity. AI algorithms can process vast amounts of data from various sources such as CCTV cameras (Shah et al 2021), social media (Rony et al 2020), and other digital devices to identify potential criminal activities. By analyzing this data, law enforcement agencies can take proactive measures to prevent criminal activities before they happen (David and Suruliandi 2017).

However, the use of AI in crime detection raises concerns about privacy and bias. It is crucial to ensure that the use of AI in law enforcement is transparent, ethical, and respects individual rights. Nonetheless, the potential benefits of AI in crime detection make it an exciting area of research and development. The use of EEG (Electroencephalography) technique in detecting crime is a relatively new and promising field of research. The EEG technique measures the electrical activity of the brain through sensors placed on the scalp, and this data can be used to identify patterns in brain activity that are associated with specific actions or behaviors. Several studies have shown that EEG signals can be used to identify brain patterns associated with lying, deception, and other criminal behaviors. These patterns can be detected through changes in brain waves, such as an increase in the amplitude or frequency of certain waves. One potential application of EEG technology in crime detection is in interrogations. By monitoring the brain activity of a suspect during an interrogation, law enforcement could potentially detect if the suspect is lying or withholding information. Additionally, EEG technology could be used to identify suspects who are predisposed to violent behavior, enabling law enforcement to take proactive measures to prevent criminal activities. While EEG technology is still in its early stages of development, it has the potential to revolutionize the field of crime detection. However, there are also ethical concerns surrounding the use of this technology, particularly around privacy and the possibility of false accusations. Nonetheless, continued research and development in this field could lead to exciting new possibilities for law enforcement and crime prevention.

It compares the effectiveness of several classifiers in recognizing emotions using EEG signals. The paper demonstrates experiment with five classifiers and find that the random forest classifier provides the highest accuracy. It discusses the application of machine learning in predicting crime and experiments with neural networks and various machine learning techniques to predict crime, and the results show that the neural network approach provides the highest accuracy (Keyvanpoura et al 2010). provide a survey of crime detection and prediction techniques. The work discusses various techniques, such as data mining (Rony et al 2020), machine learning, and computer vision that can be used for crime detection and prediction. Propose a crime prediction and strategy direction service called CRIMECAST (Mahmud et al 2016). Their service uses machine learning and data mining techniques to predict crime (Rony et al 2020) and provide direction for law enforcement agencies. Discuss the use of data mining techniques for crime detection. The research experiments with four data mining algorithms and finds that the decision tree algorithm provides the highest accuracy. Propose a machine learning and computer vision approach to crime prediction and prevention (Mahmud et al 2016). The paper shows experiments with several machine learning algorithms and find that the support vector machine (SVM) (Guo and Suvorov 2022) algorithm provides the highest accuracy. Proposes a data mining approach to crime pattern detection. He uses the Apriori algorithm to mine the crime data (Gendre et al 2022) and identify the crime patterns (Varun 2019). Propose a criminal act detection and identification model using machine learning techniques (Safat et al 2021). The work demonstrates experiment with three machine learning algorithms and find that the decision tree algorithm provides the highest accuracy. Proposes a criminal behavior analysis method based on data mining technology. He uses the K-means clustering algorithm to analyze criminal behavior and identify the crime patterns (Varun 2019). Propose a machine learning and OpenCV approach to criminal and crime detection. The work demonstrates experiment with the k-nearest neighbors (KNN) algorithm and find that it provides the highest accuracy. Propose a general framework for crime matching (Keyvanpoura et al 2010) and detection using data mining techniques. (Rony et al 2020) The authors highlight the importance of crime detection and investigation in modern law enforcement and discuss the limitations of traditional methods. The paper proposes a framework that combines several data mining techniques, including clustering and classification, to identify patterns in crime data and detect potential criminal activity. The authors report promising results in their experiments and suggest that the proposed framework could be used to improve crime detection and investigation in law enforcement. Presents an empirical analysis of crime prediction and forecasting using machine learning and deep learning techniques. The authors aim to develop a system that can accurately predict and forecast crime rates in a given area. The paper proposes a system that uses several machine learning and deep learning algorithms, including SVM (Support Vector Machine), LSTM (Long Short-Term Memory), and CNN (Convolutional Neural Network) (Keyvanpoura et al 2010), to analyze crime data and make predictions. The authors report promising results in their experiments and suggest that the proposed system could be used to improve crime prediction and forecasting in law enforcement. Propose a novel approach to modeling the effect of streetscape environment on crime using street view images and interpretable machine learning techniques (Xie et al 2022). The authors aim to address the limitations of traditional methods for modeling the effect of the environment on crime, which often rely on subjective assessments. The proposed approach uses street view images and interpretable machine learning techniques (Xie et al 2022), including decision trees and linear models, to analyze the impact of the streetscape environment on crime (Xie et al 2022). The authors report promising results in their experiments and suggest that the proposed approach could be used to improve crime prevention and urban planning. In this paper, the authors propose a system for criminal identification using deep learning and convolutional neural networks (CNNs) (Keyvanpoura et al 2010). The system is designed to identify suspects in criminal cases by analyzing CCTV footage. The authors highlight the limitations of traditional methods for criminal identification, which often rely on manual inspection of CCTV footage. The proposed system uses CNNs to automatically extract features from the video frames and identify potential suspects. The authors report promising results in their experiments and suggest that the proposed system could be used to improve criminal identification in law enforcement (Shah et al 2021).

3. Proposed Methodology

The previous section investigated recent studies that have used EEG and MLOps to detect crime thinking or deception. This section examines the experimental design, data collection, and machine learning techniques used in these studies to understand the strengths and limitation of the approach.

In the block diagram shown in Fig. 1, first get data from various resources like maximum data collected from hospital such as behavior of patients, how are they injured, which part are damage, who is admitted the patient, previous record of patient, Data published by the Government of the Autonomous city of Buenos Aires (Forradellas et al 2020), National crime records Bureau of India from 2001 to 2012, Hostpots and Prioritize, (Mahmud et al 2016) The Indian crime database, sourced from the Indian National Crime Records, comprises data on criminal activities, criminal scenes, social media (Rony et al 2020; Hamdy et al 2015), and images of victims, suspects, weapons, and location (Hamdy et al 2015). The dataset also includes other criminal-related aspects, such as age, previous arrests, M.O., countries visited (Li 2016), place of birth, and average ATM card usage (Hamdy et al 2015).

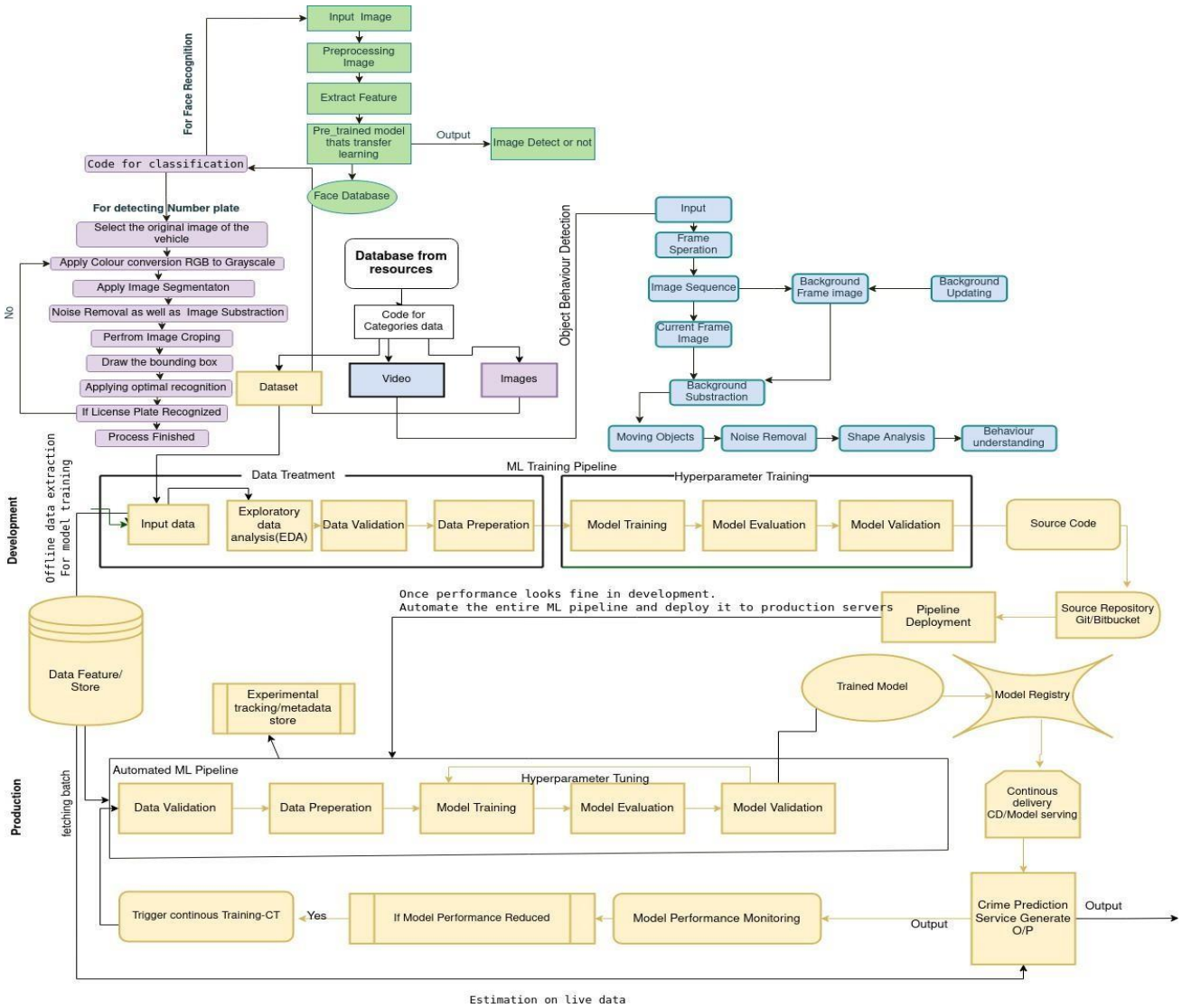

Figure 1 Categorizing data for Machine Learning models and MLOps

- For vehicle: from parking areas like entries of cars and bikes number plates (Rony et al 2020; Shah et al 2021), street light CCTV, vehicle that also collected from police complaint (David and Suruliandi 2017), record of vehicle stolen and their license number, location of vehicle stolen (Tripathi et al 2021), model of vehicle, RC number etc., picture of vehicle. Then generate code on it and that code categorizes the image, video, database, live stream etc. Then after classification the input will be relevant. Especially the model for Vehicle Number Detection.
- Electroencephalogram or EEG: This is a technique used to measure the electrical activity of the brain. In EEG, electrodes are usually placed on the patient's scalp and are primarily used to find the electrical activity of neurons, when they find the signals produced. Groups of neurons are active. It records signals from a small area of the brain around each electrode. It measures changes in the postsynaptic potential or membrane potential, which is stimulated by the binding of neurotransmitters to receptors on the postsynaptic membrane. It is used to measure brain activity during events such as crime. A measure of spontaneous brain activity in the absence of a task or a demonstration of what they are conceiving about -- or a specific event. It is the specific electrical activity associated with an event. Sometimes called event-related probability. EEG has many clinical applications. For example, it is used to analyze a patient's behavior and give some clues about what they are planning, among other various brain disorders. The time taken to measure brain activity by EEG is milliseconds. This can be done by identifying emotions. Using EEG. EEGbased models help doctors find the emotions of patients in critical. It records brain task in real time. Doctors use EEG to appraise a suspected case. It is used to monitor what suspects are conceiving, feeling.

What is actually EEG Recording?

It shows Multiple Waves and shapes of waves are depends on how is it currently brain are active. This is different when every wave awake for providing information about activity of nerve cells at area of cerebrum or Brain.

Table 2 Brain Wave types with respective frequency.

Brain Wave Types	Frequency Range	Description
Alpha Waves	8Hz – 13Hz	Occur when resting with closed eyes
Beta Waves	14Hz – 30Hz	Occur when eyes are open and senses are restored, mentally active
Gamma Waves	Over 30Hz	Occur when very alert and learning something new
Theta Waves	4Hz to 7Hz	Occur when falling asleep or feeling very weary
Delta Waves	0.5Hz to 3.5Hz	Occur during deep sleep

Everyone's baseline has different EEG pattern. This variation is greater in children as well as wave pattern are very slow, less regular than adult. If it is confirmed that the suspect has committed some wrongdoing (Gendre et al 2022), the following considerations must be made.

- Finding misrepresentation: Different types of fraud include check extortion, web deals, protection misrepresentation, Mastercard extortion, check misrepresentation which means obtaining charge card data through different means (Gendre et al 2022).
- Awful Crime: Focus involves two wrongful acts e.g., murder, torture (Gendre et al 2022; Mahmud et al 2016).
- Sexual harassment: The contact is actually made to the extent of having knowledge of the danger involved. Assisting the police in tracking down and arresting criminals. Information gathered by the police through surveillance, participant observation, wiretapping etc. on their personal bio teristics, social habits etc. (Gendre et al 2022; Mahmud et al 2016).
- Planned Analysis: This detailed analysis of criminal incidents or activities involves looking at common characteristics such as when, how, where the incident occurred, to help identify potential suspects and develop a design for case clearance.
- Strategic Analysis. It uses statistical methods to examine electronic databases containing large numbers of records.
- Management Analysis: This process selects the key findings from the previous analysis and formats them proper for the target audience.
- Component for checking similarity (Keyvanpoura et al 2010): The task of this component is to assess the distinctiveness between two specified entities, such as identifying resemblances between patterns of motion, comparing the proximity of related entities, matching trajectories of motion, or correlating visual features such as pictures, sounds, and videos (Hamdy et al 2015).
- The component for analyzing visual contents: It is isolating visual components and organizing them into feature and action sets (Hamdy et al 2015).
- Distance Evaluator: The Proximity Analyzer is accountable for computing the closeness metrics of a motion and assessing similarity in proximity (Hamdy et al 2015).
- Association Analyzer Component: It is the backbone of all operation. Function of this is Discovering connections between disparate clues gleaned from the system and presenting deeper, more inclusive conclusions that determine if a particular course of action constitutes questionable conduct (Hamdy et al 2015).
- Component of Criminal History Records: This component serves as a source of learning about suspicious items and actions (Hamdy et al 2015).
- Approach of Operation Information Elements (modus operandi): it's utilized to discover typical trends of unlawful conduct (Hamdy et al 2015).
- Tracking Data Component: The primary purpose is to collect location data from smartphones or surveillance cameras (Hamdy et al 2015).

MLOps stands for Machine Learning Operations and is a set of methods for developing and deploying machine learning models. Enables version management of both data as well as model artifacts, executes the development and release cycle of ML models in the CICD ecosystem, helps monitor performance of models in production to maintain effectiveness.

Need for MLOps

Lets understand this with an example. For the last couple of years there's an increase in lot of scams around the United States where scammers pretend to be a Tech Support Group from Microsoft, Amazon, NordVPN, etc. and scam elderly people for thousands of dollars. Due to which there's been a lot of complaints coming from the residents. Authorities found that in nearly 20-30% of those cases were not actually scammers but real service providers who were unsuccessful in providing respective service because of some technical difficulties and thus residents thought it was a scam call (Hamdy et al 2015). So,

task is to determine whether a particular case is of a scam or not, which will help authorities to take fast actions against those scammers.

- Just like any Machine Learning Project, First, collect the data from the past 3 months victims and authorities where obtain attributes such as Asked_Remote_Access
- Opened_Bank_Website
- Location_Of_Call (Hamdy et al 2015)
- Money_Asked, etc. and train a Machine Learning model to predict whether or not a case is of scam or not.

Deployed a model somewhere in a secure cloud, and authorities have started using it.It's been a month and authorities have found ML model useful and they even award for this brilliancy! But just after weeks something bad happened...

Authorities noticed that the lot of cases which were flagged as not scam, came out to be actual scams So they investigated those victims and found that well scammers have changed their ways and they've started scamming people with different strategies i.e., Input data has been changed, it's no more Opened_Bank_Website cuz now its UPI_PAYMENTS and the model is not trained on this change in the data.

This concept is also known as Model Drift Model drift refers to the degradation of model performance due to changes in data and relationships between input and output variables. It is relatively common for model drift to impact an organization negatively over time or sometimes suddenly.

AHHH which means now again have to

- Collect and integrate the new data
- Train model on new data
- Track training
- Experimentation with model
- Validate model
- Deploy model
- Monitor model

because never know when scammers will learn new strategies and the model might fail again and if it fails then have to perform all those steps again but do it again right XD?

Well don't worry cause here's where MLOps comes into the picture MLOps will provide an automated way to do all those steps easily! It can help us in

- Model Development (Designing and Training Model on the new data)
- Continuous integration and Continuous Deployment (Deploying that model without any interruptions)
- Monitoring (Checking if scammers have found a new strategies)
- Validation (Validating the results obtained)

ML Experiments: The process of creating the ML model; The whole process, in which the model builds and optimizes, Experiment running: each test in the ML experiment; Each run is within a ml experiment, Run the artwork: Any file related to the ML run: Examples include the model itself, package versions etc. Each work of art is associated with an experiment, Metadata experimentation: The metadata is bound to every experiment.

Experiment tracking: Keeping track of all relevant information from the ML experiment; according to the experiment. Experiments help in tracking reproducibility, organization and optimization tracking experiments in a spreadsheet helps but falls short of all key pointsTracking experiments with MLflow. Mlflow conducts experiments in runs and keeps track of any variables that can affect the model as well as its outcome; Such as: parameters, metrics, metadata, the model itself...MLFlow automatically logs in to additional information about each run such as: source code, git commit, start and finish time, and author. commands to run the MLflow UI locally: mlflow ui --backend-store-uri sqlite:///mlflow.dbThe backend storage is essential to access the features of MLflow, in this command if use a SQLite backend with the file mlflow.db in the current running repository. This URI is also given later to the MLflow PythonAPI mlflow.set_tracking_uri. By accessing the provided local url can access the UI. Within this UI have access to MLflow features. In addition to the backend URI, add an artifact root directory where store the artifacts for runs, this is done by adding a --default-artifact-root paramater: mlflow ui --backend-store-uri sqlite:///mlflow.db --default-

1. Hyperparameter Optimization Tracking: Hyperopt optimization can track each optimization run driven by hyperopt, by wrapping the objective inside the mlflow.start_run () block. If then log in to the parameters passed by the hyperopt as well as the metric. In this block, the search space and objective were defined rather than running the optimizer. The training and validation block mlflow.start_run () with the wrap in and using the log_metric, log_params , RMSE. Log the parameters used. Compare each run of the optimizer in the UI and their metrics and parameters. One can also see how different parameters affect the RMSE using a parallel coordinate plot, a scatter plot (1 parameter at a time) and a contour plot.
2. Autologging: Instead of logging parameters "manually" by specifying logged parameters and passing them. use the auto logging feature in mflow. There are two ways to use autologging mlflow.autolog() or by enabling a framework-

specific autologger; XG Boost with XGBoost:mlflow.xgboost.autolog(). The autologger then not only stores the model parameters for ease of use, but it also stores other files inside the model (can be specified) folder in experiment artifacts folder, these files include:conda.yaml and requirements.txt: files that define the current environment to use conda or pip respectively, MLflow file under MLmodel for organization, Other framework-specific files such as the model itself.

Saving Models: use MLflow to log whole models for storage to do this if add a line to with mlflow.start_run() block:mlflow.<framework>.log_model(model, artifact_path="models_mlflow")

3. Loading the model: the model to make predictions in several ways as needed: load the model as Spark UDF (User Defined Function) to use with Spark DataFrame, load the model as an MLflow PyFuncModel structure, then use it to infer data from a Pandas DataFrame, NumPy Array, or SciPy sparse array. The resulting interface is common to all models in all frameworks, load the model as is, i.e. load and treat the XGBoost model as an XGBoost model.
4. Model Registration: Just as MLflow helps us store, compare and transact ML experiment runs. It also allows us to store and categorize models. Although it is possible to manually store the models in the folder structure, this is difficult to do and leaves us open to errors. MLFlow deals with this by using a model registry, where models can be stored and labeled according to their position in the project.
5. Storing models in the registry: To register a model using the UI, select Register, and then select "Register Model". There can either register a new power by selecting the "Models" tab and selecting Registry and view its top.
6. Promoting and demoting people in the registry: All in the registry are labeled as staging, production, or archive. Promoting and demoting people can be done by selecting Registri and selecting the "Stage" menu option at the bottom of the top.

   Workflow Orchestration: It's a set of tools that schedule and monitor the work want to accomplish. E.g., Scheduling ML model training, Example pipeline:

   PostgresQL -> Parquet -> Pandas -> Sklearn -> mlflow

   | Rest API | Flask (if deploying)

   A pipeline can have random points of failure. The goal of workflow orchestration is to minimize errors and fail gracefully. Failure points are more common in more interconnected pipelines (different pipelines connected to each other). 90% of time is spent on Negative Engineering: Retry when APIs go down, Malformed data, Notification, Observability in failure, Conditional failure logic, Expiration. Prefect's goal is to reduce this time to increase productivity.
7. Prefect: An open-source workflow orchestration framework to eliminate negative engineering Open source, Python-based, Modern data stack, Native Dask integration, very active community, Prefect Cloud/Prefect Server -> Cloud is Prefect, the server is self-hosted. Prefect Orion (aka Prefect 2.0) is a complete take on Prefect 1.0; No backward/forward compatibility.

   It's in beta Current Status of Prefect:

   Prefect Core (1.0)

   Prefect Orion (2.0 Beta)

   Prefect uses decorators to wrap code.
8. To deploy the notebook: do not deploy notebooks and if deployed they are deployed as a single step. Thus notebooks are refactored into scripts for deployment. Adding Prefect Flow: implement the prefect in code by wrapping a workflow function (which fetches the data, preprocesses it, trains the model... etc.) with the @flow decorator: This enables additional logging. A main function is one that is usually placed in an if "__name"" == "__main__": block wrapped as a function. (The name doesn't matter). Multiple streams can be placed in a single file. open the Prefect UI (in Orion) using Prefect Orion to spin up a localhost instance. The UI contains various detailed information and logs about each flow run and process errors as well as stacktraces and task flow errors.
9. Parameter type validation: If Orion Flow receives a bad parameter type, instead of the flow running and inevitably failing, it will not run the flow at all and output a failed run to save computation time.
10. Remote prefect Orion deployment: To deploy Prefect remotely. The remote VM needs to open some ports: Connection

    Type Port, HTTP [80],HTTPS [443],TCP 4200,UDP 4200,

    i) How to do it in AWS is how to do it in GCP

    (source can be set as "anywhere" for AWS or "0.0.0.0/0" for GCP)

    To launch Prefect Orion, follow these steps :

    - pip install prefect==2.0b5 (replace with the most recent 2.0 version on pip)
    - Set PREFECT_ORION_UI_API_URL with: prefect configuration set PREFECT_ORION_UI_API_URL="http://<external-ip>:4200/api"
    - start orion: prefect orion start --host 0.0.0.0

- On the local machine, configure the API: prefect config set PREFECT_API_URL="http://<external-ip>:4200/api"

The remote UI will be visible on port 4200. Example: http://1.2.3.4:4200

The variables will appear set with the prefect configuration view.

If Prefect UI_API_URL or PREFECT_API_URL is already set to the old IP address, unset the variable with:

Prefect Configuration Unset PREFECT_ORION_UI_API_URL

And then set the key as described above. (replace PREFECT_ORION_UI_API_URL with PREFECT_API_URL to reset PREFECT_API_URL)

Now when running the script with Prefect Flow, the data should be logged to the remote VM Prefect instance.

Using Prefect Cloud: Instead of running Prefect on a VM itself, use Prefect's cloud service at https://beta.prefect.io which provides token login in addition to all other Prefect features.

Defining storage for the prefect: use prefect storage ls to see prefect current configured storage. By default, Prefect has no storage set and stores results for runs in a temporary directory in the runtime environment. To create storage, use Prefect Storage Create and then select the storage type want.

Creating and configuring AWS S3 storage: First create an S3 bucket; Search for S3 in the "Services" search bar and select "S3".

1. Adding a user with S3 permissions to AWS:

To access the S3 bucket with Prefect, need to add a new user with S3 permissions:

i. Adding a user: To create a "user" that Prefect will use to access the S3 bucket, open the drop-down menu next to the account name in the upper right and select "Security Credentials">"Users" (left menu)) >"Add Users".
ii. Add a new user (e.g., Prefect) with an AWS Access Type of "Programmatic Access" and select "Next: Permissions".
iii. To add a group with S3 permissions and add user to it; Select "Create Group" and give it a name (eg: S3-FullAccess) then find "S3FullAccess" in policies and select it (if click on it, it will redirect to the policy details and have to start over).
iv. Select the new group and click "Next: Tags" (put tags here if want), then "Next: Review" then "Create User". Do not close this window! Access secret key once lost cannot be recovered!
v. Create a new Prefect Storage with Prefect Storage and select Amazon S3.
vi. Name the S3 bucket created earlier.
vii. Copy the access key from the user window and paste it when prompted "AWS Access Key ID".
viii. Copy the Secret Access Key from the User window and paste it when prompted "AWS Secret Access Key".
ix. Skip session token, profile name, region name (press enter).
x. Choose a "locally" unique name for the configuration.
xi. When prompted if want to set it as default, select Y.

A flow is a flow. Schedule is the schedule on which run the stream. For example here every 5 minutes run main. Tags are tags associated with streams. They can be used for example for filtering. In this case flow_runner specifies that the flow will only be run locally; i.e., not on Kubernetes or Docker containers. To create the deployment, use: prefect deployment create prefect_deploy.py It simply creates the deployment and schedules the runs. Don't know how to run it. To run them, use job queues.

Work Queues: A job queue is a queue that will notify its affiliated agents to run scheduled runs.

Use the Prefect UI to create a new task queue and select Task Queue in the side panel (a name is required to create a queue), it is possible to filter by tags. A window will pop up containing the commands used to add agents to the work-queue; Prefect Agent Start <UUID>. Since the UUID is the UUID of the work queue. can check the status of the work-queue using the prefect work-queue preview <UUID> where it will show the scheduled runs. Adding a local agent: An agent is a scheduled run for a job queue. It checks every 5 seconds if there is a job in the job queue, fetches the flow file from storage and runs it. To run the agent use the command given on the local computer in the work-queue page: prefect agent start <UUID>.

2. Deployment: There are 2 types / paradigms or deployments: The type of deployment depends on how the prediction result is desired. Do an offline batch deployment of the model that runs intermittently while waiting, say, an hour or a day for the prediction result. On the other hand, if prediction is needed in real time it must be an online deployment where the model is always ready and running on the computer to serve. Again, when it comes to online deployment, depending on the use case, deploy the model as a web service or a streaming service.
3. Web Service: Wrap the model in a web service where the model can be loaded and served to provide predictions in REST API calls. For the entire set of data received in an API call, the output from the model is sent in a one: one fashion in response.
4. Streaming: Runs in a producer and consumer model where the producer pushes information to an event stream and the consumer listens to the stream to get updates.

Batch Deployment, Offline: If wait a bit to get predictions. Then periodically estimate new data. There is a database and scoring job. A scoring job periodically pulls data from the database and runs a model on it. The result is written to the prediction DB.
Example: brainstorming task.

5. Online: The model is always available for estimation. There are two ways to deploy an online model:
   Web Services: Example: crime scene prediction (Tripathi et al 2021). The app requires an instant estimate. The backend sends data to the model and the model replies with a predicted result.
   The relationship between the client (backend in this case) and the model is 1x1.
6. Flow: A producer(s) and a consumer. The producer pushes some data into the data stream and the consumer receives the data. Customers can then estimate multiple variables from the same data stream. Also run a simple model as a web service with a backend. Then if the user agrees to push data into the data stream with events and a more accurate prediction model (e.g., C1) is run, it will be pushed back to the backend. Customers can push their predictions to the prediction stream and decision.
7. Deploying as a web service: Following are the steps to deploy as a web service:
   i. Using Python environment to train/test the model using pipenv.
   ii. Rewriting the prediction script and wrapping it with a backend (flask is used here).
   iii. Creating a docker container and placing guess backend with a python environment.
8. Flask Application: Flask input and output JSON files. There are two functions used to handle JSON files: jsonify(D) converts the dictionary D to JSON, Both are imported using `request.get_json() Reads the JSON passed to the app. From the Flask import request, jsonify wrap the 3 guesses in a function and build an app on it: Docker container: reproducibility, scalability, security (it's connected to the internet)... etc To deploy predictor in a docker container, to use pipfile and Pipfile.lock files, to start a pipenv environment, copy the predict.py file created earlier, then run gunicorn WGSI. This is done via the Dockerfile.
9. To deploy as a stream: The core of deployment in streaming is 4 components:
   Event Stream: Where events are continuously pushed
   Stream Data: All data passed through a stream consists of two components:
   a. Event: An event has at least two fields. The "event" itself, ie: the message that triggers the consumer and the "data" payload that the consumer will carry. It can also take attributes for events.
   b. Context: which contains metadata.
   Producers: They create events that are pushed to the event stream
   Consumers: They receive and consume data from event streams; Process, estimate...etc
   Production and production of programs and data is handled by the backend. The other 2 components are usually hosted using
   online services. Popular services include:
   Event Stream: Kafka, AWS Kinesis, G PubSub
   Customers: AWS Lambda, G Cloud Functions
   Example workflow: (wrong prediction on backend / correct prediction on stream; very common workflow)
   The backend receives unconfirmed crime information by the user. Sends information to the backend. The previous model on the backend is an inaccurate estimate for crime location.
   User confirms or declines.
   If the user confirms, send the data through the event stream to get more accurate predictions.
   GCP: Use Google Functions for clients and Google PubSub (short for publish subscription) for streaming.
   The outline is similar to AWS: Create a data stream A to send data from the backend to the client. In case a PubSub "topic".
   Create a customer function
   Deploy customer functions in Google Cloud Functions
   Create a data flow B to ingest the customer output.
   Pulling data from stream B in the backend
   To deploy as a stream: The backend receives unconfirmed crime information from the user. Sends information to the backend.
   The previous model on the backend has an incorrect estimate for crime location.
   User confirms or declines.
   If the user confirms, send the data through the event stream to get more accurate predictions.
10. GCP: Use Google Functions for clients and Google PubSub (short for published subscription) for streaming.
    The outline is similar to AWS:
    Create a data stream A to send data from the backend to the client. In case a PubSub "topic".
    Create a customer function

Deploy customer functions in Google Cloud Functions
Create data flow B to contain the customer output
Pulling data from stream B in the backend

11. Maintenance
    Why monitor ML models: ML production models are production software and therefore face the same issues as other production SE/SD software. However, in addition to these general problems, some ML-specific problems may occur in ML production models that are not present in SE/SD. Therefore, SE/SD tools are not sufficient to monitor ML production models.
    Monitoring. Over time, ML models may deteriorate. This is due to one of two effects:
    Data drift: In which new input data is no longer represented by the model's training dataset.
    Concept flow: in which the concept changes, i.e., the relationship between input and output is changed (though not necessarily the data itself). As the name implies this inversion is due to changing the "assumptions" (i.e., hidden variables, underlying assumptions… etc.). Example: Criminal records are replaced by new ones. model can no longer accurately predict the outcome. In a more comprehensive setting, the monitor has 4 criteria:
    Segment wise performance: Performance in each segment of the input distribution
    Model bias/ unbiasedness outliers Explainability
12. Observations in different paradigms.
13. Batch: In batch models, implement batch monitoring. Add some calculation blocks after the pipeline step and do some checks to make sure the model behaves as expected. In other words: calculate performance metrics and health metrics, log metrics to SQL or NoSQL database.
14. Online models: In real-time served models, one wants to keep a close eye on how the model is performing. Add a service that pulls metrics and updates visuals in real time. Sometimes, even though the model is online to examine the model in the batch as well as the mode. Some problems with models may only manifest over long periods of time or on large datasets. To observe earlier prediction. Let's examine backend model. In this monitoring deployment, both online monitoring via Prometheus and Grafana as well as offline monitoring via Evidently AI is desired.
15. Implementation of Online Learning:
    For batch learning, three components are required: Forecasting services, Service clearly, MongoDB, Prometheus and Grafana.
    Clearly serve: The backend sends data to the Evidently Service to calculate metrics. The resulting metrics are then logged into PrometheusDB which will be accessed by Grafana to create dashboards.

For the Evidently Service container, use the files provided by the instructors. Some parquet files are required beforehand. It is packaged in the ready.py file. Evidently the Service is primarily composed of the Monitoring Service. Manually calling the various monitors provided by Evidently to evaluate the quality of the mod, then calling the Monitoring Service class's iterate method to calculate the different metrics and pushing the metrics to PrometheusDB (each metric in its own "Gauge" object). Grafana Dashboard. Obviously takes some parameters to initialize. Chief among them: use_reference and dataset_path: for reference datasets; Mandatory for certain metrics. Each dataset used in Eevidently also requires some parameters; ie a column_mapping that explicitly states which columns are numeric/categorical/…

All these parameters are taken explicitly by the config.yaml file. The service is called explicitly by the Flask API after the service is initialized. Each time a new value is output, the JSON is sent to iterate/[dataset name](eg: localhost:8085/iterate/crime) and run by the service. MongoDB, Prometheus and Grafana services: Use the configuration provided by the instructors for Prometheus and Grafana. As for MongoDB containers, use containers on DockerHub.

Dashboards created by Evidently are also configured via files provided by instructors.

Implementing Batch Learning: In batch learning, run the service periodically and generate an HTML report.

- The service reads data from MongoDB
- Updates the data with the target row
- Runs the model on the reference data
- Run metrics on data from MongoDB using context data [as context].
- Creates a dashboard
- Inserts the resulting metrics into MongoDB
- Saves the HTML report

3.1 Procedure for detecting number plate

The license plate can be extracted from the image using computer vision (Shah et al 2021) techniques followed by optical character recognition identifying the license plate from CCTV.

i. Get the image for input
ii. Use Gaussian blur to reduce noise if the image is blurry

iii. Then convert the image to grayscale
iv. Then it is necessary to find the vertical edges in the image
v. Use Otsu's thresholding of the vertical image to accept the plate to binarize the image.
vi. Need to find the license plate to find the contours in the image
vii. Find the rectangle with least area then check the ratio and area
viii. Get perfect contour after authentication in license plate
ix. After that the contour needs to be extracted from the original image
x. Use image segmentation to identify the characters on the license plate
xi. The value channel image of the plate then requires adaptive thresholding to binarize it and reveal the character.
xii. A bitwise operation is required to detect the added elements in the image so that the character candidates are extracted
xiii. Then extract the contours from the character candidate mask as well as those contour areas from the value thresholded image of the plate and get each character separately.

Proposed end-to-end method for license plate detection and recognition, which is based on a deep convolutional neural network (CNN) architecture (Keyvanpoura et al 2010). The authors provide a detailed explanation of the mathematical expressions used in the CNN, including convolutional layers, pooling layers, and fully connected layers. Then describe the training process of their model, which involves data augmentation techniques such as random rotation and translation, and the use of transfer learning from a pre-trained model. Loss function used in their model, which is a combination of classification loss and regression loss. The proposed method achieves high accuracy in license plate detection and recognition, while also being computationally efficient and lightweight.

3.2 Face recognition of live video streams

To convert this image into a grayscale image so that the background color cannot affect the image and feature of the person. Face detection in video is performed using a pre-trained model. “haarcascade_frontalface_default” is a model that detects the front face. It is a transfer learning technique that takes an overall trained model and retrains it from the current weights for new classes. A pre-trained model is trained on a large dataset (Shah et al 2021). This model effectively serves the general model.

1. Feature Extraction: It is the extraction of new features from a new dataset. It is trained from scratch using new classifiers. No need to train the entire model.
2. Fine-tuning: It has a higher-order feature. It is represented in the base model to make it more attractive for a specific task.

3.3 Face recognition

It is the easiest face recognition to recognize and handle faces. State-of-art built with deep learning using dlib with 99.38% model accuracy.

Features:

1. Find all the faces in that image
```
import face_recognition
image = face_recognition.load_image_file("file.jpg")
face_locations = face_recognition.face_locations(image);
```
2. It gives the position and outline of each person like eyes, nose, mouth, chin.
```
image = face_recognition.load_image_file("file.jpg")
face_landmarks_list = face_recognition.face_landmarks(image)
```
3. Recognize face in picture: Recognizes every photo that appears.
```
import face_recognition
know_image = face_recognition.load_image_file("biden.jpg")
unknown_image = face_recognition.load_image_file("unknown.jpg")
biden_encoding = face_recognition.face_encodings(known_image)[0]
unknown_encoding = face_recognition.face_encodings(unknown_image)[0]
result = face_recognition.compare_faces([biden_encoding], unknown_encoding)
```

Use this library with other Python libraries to perform real-time face recognition. Computer vision are used for Examines information about the environment from a camera, making the software important. It also executes facial recognition, license plate recognition, enhanced and blended realities, position detection, object identification, personal photograph collection, example recognition, geometric alignment, massive databases, location identification, category identification with division, smart image editing, context and scene comprehension of comprehensive image collection, education (Tripathi et al 2021), image exploration, identification databases, and assessment suites.

1. core analytics: predict future outcomes in cases including everything from behavioral impulses to robbing a store soon (Shah et al 2021).
2. Artificial neural networks: They aid in discovering patterns in data by functioning like a human brain, imitating biological neurons. They can comprehend and even forecast a crime scene (Shah et al 2021; Tripathi et al 2021).
3. Rule-based systems: It increases the safety of the system from viruses. It works as an antivirus (Shah et al 2021).
4. Cryptographic algorithm: It is used into two parts
   a) confidential criminal data it's privately encode.
   b) Maintain the newly found possible criminal information in an encrypted state.
5. Iterative processors: apply the function of our machine repeatedly to ensure they keep working continuously and never stop monitoring the machine (Shah et al 2021).
6. Bayesian Belief Networks: These are probabilistic models that can be represented by acyclic graphs and used for prediction, anomaly detection, diagnostic, and automated insight (Shah et al 2021).
7. Data acquisition: learn from previous crimes and predict future crimes (Shah et al 2021).
8. Document processor: it is used after data collection, primarily organizing, analyzing, and learning from the data.
9. Computational linguistics: it gives the ability to a computer to understand human spoken language (Shah et al 2021).
10. Computational linguistics system: it is also utilized by computers to improve understanding of human language (Shah et al 2021).
11. Voice biometrics: it uses unique vocal characteristics to differentiate between individuals, making their voice easily identifiable (Shah et al 2021).
12. Gait analysis: used to study human emotion (Shah et al 2021).
13. Biometric identification: identify faces, thumb print (Shah et al 2021).
14. Pattern mining: it helps to observe patterns among the routine activities (Shah et al 2021).
15. Intel comprehension: it is utilized to grasp the meaning of collected data (Shah et al 2021).
16. Threat detection: Identify any potential danger while processing intelligence by ensuring that a specific set of predetermined checkboxes have been selected (Shah et al 2021).

4. Discussion

Integrating MLOps and EEG Techniques for Enhanced Crime Detection and Prevention is a novel approach that aims to enhance the performance of the criminal justice system by using advanced technologies such as Machine Learning Operations (MLOps) and Electroencephalography (EEG) techniques. This paper provides a detailed overview of the proposed approach and its potential applications Within the domain of detecting unlawful activity.

The study emphasizes the importance of integrating MLOps and EEG techniques to detect criminal behavior with high accuracy. The authors discuss the different components of the proposed system, including data collection, pre-processing, feature extraction, and classification. The work also presents the results of the experiments that were conducted to verify the efficacy of the suggested system in identifying unlawful conduct. Detecting criminal behavior.

The discussion section of the paper highlights the potential impact of this approach on the criminal justice system. The authors argue that the proposed system can significantly improve the accuracy of crime detection and prevention, thereby reducing crime rates and enhancing public safety. The use of MLOps and EEG techniques can also reduce the workload of law enforcement agencies and improve the efficiency of the criminal justice system.

Moreover, the paper highlights some limitations of the proposed approach and suggests future research directions. One limitation is the need for a large amount of high-quality EEG data to train the machine learning models effectively. Another limitation is the cost and complexity associated with EEG equipment and data acquisition. Future research can focus on developing cost-effective and user-friendly EEG systems that can be easily integrated into the criminal justice system.

5. Conclusions

The proposed system explored the use of MLops techniques for crime detection, specifically focusing on vehicle plate recognition, person detection, and object behavior detection. Through the use of various algorithms and deep learning models, the system was able to successfully recognize and classify various objects and people with high accuracy. Additionally, the system was able to extract useful information from surveillance footage, such as the make and model of vehicles and license plate numbers, which can aid law enforcement in identifying and apprehending suspects.

The literature review shows the potential effectiveness of using MLops techniques in crime detection and prevention. This technology has the potential to greatly enhance public safety and help law enforcement agencies to work more efficiently and effectively. However, further research and development is needed in order to optimize these techniques for real-world applications, and to address potential issues such as privacy concerns and false positives. Overall, this research will contribute to the ongoing efforts to develop innovative and effective technologies for crime prevention, and it will inspire further research and development in this field.

EEG has shown promising potential as a non-invasive and objective tool for detecting crime. Through the use of EEG, it is possible to capture and analyze brain activity patterns in response to various stimuli (Mane and Shinde 2022), including those related to criminal activity. By identifying distinctive patterns of neural activity associated with criminal intent or actions, EEG can assist law enforcement in identifying suspects and preventing crimes. However, it is important to note that EEG is not a standalone tool and should be used in conjunction with other evidence-gathering techniques. Furthermore, there are still many challenges that need to be addressed, including the development of more accurate and reliable algorithms for interpreting EEG data. Overall, EEG has the potential to become an important tool in the fight against crime, and continued research and development in this area is needed to unlock its full potential.

Ethical considerations

Not applicable.

Conflict of Interest

The authors declare no conflicts of interest.

Funding

This research did not receive any financial support.

References

Fredrick David HB, Suruliandi A (2017) Survey on crime analysis and prediction using data mining techniques. ICTACT J Soft Comput 7:1459-1466. DOI: 10.21917/ijsc.2017.0202

Gendre V (2022) A survey on crime detection and Prediction techniques. Int J Res Appl Sci Eng Technol 10:119–122. DOI: 10.22214/ijraset.2022.39785

Guo Z, Suvorov DV (2022) Comparison of Effectiveness of Several Classifiers in EEG Emotion Recognition. GitHub. Available in: https://denisandgzh.github.io/EEG-Emotion-Recognition/. Accessed on: May 15, 2023.

Hamdy E, Adl A, Hassanien AE, Hegazy O, Kim TH (2015) Criminal act detection and identification model. Seventh International Conference on Advanced Communication and Networking (ACN). DOI: 10.1109/ACN.2015.30

Kadam SU, Shinde SB, Gurav YB, Dambhare SB, Shewale CR (2022) A novel prediction model for compiler optimization with hybrid meta-heuristic optimization algorithm. Int J Adv Comput Sci Appl 13. DOI:10.14569/ijacsa.2022.0131068

Keyvanpour MR, Javideh M, Ebrahimi MR (2011) Detecting and investigating crime by means of data mining: a general crime matching framework. Procedia Comput Sci 3:872–880. DOI: 10.1016/j.procs.2010.12.143.

Khoje S, Shinde S (2023) Evaluation of ripplet transform as a texture characterization for Iris recognition. J Inst Eng (India) Ser B 104:369–380. DOI:10.1007/s40031-023-00863-6

Li T (2016) Criminal behavior analysis method based on data mining technology. International Conference on Smart City and Systems Engineering (ICSCSE).

Mahmud N, Zinnah KI, Rahman YA, Ahmed N (2016) Crimecast: A crime prediction and strategy direction service. 19th International Conference on Computer and Information Technology (ICCIT).

Mane SAM, Shinde AA (2022) Novel imaging approach for mental stress detection using EEG signals. Advances in Intelligent Systems and Computing. Singapore: Springer Singapore 2536.

Nath SV (2006) Crime pattern detection using data mining. IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops. DOI: 10.1109/WI-IATW.2006.55

Prabakaran S, Mitra S (2018) Survey of analysis of crime detection techniques using data mining and machine learning. J Phys Conf Ser 1000:012046. DOI: 10.1088/1742-6596/1000/1/012046

Reier Forradellas RF, Náñez Alonso SL, Jorge-Vazquez J, Rodriguez ML (2020) Applied Machine Learning in social sciences: Neural networks and crime prediction. Soc Sci (Basel) 10:4. DOI: 10.3390/socsci10010004

Rony MS, Bakchy SC, Rahman H (2020) Crime Detection using Data Mining Techniques. Comput Sci Eng Int J 10:1–5. DOI: 10.5121/cseij.2020.10501

Safat W, Asghar S, Gillani SA (2021) Empirical analysis for crime prediction and forecasting using machine learning and deep learning techniques. IEEE Access 9:70080–70094. DOI: 10.1109/access.2021.3078117

Sardeshmukh M, Chakkaravarthy M, Shinde S, Chakkaravarthy D (2023) Crop image classification using convolutional neural network. Multidisciplinary Science Journal (Accepted Articles) 2023039. DOI: 10.31893/multiscience.2023039

Shah N, Bhagat N, Shah M (2021) Crime forecasting: a machine learning and computer vision approach to crime prediction and prevention. Vis Comput Ind Biomed Art 4:9. DOI:10.1186/s42492-021-00075-z

Shewale C, Shinde SB, Gurav YB, Partil RJ, Kadam SU (2023) Compiler optimization prediction with new self-improved optimization model. Int J Adv Comput Sci Appl. 14(2). DOI:10.14569/ijacsa.2023.0140267.

Shinde S, Wadhwa L, Bhalke D (2021) Feedforward back propagation neural network (FFBPNN) based approach for the identification of handwritten math equations. Advances in Intelligent Systems and Computing 757–775. DOI: 10.1007/978-3-030-51859-2_69.

Tripathi A, Yadav A, Poojary T, Jeswani J (2021) Criminals as well as crime detection using Machine Learning & OpenCV. International Research Journal of Modernization in Engineering Technology and Science 3:2135–2141.

Xie H, Liu L, Yue H (2022) Modeling the effect of streetscape environment on crime using street view images and interpretable machine-learning technique.

Int J Environ Res Public Health 19:13833. DOI: 10.3390/ijerph192113833.

Yang Q, Wang J, Xu X, Qin T (2011) Quantitative analysis of serial criminal. The Fourth International Workshop on Advanced Computational Intelligence. IEEE.