Paper 1(Algorithms for EEG Datasets):

- Systematic literature review focused on the use of machine learning in EEG
- Aims to empower CS students interested in BCI with knowledge and insights about the field
- CNN, RNN, Transformer, SVM, KNN, RF are the recommended machine learning algorithms for CS students to begin their exploration of EEG analysis
- Motor Imagery, Seizure Detection, and Emotion Detection are the top three EEG tasks in BCI research
    - Significant detail: for Motor Imagery, the most commonly used algorithms are CNN, RNN, Transformer, SVM, RF.
    - Significant detail: for Emotion Detection, the most commonly used algorithms are CNN, RNN, SVM, KNN.
    - Significant detail: for Seizure Detection, the most commonly used algorithms are CNN, RNN, Transformer, KNN.
- DEAP, EEGEyeNet, CHB-MIT, SEED, and BCI Competition IV are the key EEG datasets for CS students starting in EEG analysis.
- DEAP for human affective state analysis, BCI Competition IV for Motor Imagery, and CHM-MIT for seizure analysis.
- Step-by-step guide for getting started with BCI research

Paper 2 (Predicting Pushback of Flights):
Significant Points:

- State the point of the research paper which is predicting the time of push of flights, including defining ‘pushing flights’
    - The need to optimize airport capacity and runway usage
- Explain why this is important such as for air traffic management
- Describe the methodology of the paper, this includes the use of a gradient boosting decision tree model which is trained on set which includes weather, airport activity, airline, and aircraft characteristics
    - Show feature engineering and model architecture to highlight the relationship between predictors and pushback durations
    - Machine learning pipeline, including data preprocessing, feature extraction, model training, and validation
    - Feature selection and encoding techniques in improving model performance and interpretability
- Provide details for the dataset that is used for training including the number of flights, airports, and features
- Talk about the training process of the predictive model, whether global or local models were used, and how they were analyzed like using mean absolute error (MAE)
- Show the validity of the approach by comparing the performance of their predictive model against a baseline model, show how much better the model improves prediction accuracy
    - Talk about results

    - Present the predictive accuracy
    - Analysis of prediction errors across different airports
        - Present potential influencing factors
- Address the efficiency of the model's performance, including training data, memory usage, scalability, etc.
    - Advantages of local training model
- Discuss limitations of the paper (not having access to some of the relevant data)

Good to have:
- Provide specific statistics including statistical analysis
- Propose future research or improvements
- Include a reference to another research paper
- Explanation of keywords

Paper 3 (Analyzing origin of biases in facial recognition analysis)
- It is critical that facial expression recognition models perform consistently and equitably across diverse populations due to their widespread usage
    - At present, this is not reality; "existing public FER models demonstrate biases in the faces of diverse populations"
- This paper analyzes these biases
    - Generated an artificial facial expression dataset (940 faces) which researchers could manipulate to "isolate the impact of distinct manipulations on our model" and "better understand" the biases
- The goal is to
    - Investigate skin color biases, their complexity and non-linearity
    - Study the impact of skin color distribution in the training set to highlight the necessity of dataset diversity and distribution
- The results reflected the existence of a skin color bias within FER models, displaying consistent disparities between skin tones in this testing environment
- Limitations
    - Small dataset lacking in diversity (ethnicity, morphology, pose, lighting, etc.)
        - All faces were European and had the same morphology
    - Only evaluated a single model and a few variables–the results only apply to that model

Paper 4(The Friendship Paradox: An Analysis on Signed Social Networks with Positive and Negative Links):
- Significant Points:
    - Network Topology: The relationships between the individuals involved in the system (Social Platform)
    - Friendship Paradox: On average, an individual's friends have more friends than that individual. A node is more likely to be a neighbor of a node with many neighbors (i.e., high degree), compared to being linked to a node with only a few edges (i.e., low degree).
    - First and Second Order Measurements:

- First Order: $\psi_{i,1}^{\alpha\beta} = \frac{1}{N_\alpha(v_i)} \sum_{v_j \in N_\alpha(v_i)} 1(|N_\beta(v_j)| > |N_\beta(v_i)|)$
- Second Order:
$$\psi_2^{\delta} = \frac{1}{N} \sum_{i=1}^{N} 1((\frac{1}{|N_+(v_i)|} \sum_{v_j \in N_+(v_i)} |N_\delta(v_j)|) > (\frac{1}{|N_-(v_i)|} \sum_{v_k \in N_-(v_i)} |N_\delta(v_k)|))$$
  - Needs to include a version of these functions
- Datasets: Bitcoin Alpha, Wiki Elections, Honduras Village, Slashdot, Epinions
- The set of users that view a node positively and the set of users the node views positively are likely to have on average even more incoming and outgoing positive reviews.
- Most nodes have both less friends and less enemies than its friends and enemies do.
- Users who view a node negatively, are on average, likely to leave more negative reviews than the node does
- Second Order Results:
  - .Positive Trend: For most nodes, their friends have more friends than their enemies do.
  - Negative Trend: For most nodes, their friends also have more enemies than their enemies do.
  - Negative Trend is slightly weaker than the positive trend.

Paper 5(Autograding for Block-based Code):
- Designs a Computational Thinking course designed around Snap!, an introductory block-based programming language, to teach computer science fundamentals.
- Uses data from this course, which focuses on abstraction, algorithms, data representation, and documentation, to train and evaluate an automated evaluator (autograder) for Snap projects.
- Autograder measures repetition and encapsulation, failed to measure features such as data representation and documentation well,
  - Significant detail: Evaluates repetition with 3 approaches: by sliding a filter over the XML code, with K-means clustering, and with a Gaussian Mixture Model
- Compared to human graders, the autograder lacked the ability to highlight student improvement, create personalized alternative coding solutions, appreciate the artistic output of multimedia programs.
- Contribution: reports which features can be extracted from XML representation of a Snap! program, and how to analyze them. This can be generalized to other block-based programming languages.

Old paper 3(Addressing the issue of representation learning in time series data mining)

- **Goal of the paper**: Addressing the issue of representation learning in time series data mining (a task which attempts to transform time series into low-dimensional representations that can represent their semantic similarity and priming the data for future tasks)
  - The current approach to this issue is using a self-supervised learning framework (SSL) called TS2Vec
    - This approach has a high computational cost
- **Contribution of the paper**: This paper proposes a single resolution-based loss function to train the model called *adaptive resolution loss* which achieves similar accuracy to TS2Vec while reducing computation time; it then evaluates this method on time series classification tasks
  - The proposed model is based on TS2Vec ("...comprises of three main components: an Input Projection Layer $h(.)$, Random Cropping Augmentation $acroping(.)$, and Time Stamp Masking Module $amask(.)$."
  - It only computes a single resolution per epoch, therefore significantly increasing the algorithm's efficiency
    - "by selecting the most important resolution to optimize, we can optimize the loss without using the full computation power in all resolutions"
  - Results were determined through an experiment which used 6 datasets from UEA/UCR Multivariate Time Series Classification Archive
    - Compared classification accuracy performance with TS2Vec
    - Repeated 5 times and reported average
- **Results of the paper**: Accuracy is similar or better, speed is consistently better
- **Summary must have**

    - This paper is a part of the broader topic of representation learning in time series data mining
        - Need an explanation of representation learning for accessibility to those who are unfamiliar with this topic
    - This paper proposes a method to improve the computational efficiency of TS2Vec, a state-of-the-art model for time series representation learning
    - This method involves adaptive resolution setting in the model's loss function
        - Explain why this is more efficient
    - It performs at the same or better level of accuracy, and consistently performs faster
- **Summary could have**
    - Definitions of more common jargon (e.g. what is TS2Vec, what is adaptive resolution, what is a loss function, what are time series…)
    - More detailed description about how the proposed method is structured and functions

Other Old Paper 3: (Improving Graph Cohesiveness by Merging Nodes)

- Graph robustness and interconnectedness can be improved by adding edges or merging nodes. The paper is possibly the first to focus on the second approach in this topic.
- k-trusses are used to measure the connectedness of the graph, and are fast to compute and effective at real-word applications. The goal of the paper is to maximize the k-truss for a given graph by merging nodes
- the k-truss of G is the maximal subgraph of G where each edge is in at least $k - 2$ triangles
- The paper proposes TIMBER(Truss-sIze Maximization By ERgers), an NP-hard optimization problem on graphs
- The paper proposes BATMAN(Best-merger sAarcher for Truss MAximizatioN) as an algorithm to solve TIMBER, which removes outside nodes, then considers the possible inside-outside and inside-inside mergers and selects the best one
- BATMAN outperforms the baseline algorithms by 1.38x to 10.08x, in terms of the average increase in truss size, with the improvement depending on the dataset.