Probabilistic Graphical Models for Psychological State Identification: An Exploratory Study with Simulated Data

Darshan Singh
CSAM, PGM, Honours in Computer Science]
University of the Witwatersrand

***Abstract*—The automated identification of psychological states from physiological and behavioral data holds significant potential for applications in mental well-being and human-computer interaction. This report explores the application of Probabilistic Graphical Models (PGMs), specifically Bayesian Networks (BNs), for this task. The study initially utilizes the public Kaggle "Psychological State Identification Dataset," a simulated multimodal dataset, to classify individuals into four states: Stressed, Relaxed, Focused, or Anxious. Initial experiments revealed challenges in learning meaningful predictive structures, consistent with community findings on this dataset. Consequently, a custom simulation was developed to generate data with more theoretically-grounded dependencies, incorporating noise and missing values. Experiments were conducted using this custom dataset, including a simplification of the target variable into two broader categories ('Agitated' and 'Calm/Engaged'). Bayesian Network models were constructed using score-based (HillClimb with BIC) and constraint-based (PC algorithm) structure learning, with parameters learned via Maximum Likelihood Estimation (MLE), implemented using the pgmpy library. Results show that while the original 4-class problem on simulated data remains challenging, simplifying to a 2-class problem on custom-simulated data with basic prior knowledge significantly improved predictive accuracy, achieving up to 80%. This highlights the potential of BNs for psychological state modeling, the importance of data quality and feature relevance, and the utility of custom simulations for model exploration.**

***Index Terms*—Probabilistic Graphical Models, Bayesian Networks, Psychological State Identification, Simulated Data, Machine Learning, Structure Learning.**

I. Introduction

The increasing prevalence of mental health concerns and the desire for more adaptive human-computer interfaces underscore the need for effective methods to monitor and understand human psychological states [1]. Advances in ubiquitous sensing technologies, particularly wearable biosensors, offer promising avenues for collecting rich, multimodal data streams (e.g., physiological signals, behavioral patterns, environmental context) that potentially correlate with underlying psychological states such as stress, relaxation, focus, and anxiety [2].

Artificial Intelligence (AI) techniques are increasingly applied to analyze such data. However, many sophisticated machine learning models, like deep neural networks, often function as "black boxes," limiting their interpretability. In sensitive domains like psychological assessment, interpretability is crucial for understanding model behavior, building trust, and enabling expert validation [3]. Probabilistic Graphical Models (PGMs), particularly Bayesian Networks (BNs), offer a compelling alternative. BNs excel at modeling complex systems with numerous interacting variables by explicitly representing probabilistic dependencies and independencies through a Directed Acyclic Graph (DAG) structure [4]. This graphical representation facilitates interpretability and allows for the incorporation of domain knowledge.

This project explores the application of Bayesian Networks for identifying psychological states. The study initially investigates the publicly available "Psychological State Identification Dataset" from Kaggle [5]. Given the challenges encountered with this dataset, a custom data simulation was developed to create a dataset with more controlled and theoretically-grounded relationships. The objectives are: (1) to implement and evaluate BN models for classifying psychological states using these simulated datasets, (2.1) to explore different structure learning approaches, (2.2) to assess the effect of simplifying the classification task by grouping similar psychological states, and (3) to discuss the findings, limitations, and potential of BNs in this domain.

This paper is structured as follows: Section II describes the datasets used and preprocessing steps, including ethical considerations. Section III details the Bayesian Network methodology. Section IV presents the experimental setup and results. Section V discusses the findings and limitations, and Section VI concludes the paper.

II. Datasets and Preprocessing

A. Dataset Descriptions

Two primary datasets were utilized in this study:

1) Kaggle Psychological State Identification Dataset: This dataset, publicly available on Kaggle [5], simulates data collected from biosensors and other sources to evaluate the psychological states of individuals (Stressed, Relaxed, Focused, Anxious). It contains 1000 instances and over 20 features, including physiological data (HRV, GSR, EEG bands, Blood Pressure, etc.), environmental factors, behavioral features, and psychological context variables. A critical aspect of this dataset is its simulated nature, which significantly impacts the interpretation of results. Initial explorations by other community members under this Kaggle dataset indicated difficulties in achieving high predictive accuracy, often with models defaulting to predicting the most frequent class.

2) Custom Simulated Dataset: To address the potential limitations of the Kaggle dataset and to explore model behavior with more controlled dependencies, a custom dataset was simulated using Python. This simulation was designed based on theoretical relationships between cognitive load, physiological responses, and psychological states, blending domain expert knowledge that was consulted on as well. Key aspects include:

- **Central Latent Variable:** 'Cognitive Load' (Low, Medium, High) influenced by 'Task Type'.
- **Physiological and Behavioral Dependencies:** The features *Heart Rate*, *HRV (ms)*, *GSR (μS)*, *EEG_Alpha*, *EEG_Beta*, *EEG_Delta*, *Respiration Rate (BPM)*, *Skin Temperature ($^\circ$C)*, *BP_Systolic*, *BP_Diastolic*, and *Focus Duration (s)* were simulated as conditionally dependent on the latent variable *Cognitive Load*.
- **Target Variable Derivation:** 'Psychological State' was derived probabilistically from 'Cognitive Load'.
- **Noise and Missing Values:** Gaussian noise was added to continuous variables, and a fraction of data points were randomly set to missing to mimic real-world imperfections.

This custom dataset comprised 1000 samples and included the features mentioned above.

B. Data Preprocessing

The following preprocessing steps were applied consistently to the relevant dataset before model training:

1) **Feature Extraction (Primarily for Kaggle Dataset):** For the original Kaggle dataset, string-based features like 'EEG Power Bands' and 'Blood Pressure (mmHg)' were parsed into their respective numerical components (e.g., `EEG_Delta`, `BP_Systolic`). This step was largely unnecessary for the custom-simulated data where features were already in a more structured format.
2) **Handling Missing Values and Data Types:**
   - Columns intended to be numerical were explicitly converted using `pd.to_numeric(errors='coerce')`.
   - Any resulting infinities (`np.inf`, `-np.inf`) were replaced with `NaN`.
   - Remaining `NaN` values in these numerical columns were imputed using the median value of that column, calculated from the training data and applied to both training and test sets to prevent data leakage.
   - `NaN` values in categorical columns were filled with a placeholder string (e.g., `Missing_Cat_Value`).
3) **Class Grouping (Experiment 2):** For the second main experiment using the custom dataset, the original four psychological states were mapped into two broader categories: `Agitated` (combining `Stressed` and `Anxious`) and `Calm_Engaged` (combining `Relaxed` and `Focused`). This new binary target variable was named `State_Group`.
4) **Numerical Feature Discretization:** All continuous numerical features were discretized into a fixed number of bins (`N_BINS`, set to 5 for these experiments) using `KBinsDiscretizer` from `scikit-learn` with an 'ordinal' encoding and 'uniform' binning strategy. The discretizer was fit exclusively on the training data.
5) **Categorical Feature Encoding:** All features (original categorical and newly discretized numerical) and the target variable were converted into integer representations using `LabelEncoder` from `scikit-learn`. Each encoder was fit on the combined data from the training and test sets for its respective column/target.
6) **Data Splitting:** The fully preprocessed dataset was split into training (80%) and testing (20%) sets using `train_test_split`, with splits based on the target variable.

C. Ethical Considerations

While this project primarily used simulated data, thereby circumventing direct ethical concerns of human subject research like informed consent for data collection, it is imperative to discuss the ethical landscape for real-world deployment. AI in mental health demands rigorous scrutiny [6]. Key considerations include:

- **Privacy and Confidentiality:** Real biosensor and psychological data are extremely sensitive. Robust encryption, anonymization, secure storage, and adherence to regulations (e.g., GDPR, HIPAA) are paramount.
- **Bias and Fairness:** AI models can perpetuate biases present in training data, leading to discriminatory outcomes. Careful bias audits and mitigation are needed. The simulated data likely lacks the complexity of real-world biases.
- **Informed Consent and Autonomy:** Users must understand data collection, usage, model limitations, and risks, with the right to opt-out.
- **Transparency and Explainability:** BNs offer an advantage due to their interpretable graphical structure, which is vital for trust and accountability.
- **Safety and Efficacy:** Rigorous validation is essential to prevent misdiagnosis or harmful recommendations. Human oversight remains crucial.

The findings from this study on simulated data cannot be directly extrapolated to real-world scenarios without addressing these ethical challenges.

III. Bayesian Network Methodology

A. Rationale for Bayesian Networks

Bayesian Networks were selected for this project due to their ability to:

- Model complex probabilistic dependencies and independencies among variables graphically.
- Offer interpretability through their DAG structure, allowing for an understanding of learned relationships.
- Handle uncertainty inherent in data and model parameters.

- Integrate heterogeneous data types (after appropriate pre-processing like discretization).

B. Structure Learning

The Directed Acyclic Graph (DAG) structure of the Bayesian Network, which encodes conditional independencies among variables, was learned from the processed training data. The primary method employed was a score-based approach:

- **Score-based Algorithm:** The *Hill Climbing* search algorithm was utilized in conjunction with the *Bayesian Information Criterion (BIC)* score. BIC is favored for its ability to balance model fit against model complexity, thereby penalizing overly complex network structures that might overfit the training data and generalize poorly.

While constraint-based methods like the PC algorithm were considered, the HillClimb with BIC approach was the main focus for the reported experiments on the custom-simulated dataset. The incorporation of domain knowledge was primarily achieved through the informed design of the custom data simulation, aiming to embed plausible relationships between variables for the learning algorithms to discover. The `pgmpy` library [7] was used for all Bayesian Network implementations.

C. Parameter Learning

Once the DAG structure G was determined, the parameters Θ, defining the Conditional Probability Distributions (CPDs) $P(X_i|\text{Parents}(X_i))$ for each variable X_i, were estimated from the training data. *Maximum Likelihood Estimation (MLE)* was used. For discrete variables, MLE typically corresponds to calculating the relative frequencies of variable states conditioned on the states of their parents.

D. Inference

With the learned structure G and parameters Θ, the BN defines a full joint probability distribution. Inference was performed to predict the probability distribution of the target variable (`Psychological State` or `State_Group`) given evidence e (the observed values of other features). The *Variable Elimination* algorithm, an exact inference method implemented in `pgmpy`, was used to compute the posterior probability $P(\text{Target}|e)$. The class with the highest posterior probability was chosen as the prediction.

IV. Experimental Evaluation and Results

A. Experimental Setup

All experiments were conducted in a Python 3 environment. Core libraries included `pgmpy` (for Bayesian network modeling), `pandas` (for data manipulation), and `scikit-learn` (for preprocessing and evaluation metrics). Visualizations were produced using `matplotlib` and `seaborn`. The custom simulated dataset was primarily used for the detailed experiments, following initial challenges with the Kaggle dataset. Data was split into 80% training and 20% testing sets, stratified by the target variable. Numerical features were discretized into five bins using `KBinsDiscretizer` with a `uniform` strategy, unless otherwise specified.

B. Baseline: Kaggle Dataset (4-Class Target)

Initial experiments on the original Kaggle dataset with a 4-class target proved challenging. Score-based learning (Hill-Climb with BIC) often resulted in sparse or empty graphs, leading to poor predictive performance, typically defaulting to the majority class. This aligned with observations from the Kaggle community, suggesting weak or obscured predictive signals in that particular simulation. The evaluation on the test set yielded an overall accuracy of 0.2650. The classification report (Table I) and confusion matrix (Fig. **??**) are shown below.

TABLE I
Classification Performance (4-Class, Kaggle Data)

Class	Precision	Recall	F1-score	Support
Anxious	0.00	0.00	0.00	48
Focused	0.00	0.00	0.00	50
Relaxed	0.00	0.00	0.00	49
Stressed	0.27	1.00	0.42	53
Accuracy			0.27	200
Macro Avg	0.07	0.25	0.10	200
Weighted Avg	0.07	0.27	0.11	200

C. Experiment 1: Custom Simulated Data (4-Class Target)

Using the custom-simulated dataset with the original four psychological states as the target, and employing HillClimb-Search with BIC, the model learned a structure with 14 edges. The evaluation on the test set yielded an overall accuracy of 0.5300. The classification report (Table II) and confusion matrix (Fig. 1) are shown below.

TABLE II
Classification Performance (4-Class, Custom Data)

Class	Precision	Recall	F1-score	Support
Anxious	0.00	0.00	0.00	44
Focused	0.54	0.58	0.56	65
Relaxed	0.62	1.00	0.77	30
Stressed	0.47	0.62	0.54	61
Accuracy			0.53	200
Macro Avg	0.41	0.55	0.47	200
Weighted Avg	0.41	0.53	0.46	200

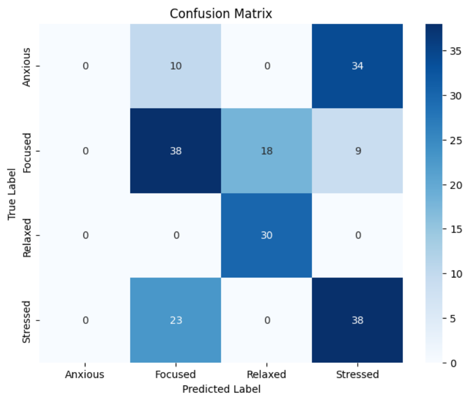

Fig. 1. Confusion Matrix (4-Class, Custom Data). (Actual values: Anxious[0,10,0,34], Focused[0,38,18,9], Relaxed[0,0,30,0], Stressed[0,23,0,38])

The model showed some ability to predict ‘Focused”, ‘Relaxed”, and ‘Stressed”, but completely failed on ‘Anxious”. The learned network (Fig. 2, showed ‘Cognitive Load” as a central hub.

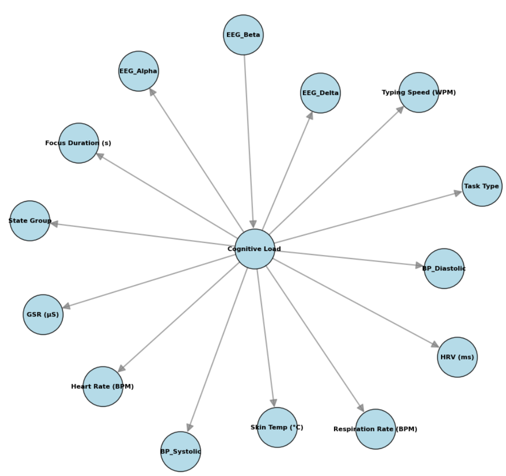

Fig. 2. Learned Network Structure (4-Class, Custom Data, HillClimb+BIC).

D. Experiment 2: Custom Simulated Data (2-Class Grouped Target)

The target variable was grouped into ‘Agitated’ and ‘Calm_Engaged’. Structure learning was again performed using HillClimbSearch with BIC in a data-driven manner.

- **Learned Edges**: 14. Thus producing the same network structure as in Experiment 1, with ‘Cognitive Load‘ remaining a central hub.
- **Performance:**
  - Overall Accuracy: 0.8000
  - Classification Report: (See Table III)
  - Confusion Matrix: (See Fig. 3)

TABLE III
CLASSIFICATION PERFORMANCE (2-CLASS GROUPED, CUSTOM DATA)

Class	Precision	Recall	F1-score	Support
Agitated	0.88	0.71	0.79	105
Calm_Engaged	0.74	0.89	0.81	95
Accuracy			0.80	200
Macro Avg	0.81	0.80	0.80	200
Weighted Avg	0.81	0.80	0.80	200

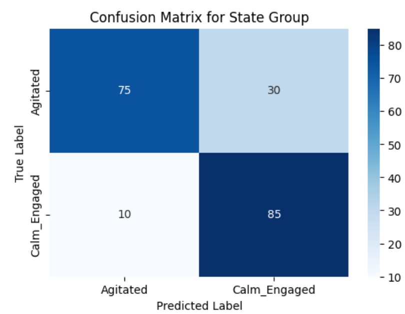

Fig. 3. Confusion Matrix (2-Class Grouped, Custom Data). (Actual values: Agitated[75,30], Calm_Engaged[10,85])

V. DISCUSSION

A. Interpretation of Results

The experiments demonstrate a clear progression. Initial attempts with the public Kaggle dataset (4-class) were largely unsuccessful in learning a predictive BN structure, which correlates with other Kaggle user’s that utilised much more complex training pipelines (XGBoost and other gradient boosting algorithms known to be especially efficient with noisy and missing data.)

The custom-simulated dataset, even with the 4-class target, allowed the BN to learn a more complex structure with ‘Cognitive Load’ as a central variable influencing various physiological outputs. While accuracy improved to 0.53, the model struggled significantly with the ’Anxious’ class, often misclassifying it as ‘Stressed’ or ‘Focused’. This suggests that the simulated distinctions for ‘Anxious’ might have been insufficient or too overlapping with other states after discretization.

The most significant improvement came from simplifying the problem by grouping the target into two broader categories (‘Agitated’ and ‘Calm_Engaged’). This resulted in an accuracy of 0.80, with good precision and recall for both grouped classes. This highlights that for complex, nuanced states, a simpler, more robust classification task might be more achievable, especially with data that has inherent noise or subtle differentiating features.

B. PGM Concepts Demonstrated

This project effectively demonstrated several core PGM concepts:

- **Representation:** BNs were used to model the joint probability distribution over features.

- **Structure Learning:** Score-based methods (`HillClimbSearch` with BIC) were primarily used to learn dependencies from data. The influence of domain knowledge was primarily channeled into the design of the custom simulation to create a dataset with plausible underlying relationships.
- **Parameter Learning:** MLE was used to quantify relationships.
- **Inference:** Variable Elimination enabled probabilistic predictions.

C. Limitations

- **Simulated Data:** The primary limitation of this study is the use of simulated data. While the custom simulation was designed with theoretical grounding, it cannot fully replicate the complexity, noise, and individual variability found in real-world human data. Therefore, the generalizability to real subjects remains uncertain.
- **Discretization Strategy:** The use of `N_BINS=5` and the `uniform` strategy for discretization represents a simplification. Alternative binning methods or different numbers of bins could influence the results and model behavior.
- **Structure Learning Algorithms:** The `HillClimbSearch` algorithm is susceptible to converging on local optima. More exhaustive search strategies or alternative algorithms may yield different and potentially improved network structures.
- **Model Complexity vs. Data Size:** With approximately 20 features and 800 training samples, learning highly complex dependencies remains challenging. Larger datasets would likely improve robustness and support more expressive models.

D. Future Research

- The most critical step is validation on real-world biosensor data from human participants.
- Exploring Dynamic Bayesian Networks (DBNs) could capture temporal dependencies in psychological states.
- Investigating more sophisticated feature engineering from raw sensor signals.
- Comparing BN performance against other interpretable and non-interpretable machine learning models on the same custom dataset.
- Further refinement of the custom simulation, particularly for differentiating states like ‘Anxious’.
- Systematic sensitivity analysis of model parameters and structure to variations in data.
- Utilising domain-expert knowledge not only for custom-simulation but also for more unknown conditional dependencies between features. Effectively guiding the model to improved prediction accuracies on more complex/noisy datasets, and incorporating effective predictions on rare-event prediction.

VI. Conclusion

This project successfully explored the application of Bayesian Networks for identifying psychological states using simulated data. While the initial Kaggle dataset proved challenging for learning predictive structures, the development and use of a custom-simulated dataset with theoretically-grounded dependencies allowed for more insightful model building and evaluation.

The experiments demonstrated that simplifying the classification task by grouping similar psychological states (from four classes to two) significantly improved predictive accuracy, achieving 80% on the custom binary task(this was also assisted due to the incorporation of domain knowledge to create the simulation). This highlights the BN’s ability to model and predict, given a sufficiently clear problem definition and data with learnable patterns. The interpretability of the learned BN structures, particularly the central role of ‘Cognitive Load‘, provided valuable insights into the simulated system.

The study underscores the importance of data quality and feature relevance in PGM modeling. While BNs offer a powerful and interpretable framework, their performance is intrinsically linked to the underlying data’s ability to reflect true dependencies. The custom simulation proved a useful tool for exploring these aspects in a controlled manner. Future work should focus on validating these approaches with real-world data and further exploring the nuances of PGM application in the complex domain of psychological state assessment, always with careful consideration of ethical implications.

References

[1] World Health Organization, ”Mental health action plan 2013-2020,” 2013.

[2] R. W. Picard, ”Affective computing,” MIT Press, 1997.

[3] C. Rudin, ”Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead,” Nature Machine Intelligence, vol. 1, no. 5, pp. 206-215, 2019.

[4] D. Koller and N. Friedman, ”Probabilistic graphical models: principles and techniques,” MIT Press, 2009.

[5] Ziya, ”Psychological State Identification Dataset,” Kaggle. [Online]. Available: https://www.kaggle.com/datasets/ziya07/psychological-state-identification-dataset. [Accessed: April 29, 2025].

[6] Joint Task Force for the Development of Telepsychology Guidelines for Psychologists (2013). Guidelines for the practice of telepsychology. The American psychologist, 68(9), 791–800. https://doi.org/10.1037/a0035001

[7] Ankur Ankan, & Johannes Textor (2024). pgmpy: A Python Toolkit for Bayesian Networks. Journal of Machine Learning Research, 25(265), 1–8.