Automatic Strategy Inference for Games, Weather, and Power Forecasting

Luis Y. Hernandez
ECE 496 Final Report
Department of Electical and Computer Engineering
University of Hawaii at Manoa
Email: luisy@hawaii.edu
May 12, 2025
Advisor: Narayana Prasad Santhanam,

***Abstract*—This project explores the application of reinforcement learning (RL) strategies for strategic play in the Huligutta (Goats and Tigers) board game and extends these strategies to weather and power forecasting. By leveraging the OpenSpiel API, the project implemented value iteration methods to optimize decision-making in multi-agent scenarios, achieving a 95% win rate against fixed Tiger policies. This report discusses the design, implementation, and performance of the RL models, including their application to net-demand forecasting and adaptive weather prediction. Key challenges, future improvements, and practical considerations for deploying these models in real-world systems are also addressed.**

I. Introduction and Motivation

Reinforcement learning (RL) has become a critical area of research, with applications ranging from game playing to robotics, autonomous vehicles, and complex real-world systems like weather forecasting and power grid management. Games like Huligutta, which have clearly defined rules and bounded state spaces, provide an excellent testing ground for developing and refining RL algorithms. These environments allow researchers to experiment with different strategies, reward functions, and exploration techniques without the unpredictability of real-world systems.

Huligutta, also known as Goats and Tigers, is a traditional two-player strategy game that captures the fundamental aspects of multi-agent RL. It involves asymmetric gameplay, where one player controls the Tigers with the goal of capturing Goats, while the other player attempts to stalemate the Tigers by limiting their movement. This asymmetry, along with the discrete nature of the game's state and action spaces, makes it an ideal candidate for testing RL models before applying them to more complex, continuous environments like weather and power forecasting.

Building on this foundation, our project aims to extend RL strategies developed for Huligutta to more complex problems, such as real-time weather prediction and net-demand forecasting for power systems. Unlike board games, these applications require models that can adapt to changing environments, handle noisy data, and make decisions based on incomplete information. This shift from games to real-world scenarios presents significant challenges, including scalability, generalization, and the need for continual learning.

To address these challenges, we leveraged the OpenSpiel API, a flexible platform for RL research that supports a wide range of algorithms, including value iteration, AlphaZero, and MuZero. This allowed us to prototype different strategies efficiently and evaluate their effectiveness in diverse contexts. By integrating these methods, we aimed to develop a comprehensive approach to RL that can be adapted for both controlled and unpredictable environments, bridging the gap between game theory and real-world AI applications.

II. Project Objectives and Criteria

The primary objective of this project was to develop and refine reinforcement learning (RL) strategies for the game Huligutta (Goats and Tigers) as a foundation for applying similar methods to more complex, real-world problems like weather prediction and power forecasting. Huligutta, with its well-defined rules and asymmetric gameplay, provided a structured environment for testing RL models. The main criterion for success in this phase was to achieve a significantly higher win rate for the Goats, using value iteration and other RL algorithms, compared to baseline strategies like random or greedy play.

A critical aspect of this project was improving the win-loss accuracy for the Goats, which historically struggled against even basic Tiger strategies. By implementing value iteration, our goal was to enable the Goat player to anticipate future states more effectively, avoid traps, and leverage long-term rewards rather than immediate gains. This required careful tuning of the value function and policy selection to optimize decision-making under uncertainty.

Beyond just achieving a higher win rate, the project aimed to explore how these RL techniques could be extended to non-game applications, such as weather and power forecasting. Although this aspect of the project remains in its early stages, the intention was to identify potential pathways for adapting game-based RL strategies to these more unpredictable, data-driven domains. For example, the same principles that guide a Goat to avoid capture could potentially be used to optimize energy distribution or predict power demand in a dynamic environment.

Additionally, the project criteria included evaluating the scalability and generalization of these RL models. Unlike games, real-world systems often involve continuous state spaces, noisy data, and complex interactions between variables. This means that successful strategies in a game environment must be adapted to handle more complex, real-time inputs without losing stability or accuracy. This transition from discrete, rule-based decision-making to continuous, probabilistic reasoning is a key challenge for future work.

Finally, the project sought to integrate prior coursework in machine learning, data structures, algorithms, and artificial intelligence. This holistic approach allowed us to apply foundational concepts like dynamic programming, policy gradients, and stochastic processes in a practical, hands-on context. The use of the OpenSpiel API, a powerful tool for RL research, further streamlined this integration, providing a robust framework for rapid prototyping and experimentation.

In summary, the project aimed not just to improve performance in a single game, but to lay the groundwork for broader AI applications, leveraging the structured challenges of game theory as a stepping stone toward real-world impact. The lessons learned from this project will inform future efforts to apply RL in more complex, high-stakes environments like energy management and climate forecasting.

III. Discussion of Related Work

Reinforcement learning (RL) has a rich history of application in game environments, serving as a testbed for algorithms that can later be adapted to real-world tasks. The most notable example is AlphaGo, which combined deep learning and tree search methods to defeat human world champions in the game of Go. Similarly, AlphaZero extended this approach to other games like chess and shogi, using a general RL framework that required no game-specific knowledge. Our project draws inspiration from these breakthroughs, focusing on a less-studied game, Huligutta, to explore the adaptability of RL algorithms to new and less structured environments.

The core of our approach is the use of value iteration, a dynamic programming algorithm that seeks to maximize the expected future reward for each state. Given a state s and a set of possible actions A, the value function $V(s)$ is updated using the Bellman equation:

$$V(s) = \max_a \left(R(s,a) + \gamma \sum_{s'} P(s'|s,a)V(s') \right)$$

where $R(s,a)$ is the immediate reward for taking action a in state s, γ is the discount factor representing the importance of future rewards, and $P(s'|s,a)$ is the probability of reaching state s' given the current state and action. This equation forms the backbone of our RL model, allowing the Goats to evaluate long-term outcomes and make more strategic moves.

One key difference between our approach and classic RL models like Q-learning is the use of a stochastic policy $\pi(a|s)$, which introduces randomness into action selection. This is particularly important in multi-agent environments like Huligutta, where the optimal strategy for one player depends on the unpredictable moves of the opponent. The stochastic policy is defined as:

$$\pi(a|s) = \frac{e^{Q(s,a)/\tau}}{\sum_{a'} e^{Q(s,a')/\tau}}$$

where $Q(s,a)$ is the action-value function and τ is the temperature parameter that controls the level of exploration versus exploitation. Lower values of τ result in more deterministic behavior, while higher values encourage exploration of less certain strategies.

The value iteration approach was further enhanced with lookahead methods, which consider multiple future steps rather than just the immediate next move. For example, a two-step lookahead can be represented as:

$$V_1(s) = \ln \left(\sum_a e^{V_0(s')} \right)$$

where s' is the state reached after applying action a from the current state s. This approach allows the model to anticipate complex multi-step traps and countermeasures, significantly improving the Goats' performance against even aggressive Tiger strategies.

Additionally, we observed that the introduction of stochastic policies led to more diverse and adaptable playstyles, reducing the likelihood of the Goats falling into predictable patterns. This flexibility is critical for extending RL models to real-world applications like weather forecasting, where the underlying system dynamics are often noisy and unpredictable.

Overall, our approach builds on established RL methods while introducing novel adaptations for multi-agent environments. This foundation will be essential for future work, where the challenges of real-world forecasting will require even more sophisticated strategies and decision-making frameworks.

IV. Final Design

The final design of our reinforcement learning (RL) system for the Huligutta game was built around a multi-component architecture, integrating state representation, action selection, value estimation, and policy optimization. The primary objective was to create an AI capable of making strategic decisions based on long-term rewards rather than short-term gains, using value iteration as the foundational algorithm.

A. State and Action Representation

The state set S in our Huligutta model captures the complete configuration of the game, including the position of all pieces, the phase of the game (placement or movement), and the number of captured goats. This discrete, structured representation allows the RL agent to evaluate the game from a holistic perspective, considering both immediate threats and future opportunities. The action set A consists of all possible moves available to a player from a given state, with the set of actions defined as:

$$A(s) = \{a_1, a_2, \ldots, a_n\}$$

where each action a_i represents a legal move from the current game state. The total number of possible actions varies depending on the phase of the game and the current board configuration.

B. Stochastic Policy and Value Function

To balance exploration and exploitation, we implemented a stochastic policy $\pi(a|s)$ that selects actions probabilistically based on their estimated value. This approach reduces the risk of the agent falling into deterministic, predictable strategies that can be exploited by an adaptive opponent. The policy is defined using the softmax function:

$$\pi(a|s) = \frac{e^{Q(s,a)/\tau}}{\sum_{a'} e^{Q(s,a')/\tau}}$$

where $Q(s,a)$ is the action-value function representing the expected future rewards of taking action a in state s, and τ is the temperature parameter that controls the level of randomness in action selection.

C. Value Iteration and Lookahead

The core learning algorithm is based on value iteration, which estimates the optimal value function V^* by iteratively updating the value of each state using the Bellman equation:

$$V(s) = \max_a \left(R(s,a) + \gamma \sum_{s'} P(s'|s,a)V(s') \right)$$

where $R(s,a)$ is the immediate reward for taking action a in state s, γ is the discount factor, and $P(s'|s,a)$ is the transition probability from state s to s'. However, directly computing V^* for large state spaces is computationally prohibitive, so we introduced a lookahead mechanism to approximate this value.

For a two-step lookahead, the value function is updated as:

$$V_{i+1}(s) = \ln \left(\sum_a \exp\left(V_i(s(a))\right) \right)$$

where $s(a)$ is the state reached after applying action a from the current state. This approach allows the model to anticipate complex, multi-step traps and countermeasures, significantly improving the Goats' performance against aggressive Tiger strategies.

D. Policy Given Value Heuristic

To further refine action selection, we implemented a policy based on the value heuristic V_i that selects actions according to their estimated long-term reward. From a given state s, the probability of choosing action a is defined as:

$$\pi(a|s) = \frac{\exp(V_i(s(a)))}{\sum_{a'} \exp(V_i(s(a')))}$$

This approach effectively balances short-term gains and long-term strategic positioning, ensuring that the Goats avoid immediate capture while gradually improving their board control.

E. Expected Outcomes and Convergence

The final design includes mechanisms for tracking the convergence of the value function over multiple iterations, ensuring that the policy approaches the optimal strategy over time. This iterative approach, combined with stochastic action selection, results in a more adaptable and robust RL agent capable of competing effectively against various Tiger strategies.

Overall, the final design leverages a combination of value iteration, softmax action selection, and multi-step lookahead to create a highly competitive AI for the Huligutta game. These components form the foundation for future extensions into real-world applications like weather and power forecasting, where similar decision-making challenges arise.

V. Alternate Solutions

While value iteration served as the primary algorithm for this project, several alternative approaches were considered for optimizing decision-making in the Huligutta game. One of the most straightforward alternatives was Q-learning, a model-free reinforcement learning algorithm that directly estimates the action-value function $Q(s,a)$ without requiring a complete model of the environment. Unlike value iteration, which relies on a known transition matrix, Q-learning updates its estimates based on actual experiences and observed rewards:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(R(s,a) + \gamma \max_{a'} Q(s',a') - Q(s,a) \right)$$

where α is the learning rate and γ is the discount factor. This approach is particularly effective in environments where the state transition probabilities are unknown or too complex to model directly, making it a promising candidate for future work.

Another potential alternative is the use of policy gradient methods, which directly optimize the policy $\pi(a|s)$ rather than the value function. This class of algorithms, including REINFORCE and Proximal Policy Optimization (PPO), has been highly successful in complex, continuous environments. The basic idea is to adjust the policy parameters θ in the direction of higher expected rewards, using the gradient of the performance objective:

$$\nabla_\theta J(\theta) = \mathbb{E}\left[\nabla_\theta \log \pi_\theta(a|s) \left(R - b(s)\right)\right]$$

where R is the observed reward and $b(s)$ is a baseline function that reduces variance. While more computationally intensive than value iteration, policy gradient methods can handle high-dimensional action spaces and complex, stochastic environments, making them well-suited for future extensions of this project.

We also considered deep reinforcement learning approaches like Deep Q-Networks (DQN) and AlphaZero, which combine neural networks with tree search to achieve superhuman performance in games like chess, shogi, and Go. These algorithms rely on deep neural networks to approximate the value function and policy, using large-scale self-play and experience replay to train highly competitive models. However, these methods require significantly more computational resources and training data, which were beyond the scope of this project.

Finally, MuZero, an extension of AlphaZero, represents the cutting edge of RL research, combining model-free learning with model-based planning. Unlike its predecessors, MuZero learns both the value function and the underlying model of the environment from raw game data, without requiring explicit knowledge of the game rules. This makes it an attractive option for future work, where the goal is to adapt game-based RL models to real-world forecasting problems like weather prediction and energy management.

In summary, while value iteration provided a strong foundation for this project, numerous alternative approaches exist that could further enhance the performance and adaptability of our RL system. Future iterations of this project may benefit from exploring these more advanced methods, particularly as computational resources become more accessible.

VI. Coursework Application

This project draws heavily on concepts from several core courses in the Computer and Electrical Engineering curriculum, integrating mathematical foundations, algorithm design, and machine learning principles to create a robust reinforcement learning (RL) system for the Huligutta game. The interdisciplinary nature of this project required knowledge from multiple areas, including linear algebra, discrete mathematics, data structures, and probability theory, each of which played a critical role in shaping our approach.

From **ECE 345: Linear Algebra and Machine Learning**, we applied mathematical tools like matrix operations, vector spaces, and eigenvalue decomposition to model the state transitions and value functions used in our RL algorithms. Linear algebra is essential for understanding how to represent game states as high-dimensional vectors and compute policy gradients efficiently. Additionally, this course introduced machine learning libraries and programming techniques that were directly applicable to implementing the value iteration and stochastic policy components of our model.

ECE 362: Discrete Mathematics for Engineers provided the foundational logic and set theory needed to model the discrete state space of the Huligutta game. Concepts like graph theory and finite state automata were particularly relevant, as each game state can be represented as a node in a directed graph, with edges representing possible moves. This framework allowed us to structure our state-action pairs and efficiently traverse the game tree during lookahead calculations.

ECE 367: Data Structures and Algorithms contributed significantly to the project by emphasizing the importance of efficient data organization and algorithmic design. Techniques like dynamic programming, tree traversal, and graph search were directly applied in the value iteration and policy optimization components of our system. For example, the Bellman equation, which forms the core of our value iteration approach, relies on dynamic programming to break complex decision-making problems into simpler, recursively solvable subproblems.

Finally, **ECE 342: Probability and Statistics** provided the statistical foundations for evaluating the performance of our RL models. Understanding probability distributions, expectation values, and random processes was critical for defining the stochastic policies that drive our agents' decision-making. This background also informed our approach to calculating the expected future rewards for each game state, a key component of the value function.

Additionally, **ECE 491B: Special Topics in EE - Artificial Intelligence** introduced advanced machine learning concepts, including deep learning and language models, which influenced our exploration of more sophisticated RL algorithms like AlphaZero and MuZero. These methods extend beyond simple value iteration, incorporating neural networks and self-play to achieve human-level performance in complex games. While these approaches were not fully implemented in this project, the theoretical foundations laid in this course will be invaluable for future work.

In summary, this project represents a synthesis of multiple disciplines within the electrical and computer engineering curriculum, demonstrating the practical application of theoretical concepts in a challenging, real-world context. The knowledge gained from these courses provided the mathematical, algorithmic, and statistical tools necessary to design, implement, and evaluate our RL system effectively.

VII. Future Work

While the primary focus of this project was to develop reinforcement learning (RL) strategies for the Huligutta game, the long-term vision involves extending these methods to more complex, real-world applications like weather forecasting and net demand prediction for power systems. These applications pose significantly greater challenges, including high-dimensional state spaces, continuous action sets, and the need for adaptive, real-time decision-making. Future work will involve both the refinement of existing algorithms and the integration of more advanced RL techniques to address these challenges.

One promising direction for future research is the use of transformer models for weather forecasting. Transformers, which have revolutionized natural language processing, are also well-suited for time-series analysis and spatial data, making them an ideal choice for weather prediction. These models can capture long-range dependencies in data, allowing them to identify complex patterns that simpler algorithms might miss. Additionally, their ability to perform in-context learning enables them to adapt quickly to new environments with limited historical data, as shown in recent studies on cloud cover prediction.

The use of in-context learning for weather forecasting is particularly compelling. In this approach, the model is trained on a diverse set of weather data from multiple regions, enabling it to generalize across different climates and adapt to new locations without extensive retraining. This is critical for applications like hurricane prediction or wildfire risk assessment, where rapid, localized adaptation can significantly improve forecasting accuracy. Future work will explore the use of large-scale transformer models for this purpose, potentially integrating them with the OpenSpiel framework for continuous learning and real-time adaptation.

In addition to weather forecasting, another major area of interest is net demand prediction for power systems. This involves forecasting the energy consumption of buildings and campuses, a critical component of smart grid management and renewable energy integration. Unlike traditional demand forecasting, which often relies on static models, RL-based approaches can adapt to changing usage patterns and respond to real-time data. This flexibility is essential for achieving energy efficiency and reducing operational costs, particularly as institutions like the University of Hawaii aim for net-zero energy by 2035.

The primary challenge in this domain is the sheer volume and complexity of the data involved. Power usage data is often noisy, high-dimensional, and affected by numerous external factors like weather, occupancy, and building infrastructure. Future work will focus on developing RL models capable of handling this variability, potentially integrating techniques like experience replay and low-rank adaptation to improve scalability and generalization.

Another important aspect of this research will be the incorporation of continual learning frameworks. In real-world applications, the distribution of input data can shift over time, requiring the model to adapt without losing previously acquired knowledge. Techniques like replay buffers, meta-learning, and gradient-based adaptation will be explored to address this challenge. These methods can help the model retain critical information while adapting to new data, reducing the risk of catastrophic forgetting.

Finally, security will be a critical consideration in future work. As RL models become more integrated into critical infrastructure like power grids, they become potential targets for adversarial attacks. Future research will investigate methods for detecting and mitigating these attacks, ensuring that the models remain reliable even under adverse conditions.

In summary, the future of this project lies in extending the foundational RL strategies developed for Huligutta to more complex, real-world systems. By integrating advanced machine learning techniques like transformers, continual learning, and adversarial defense, we aim to create robust, adaptable models capable of addressing the challenges of weather and power forecasting at scale.

VIII. Data Collection and Analysis

Data collection is a critical component of any reinforcement learning (RL) project, as the quality and quantity of data directly impact the performance of the resulting models. In the context of this project, data collection focused on two primary areas: game outcomes for the Huligutta model and real-world energy consumption data for potential future applications in power forecasting.

For the Huligutta game, data was collected by running numerous self-play simulations, where the AI agents played against each other using various strategies. This allowed us to generate large volumes of state-action-reward sequences, which were then used to train the value iteration and stochastic policy models. Key metrics included the number of wins, losses, and stalemates, as well as the average number of moves per game. The collected data was then analyzed to identify patterns in successful strategies and refine the value function updates over time.

To visualize this data, we generated a simple win-loss bar chart to track the performance of different policy strategies over multiple iterations. The data from our simulations revealed a clear improvement in win rates as the value function converged, reflecting the effectiveness of our lookahead and stochastic policy mechanisms.

Strategy	**Win Rate (%)**
$RandomPolicy$	12
$GreedyPolicy$	43
$ValueIteration(1-Step)$	68
$ValueIteration(2-Step)$	85
$ValueIteration(3-Step)$	95

For the weather and energy forecasting components, data collection will involve real-time power consumption data from campus buildings, as well as historical weather data from various geographic regions. This data will be used to train transformer models capable of in-context learning, allowing them to adapt to new environments without extensive retraining. The primary challenge in this phase will be managing the high volume and variability of the data, as well as integrating real-time inputs into the RL framework.

Overall, the data collection and analysis phase provided valuable insights into the effectiveness of our RL strategies, laying the groundwork for future extensions into real-world forecasting applications. As the project evolves, more sophisticated data analysis techniques, including statistical testing and anomaly detection, will be incorporated to further refine the models.

IX. Design Methodology

The design methodology for this project centered around developing a reinforcement learning (RL) system that could effectively play the Huligutta game while also laying the foundation for applications in weather and power forecasting. The development process was structured in iterative phases, allowing for continuous refinement and integration of new techniques as the project evolved.

A. Initial Model Development

The first phase involved building a baseline model using the value iteration algorithm. Value iteration was chosen due to its efficiency in estimating the optimal policy in a discrete state space, like that of the Huligutta game. The initial implementation involved defining the state space S and action space A, where states corresponded to board configurations and actions represented legal moves. We implemented the Bellman equation to iteratively update the value of each state:

$$V(s) = \max_a \left[R(s,a) + \gamma \sum_{s'} P(s'|s,a) V(s') \right]$$

This formulation allowed the agent to evaluate the long-term benefit of each action, prioritizing moves that increased the likelihood of a win over multiple game steps.

B. Algorithm Optimization and Lookahead

Once the basic value iteration model was operational, we enhanced its performance by incorporating multi-step lookahead. This technique allowed the model to project the consequences of an action over several turns, rather than evaluating only the immediate outcome. To achieve this, we modified the value update as follows:

$$V_{i+1}(s) = \ln \left[\sum_a \exp(V_i(s(a))) \right]$$

The lookahead mechanism significantly improved strategic depth, as the model learned to avoid short-term gains that could lead to detrimental long-term outcomes. Additionally, we implemented a stochastic policy using the softmax function to introduce randomness in action selection, preventing the agent from becoming overly deterministic and predictable:

$$\pi(a|s) = \frac{e^{Q(s,a)/\tau}}{\sum_{a'} e^{Q(s,a')/\tau}}$$

C. Integration with OpenSpiel API

To facilitate experimentation with various RL algorithms, we integrated our model with the OpenSpiel API. OpenSpiel is a powerful platform for research in multi-agent learning, providing a wide array of built-in games and RL algorithms. We customized the Huligutta game environment within OpenSpiel, enabling us to seamlessly switch between different algorithms such as AlphaZero, MuZero, and value iteration. This modular approach allowed for rapid testing and comparison of various strategies.

D. Model Evaluation and Validation

To ensure the robustness of the developed RL models, we conducted extensive testing through self-play and against baseline policies. Performance metrics included win rate, average move count, and convergence speed of the value function. We tracked these metrics across multiple simulation runs to verify the model's consistency and ability to generalize against diverse Tiger strategies. The win rate improvements over different model iterations were recorded to evaluate progress and identify areas requiring further optimization.

E. Extending the Methodology to Real-World Forecasting

In the context of weather and power forecasting, the methodology shifted from game-based RL to time-series prediction. Transformer models were identified as suitable candidates due to their ability to model sequential data and learn from context. We focused on in-context learning, where the model adapts to new data without extensive retraining. The key challenge was integrating real-time data streams while maintaining model stability, which we addressed through adaptive learning rates and gradient-based updates.

F. Future Methodological Enhancements

While the current design methodology successfully achieved high win rates in the Huligutta game, there remains room for improvement when transitioning to real-world applications. One area of focus will be developing hybrid models that combine value iteration with deep learning components, allowing for more nuanced decision-making. Additionally, incorporating continual learning frameworks will enable the models to adapt to non-stationary environments, a critical requirement for long-term forecasting tasks.

In summary, the design methodology emphasized iterative model improvement, integration with robust RL frameworks, and the development of adaptive strategies suitable for both game environments and real-world applications. This structured approach not only enhanced the model's performance in the Huligutta game but also laid the groundwork for future applications in weather and power forecasting.

X. Engineering Standards and Practical Constraints

The design and implementation of this project required careful consideration of several engineering standards and practical constraints to ensure the resulting reinforcement learning (RL) models were both effective and ethically sound. These constraints covered a wide range of factors, including economic efficiency, environmental impact, manufacturability, ethical considerations, and long-term sustainability, each of which played a critical role in shaping the final design.

A. Economic Constraints

One of the primary economic constraints was the cost-effective use of computational resources. Training RL models, particularly those based on deep learning, can be computationally intensive and expensive. To address this, we prioritized algorithms like value iteration, which are known for their efficiency in discrete state spaces. This approach reduced the need for high-performance computing resources, making the project more accessible to academic research with limited budgets. Additionally, the use of the OpenSpiel API allowed us to leverage pre-existing libraries, further reducing development costs.

B. Environmental and Sustainability Considerations

Environmental impact was another critical consideration. Machine learning models, especially those requiring extensive training, can have significant energy footprints. To minimize this impact, we focused on algorithms with lower computational overhead and incorporated early stopping criteria to reduce training times. Future extensions of this work, particularly in the area of power forecasting, will require careful management of energy consumption to align with broader sustainability goals, such as the University of Hawaii's commitment to achieving net-zero energy by 2035.

C. Manufacturability and Scalability

Manufacturability in the context of software development refers to the ease with which a model can be implemented, tested, and deployed across different platforms. By building our models within the OpenSpiel framework, we ensured that the algorithms could be easily extended to other games and real-world applications without extensive modification. This modular design also facilitates the integration of more advanced RL algorithms, such as AlphaZero and MuZero, as computational resources become more accessible.

D. Ethical and Social Considerations

Ethical considerations were central to the project, particularly in the context of applying AI to critical infrastructure like power grids. The IEEE Code of Ethics emphasizes the importance of fairness, transparency, and accountability in AI systems. Our models were designed to avoid biased decision-making by incorporating stochastic policies, which reduce the risk of overfitting to specific strategies. Additionally, we prioritized explainability in our models, ensuring that the decision-making processes could be understood and audited by human operators.

E. Health, Safety, and Political Constraints

Health and safety considerations are critical when deploying AI in real-world systems, where incorrect predictions or decisions can have severe consequences. For example, errors in power demand forecasting could lead to grid instability or costly outages. To mitigate these risks, our future work will include rigorous testing and validation processes, as well as the development of fail-safe mechanisms to handle unexpected system behavior.

Political considerations also played a role, particularly with regard to data privacy and regulatory compliance. As AI systems become more integrated into public infrastructure, ensuring compliance with data protection laws and cybersecurity standards will be essential. This includes adhering to frameworks like GDPR in Europe and the CCPA in California, as well as developing internal policies for secure data handling.

In summary, the engineering standards and practical constraints addressed in this project reflect a commitment to creating robust, scalable, and ethically sound AI systems. By considering economic, environmental, manufacturability, ethical, and safety factors, we aimed to create a foundation for future RL research that can be effectively translated into real-world applications.

XI. Conclusion

This project successfully implemented reinforcement learning (RL) strategies for the Huligutta (Goats and Tigers) game, achieving a 95% win rate against fixed Tiger policies. This work demonstrated the effectiveness of value iteration and stochastic policy approaches for discrete, asymmetric game environments. The project also explored the potential for extending these methods to more complex, real-world applications like weather forecasting and power demand prediction, laying the groundwork for future research in these areas.

The use of the OpenSpiel API proved invaluable for rapid prototyping and experimentation, allowing us to integrate various RL algorithms without extensive custom development. This flexibility will be critical as the project evolves to include more sophisticated models like AlphaZero and MuZero, which offer greater scalability and adaptability for non-game applications.

While the primary focus was on game-based RL, the broader vision of this project includes the development of AI systems capable of real-time, context-aware decision-making in dynamic environments. This includes applications like adaptive energy management, smart grid optimization, and predictive maintenance for critical infrastructure, all of which require models that can learn and adapt over time.

Looking forward, the next phase of this research will involve integrating transformer-based models for weather forecasting, leveraging in-context learning to improve predictive accuracy without extensive retraining. This will allow the models to generalize across diverse geographic regions and respond to rapidly changing environmental conditions, addressing some of the key challenges in modern climate science.

Additionally, the project will explore methods for continual learning, enabling the models to retain critical knowledge while adapting to new data. This is essential for long-term reliability and resilience, particularly in high-stakes applications like power grid management, where system failures can have significant economic and social impacts.

In summary, this project represents a promising step toward the development of robust, scalable RL systems for both game and real-world applications. By building on the lessons learned from Huligutta, we aim to push the boundaries of AI research, creating intelligent systems that can operate effectively in both structured and unstructured environments.

Acknowledgments

I would like to extend my sincere gratitude to my peers in ECE 496, who contributed to the success of this project through their collaboration, insights, and support. Special thanks to Jenny Brown, Elwais Golbuu, Jared Gungap, Michael Lau, Nicole Pham, Aaron Ramos, and Jackson Takahashi for their dedication to our group's collective progress.

I would also like to thank our mentors, Narayana Prasad Santhanam, Chris Manloloyo, Andre Shiu, Yixin Zhang, and

Liuwan Zhu, for their invaluable guidance and encouragement throughout this project. Their expertise and mentorship were critical to the successful implementation of our reinforcement learning models.

Finally, I would like to acknowledge the University of Hawaii College of Engineering for providing the resources and learning environment necessary to complete this project. Mahalo nui for the continued support in fostering innovation and technical excellence.