รายงานการวิเคราะห์พฤติกรรมบอทเทรดอัจฉริยะ
กรณีศึกษา: ปัญหา Gradient Explosion ในสถาปัตยกรรม PPO (26 Features / Multi-timeframe: OFF)

1. บทสรุปผู้บริหาร (Executive Summary)

รายงานฉบับนี้ประเมินผลการรันบอทเทรดอัจฉริยะโดยใช้ข้อมูล State Space จำนวน **26 มิติ** (ปิดการใช้งาน Multi-timeframe) เพื่อลดสัญญาณรบกวนจากข้อมูลที่สูญหาย (NaN)
ผลการทดสอบชี้ให้เห็นถึงสภาวะ **การพังทลายของโครงข่ายประสาท (Policy Collapse และ Gradient Explosion)** อย่างรุนแรงตั้งแต่เริ่มต้นกระบวนการเรียนรู้ ทำให้บอทสูญเสียความสามารถในการปรับตัวและยึดติดกับแอ็กชันชุดเดิมตลอดการฝึกสอนกว่า 300,000 Timesteps
อย่างไรก็ตาม เป็นความบังเอิญทางสถิติที่พฤติกรรมการเทรดแบบคงที่ (Static Policy) ชุดแรกที่บอทสุ่มเจอสามารถทำผลงานในตลาดย้อนหลังช่วง Out-of-Sample (2024-2026) ได้อย่างดุดัน โดยสร้าง**ผลตอบแทนทบต้น (CAGR) ที่ 19.37% ต่อปี** แต่ต้องเผชิญกับระดับความเสี่ยงที่สูงมาก (**Max Drawdown -22.76%**) ซึ่งบ่งชี้ว่าระบบขาดความเสถียรและไม่พร้อมสำหรับการนำไปใช้งานจริง

2. โครงสร้างข้อมูลและการตั้งค่า (Data Setup & Integrity)

การปิดโหมด Multi-timeframe ทำให้ชุดข้อมูลมีความสะอาดและต่อเนื่องสูง ไม่มีปัญหา Data Shift จากการต้องเติมค่า 0.0 เหมือนในเวอร์ชันที่ใช้ 30 Features

- **Training Set (1999 - 2019):** ตัดช่วง Warm-up ออก 1,074 แถว (ข้อมูลสมบูรณ์)
- **Validation Set (2020 - 2023):** ตัดช่วง Warm-up ออก 597 แถว (ข้อมูลสมบูรณ์)
- **Testing Set (2024 - 2026):** ตัดช่วง Warm-up ออก 597 แถว (ข้อมูลสมบูรณ์)

ข้อสังเกต: เนื่องจากชุดข้อมูลไม่มีช่องโหว่ ปัญหาความล้มเหลวของบอทเทรดในรันนี้จึงเกิดจาก **โครงสร้างสมการและการปรับตั้งค่า Hyperparameters (Architecture & Hyperparameter Mismatch)** โดยตรง ไม่ใช่ปัญหาจากคุณภาพของข้อมูล

3. วิเคราะห์สภาวะการเรียนรู้ที่ผิดปกติ (Training Anomalies & Loss Analysis)

บันทึกการเรียนรู้ (Log) แสดงให้เห็นถึงการแตกกระจายของค่าน้ำหนักในระบบ (Weight Destabilization) อย่างชัดเจน:

1. สภาวะหยุดการเรียนรู้ฉับพลัน (Premature Policy Collapse)

- ที่สเตป 2,048 ค่า mean_reward สุ่มเจอจุดที่ให้ผลตอบแทนสูงถึง **43.77** (Episode 10 ดันสินทรัพย์ไปที่ 1,444,015.04 USD ด้วยการเทรด 806 ครั้ง)
- หลังจากจุดนั้นจนจบ 303,104 สเตป ค่าผลตอบแทน, สินทรัพย์สุทธิ และจำนวนการเทรด ถูกล็อกตายตัว ไม่มีการขยับหรือพัฒนาหาแนวทางทำกำไรใหม่ๆ อีกเลย

2. การระเบิดของค่าคลาดเคลื่อน (Gradient Explosion)

ระบบสูญเสียการควบคุมสมการคณิตศาสตร์ภายใน PPO สังเกตได้จากสเตปท้ายๆ:

- **Actor Loss:** ระเบิดจากหลักพัน ไปแตะระดับ **6.04e+10** (หกหมื่นล้าน)
- **Critic Loss:** พุ่งทะลุไปถึง **3.55e+18** ซึ่งหมายความว่า Value Network (ตัวประเมินมูลค่าสถานะ) คำนวณตัวเลขผิดพลาดจนล้นขีดจำกัด
- **Entropy Coefficient (ent_coef):** พุ่งไปที่ **1.24e+08** แสดงให้เห็นว่าระบบพยายามจะบังคับให้บอทออกจากสภาวะล็อกตาย แต่สมการพังทลายไปแล้ว

4. ผลการทดสอบในสภาวะตลาดจริง (Out-of-Sample Generalization)

กลยุทธ์ที่บอทยึดติดมาจากการพังทลายของระบบ ดันบังเอิญสอดคล้องกับพฤติกรรมตลาดขาขึ้นอย่างรุนแรงในช่วงปี 2024-2026 จึงให้ผลลัพธ์เชิงตัวเงินที่ดูน่าประทับใจ:

- **ผลการทำกำไร (Return):** กำไรสะสม **36.61%** (CAGR **19.37%**) ซึ่งเป็นอัตราการเติบโตที่เอาชนะดัชนีตลาดส่วนใหญ่ได้
- **การควบคุมความเสี่ยง (Risk Management):** แย่ลงอย่างเห็นได้ชัด Max Drawdown ทรุดลงไปถึง **-22.76%** เนื่องจากบอทไม่รู้วิธีป้องกันตัวเมื่อตลาดปรับฐาน
- **ประสิทธิภาพ (Efficiency):** Sharpe Ratio ตกมาอยู่ที่ **0.91** (ความผันผวนสูงเกินไปเมื่อเทียบกับกำไร) แลกมากับ Win Rate ระดับ 55.86%

5. แนวทางการแก้ไขเชิงวิศวกรรม (Engineering Recommendations)

การนำโมเดลที่มีสภาวะ Gradient Explosion ไปรันบอทเทรดจริงบนเซิร์ฟเวอร์ตลอด 24 ชั่วโมง มีความเสี่ยงที่จะเกิดเหตุการณ์ล้างพอร์ตได้หากสภาวะตลาดเปลี่ยนทิศ (Regime Shift) ควรดำเนินการแก้ไขโครงสร้างดังนี้:

1. **ทำ Normalization แบบไดนามิก (Environment Wrappers):** ค่า Critic Loss ที่ระเบิดมักเกิดจากตัวแปรเศรษฐกิจมหภาค (เช่น m2, vix) มีสเกลที่แตกต่างกันรุนแรง ควรหุ้ม Environment ด้วย VecNormalize เพื่อจัดสเกลทั้ง Observation และ Reward ให้อยู่ในค่ามาตรฐาน [-5, 5] เสมอ
2. **จำกัดขนาด Gradient (Gradient Clipping):** เพิ่มพารามิเตอร์ max_grad_norm=0.5 ใน PPO เพื่อป้องกันไม่ให้การอัปเดตน้ำหนักของรอบใดรอบหนึ่งมีค่ากระโดดจนทำลายโครงข่ายประสาท
3. **ลด Learning Rate (LR):** การใช้ LR คงที่ที่ 0.00025 อาจรุนแรงไปสำหรับข้อมูลชุดนี้ แนะนำให้ปรับลดลงเหลือ 0.00005 หรือใช้ฟังก์ชันลด LR อัตโนมัติ (Linear Decay)

ต้องการให้ผมช่วยร่างโค้ดตัวอย่างสำหรับการฝัง VecNormalize และกำหนดพารามิเตอร์ป้องกัน Gradient Explosion ในสคริปต์เทรนโมเดลเลยไหมครับ?