



เวลาในการตาย : การทำนายการตายใน DOTA2 โดยการใช้การเรียนรู้เชิงลึก

Time To Die: Death Prediction in DOTA2 Using Deep Learning

นายกรกฎ ไกรศรี โกศล

โครงการนี้เป็นส่วนหนึ่งของการศึกษาในรายวิชา 040223491 Seminar
สาขาวิชาคณิตศาสตร์เชิงวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์
คณะวิทยาศาสตร์ประยุกต์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

บทคัดย่อ

Esports ได้กลายเป็นกีฬาระดับนานาชาติที่มีผู้ชมถึงหนึ่งร้อยล้านคน Esports Games ได้สร้างข้อมูลที่มาจากการวัดเป็นจำนวนมหาศาลขึ้นมาในการใช้ข้อมูลเหล่านี้มาทำนายผลลัพธ์การแข่งขัน Esports ได้รับความสนใจเป็นอย่างมาก แต่ว่า micro-predictions พยายามทำนายเหตุการณ์ในการแข่งขันที่มันยังไม่รู้จัก Micro-predictions เป็นสิ่งที่น่าสนใจสำหรับนักวิจารณ์และผู้ชมเพราะว่าพวกเขาเหล่านั้นให้ความสามารถในการสังเกตเหตุการณ์ที่อาจจะพลาดไป Esports Games มีความซับซ้อนอย่างมากพร้อมการกระทำที่รวดเร็วซึ่งทำให้สมดุลของเกมสามารถเปลี่ยนแปลงได้ในเวลาสั้น ๆ และเหตุการณ์ที่สามารถเกิดขึ้นพร้อมกันในบริเวณที่เดียวกันนั้นซึ่งเหตุการณ์ที่ว่ามาสามารถเกิดขึ้นได้อย่างรวดเร็วและมันง่ายสำหรับนักวิจารณ์และผู้ชมที่จะพลาดชมเหตุการณ์และสังเกตผลกระทบของเหตุการณ์ใน DOTA2 ซีรีส์ของผู้เล่นที่ถูกสังหารโดยทีมฝั่งศัตรูจะเป็นเหตุการณ์ที่น่าสนใจสำหรับนักวิจารณ์และผู้ชม พวกเรานำเสนอการเรียนรู้เชิงลึกเครือข่ายด้วย weight sharing ซึ่งทำให้การทำนายการตายได้แม่นยำภายใน 5 วินาที เครือข่ายถูกฝึกฝนเกี่ยวกับการเล่น DOTA2 ด้วยชุดข้อมูลการแข่งขันของผู้เล่นระดับมืออาชีพแม้ว่าเหตุการณ์การตายจะเกิดขึ้นได้ยากภายในเกม (1%ของข้อมูล) แบบจำลองมีตัวประสิทธิภาพ Precision 0.377 กับ Recall 0.725 ในชุดข้อมูลทดสอบเมื่อได้รับการแจ้งมาว่าให้ทำนายว่าจะมีผู้เล่นคนใดคนหนึ่งจาก 10 คนตายภายใน 5 วินาที ในตัวอย่างของระบบที่ใช้กับการแข่งขัน DOTA2 ที่ถูกนำเสนอนี้ ตัวแบบจำลองสามารถทำนายการตายแบบเรียลไทม์ใน DOTA2 ได้ซึ่งเป็น Esports ที่มีผู้เล่นมากที่สุดในโลกทำให้นักวิจารณ์และผู้ชมมีเวลาให้ความสนใจกับเหตุการณ์สำคัญเหล่านี้

คำสำคัญ : การเรียนรู้เชิงลึก Esports DOTA2 Micro Predictions

สารบัญ

บทคัดย่อ	ก
สารบัญ	ข
สารบัญรูปภาพ	ค
สารบัญตาราง	ง
บทที่ 1 บทนำ	
1.1 ความเป็นมาและความสำคัญ	1
บทที่ 2 งานที่เกี่ยวข้อง	
2.1 งานที่เกี่ยวข้อง	3
บทที่ 3 ขั้นตอนและวิธีดำเนินการ	
3.1 ขั้นตอนและวิธีดำเนินการ	4
บทที่ 4 ผลการทดลอง	
4.1 การประเมินผล	10
บทที่ 5 สรุปผลการทดลอง	
5.1 การอภิปรายผล	13
5.2 การสรุปผล	13
เอกสารอ้างอิง	15
ภาคผนวก	17

สารบัญภาพ

ภาพที่

1. ภาพรวมของสถาปัตยกรรมของระบบเครือข่าย	8
2. การกระจายของการทำนายสำหรับจุดข้อมูลด้วยเวลาที่แตกต่างกัน	10
3. เส้นโค้งของ Precision-Recall	11
4. แสดงผลลัพธ์ของตัวอย่างเกมแบบสุ่ม	11
5. ส่วนที่แรเงาของรูปภาพที่ 4	12
6. ส่วนที่แรเงาของรูปภาพที่ 4	12

สารบัญตาราง

ตารางที่

1. รายละเอียดความแตกต่างชนิดของคุณลักษณะ	6
2. ไฮเพอร์พารามิเตอร์สำหรับแต่ละชุดคุณลักษณะ	9
3. ค่าเฉลี่ย Precision(พื้นที่ใต้เส้นโค้ง Precision-Recall)	10

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

คำว่า Esports อธิบายถึงวิดีโอเกมที่สามารถแข่งขันได้และมีผู้รับชมเป็นจำนวนมาก Superdata ได้ทำนายว่าจะมีผู้ชม Esports ถึง 330 ล้านคนภายในปี 2562 Esports กลายเป็นงานวิจัยสำคัญของในวงการศึกษาศาสตร์และอุตสาหกรรม เนื่องจากข้อมูล หามาได้ง่ายและมีมิติที่สูงและปริมาณที่สูงจากแทบทุกการแข่งขันข้อมูลเหล่านี้ทำให้เกิดการวิเคราะห์ Esports กระบวนการนี้ใช้ข้อมูลที่เกี่ยวข้องกับ esports เพื่อหารูปแบบที่มีความหมายและแนวโน้มในข้อมูลและการสื่อสาร รูปแบบเหล่านี้ด้วยการทำให้อยู่ในรูปแบบภาพเพื่อช่วยในการตัดสินใจ การออกแบบนี้เน้นพื้นฐานใน esports ทำให้ผู้ชม สามารถเข้าใจการแข่งขันได้ Esports หลายประเภทนั้นมีความซับซ้อนและดำเนินการกระทำอย่างรวดเร็วยากที่จะดูการแข่งขันที่ถ่ายทอดสดด้วยตาเปล่า

การวิเคราะห์ Esports มีขอบเขตสำคัญที่มุ่งเน้นไปที่เกมประเภท Multiplayer Online Battle Arena (MOBA) ในที่นี้ได้ ยกตัวอย่างเกมประเภท MOBA คือ DOTA2 โดยว่าลอร์ดออฟเธอร์แลนด์เป็นเกมที่มีผู้เล่นทั้งหมด 10 คน โดยแต่ละทีมจะมีผู้เล่น 5 คน (โดยปกติแล้วเกมประเภทนี้ใช้เวลาเล่นประมาณ 30 – 40 นาทีต่อการแข่งขัน) DOTA2 เป็นเกมที่มีความซับซ้อนมากผู้เล่นแต่ละคนจะต้องเลือกฮีโร่แต่ละตัวที่ไม่ซ้ำกันในการแข่งขันแต่ละผู้เล่นจะต้องใช้ความสามารถฮีโร่กับไอเทมที่ซื้อได้จากในเกมจากการกำจัดครีปทั้ง 3 ทาง ฮีโร่แต่ละตัวมีหน้าที่แตกต่างกันแต่มีเป้าหมายเหมือนกันคือกำจัดครีปสะสมเงินในเกม ต่อสู้กับทีมศัตรูเพื่อพัฒนาเลเวลของฮีโร่ให้แข็งแกร่งมากกว่าทีมศัตรู เป้าหมายหลักสำคัญของเกมนี้คือการทำลายฐานทัพใหญ่ของทีมศัตรู ในแต่ละทีมจะมีกลยุทธ์มากมายและการชนะเกมนี้ต้องมีความร่วมมือกันระหว่างผู้เล่นในทีมและต้องสามารถโต้ตอบกลยุทธ์ของทีมศัตรูได้อีกด้วย เกมนี้เป็นแบบเรียลไทม์ด้วยข้อมูลที่ซ่อนอยู่และมีการเล่นด้วยกลยุทธ์ขั้นสูง

ในเกม DOTA2 ผู้เล่นจะต้องคอยตรวจสอบสถานะของฮีโร่อย่างสม่ำเสมอโดยเฉพาะในช่วงที่มีโอกาสถูกสังหาร ดังนั้นวิธีการทำนายการตายใน DOTA2 สามารถเป็นประโยชน์ได้ในหลายระดับผู้เล่นมืออาชีพและนักวิเคราะห์เกม ในปัจจุบันการวิเคราะห์ esports มีตัวชี้วัดประสิทธิภาพหลายตัว(ตัวแปรข้อมูล)ที่เคยใช้แล้วทีมประสบความสำเร็จมากกว่า โอกาสที่ผู้เล่นจะถูกสังหาร โดยในบทที่ 5 เราได้แสดงให้เห็นว่าตัวแปรพลังชีวิตของฮีโร่จะเป็นตัวบ่งชี้ชี้วัดของความน่าจะเป็นที่จะถูกสังหารไม่ได้เกี่ยวข้องกับการตายของฮีโร่ ฮีโร่บางตัวมีความสามารถฟื้นฟูพลังชีวิตของตัวเองหรือฟื้นฟูพลังชีวิตของเพื่อนร่วมทีมในขณะที่ตัวฮีโร่ก็สามารถซื้อไอเทมในเกมเพื่อฟื้นฟูพลังชีวิตหรือเคลื่อนย้ายออกจากสถานการณ์ที่อันตรายได้ ดังนั้นเราจึงไม่สามารถใช้พลังชีวิตสำหรับทำนายการตายได้เนื่องจากการทำนายมีความซับซ้อน และต้องระมัดระวังจะทำให้เกิดความผิดพลาดและนำไปสู่การวิเคราะห์ที่เอนเอียง

ในวิจัยนี้แนะนำขั้นตอนแรกของ micro-predictions ใน esports Micro-predictions เป็นการทำนายเล็ก ๆ เกี่ยวกับสิ่งที่จะเกิดขึ้นในอนาคตอันใกล้ในเกมเมื่อเทียบด้วยการทำนายผลลัพธ์ของการแข่งขันนั้นแนวคิดนี้เช่นเดียวกับการกีฬา ยกตัวอย่าง การทำนายว่านักฟุตบอลคนไหนจะทำประตูเป็นคนต่อไป การทำความเข้าใจความน่าจะเป็นของการตายในอนาคตอันใกล้มีความสำคัญต่อการกำหนดกลยุทธ์ที่ผู้เล่นควรใช้ในการวิเคราะห์หลังเกมจบ การใช้เครื่องมือการทำนายในเกมถ่ายทอดสดระดับมืออาชีพจะทำให้ความสามารถในการอ่านเกมของผู้ชมมากขึ้นและเมื่อผู้เล่นเล่นพลาดพวกเขาควรจะใช้กลยุทธ์เป็นรายคนหรือทั้งทีม การถ่ายทอดสดน่าจะให้ประโยชน์จากข้อมูลเหล่านี้กลับไปยังผู้ชมพร้อมกล้องอัตโนมัติจับการเคลื่อนไหวเมื่อไหร่ที่การทำนายนี้เกิดขึ้นในระดับที่ความแม่นยำสูงพอสมควรทำให้ทั้งนักวิจารณ์และผู้ชมได้รับ

ประสบการณ์รับชมที่ราบรื่น นอกจากนี้การวิเคราะห์ข้อมูลหลังเกมจบเป็นส่วนหนึ่งของการฝึกอบรมผู้เล่นระดับมืออาชีพ โค้ชในปัจจุบันต้องเลือกจุดของเกมที่น่าไปดูการจบเกมหรือเมื่อไหร่ที่ทีมได้รับความได้เปรียบ (snowballed) ด้วยการเข้าถึงเครื่องมือของเราสามารถสร้างแบบจำลองทำนายการตายว่าตายเมื่อไหร่หรือไม่ก็ตรวจสอบว่าทำไมตัวแบบจำลองถึงบอกว่าอันตราย เรานำเสนอการทำนายแบบใหม่โดยมีส่วนความจำของ neural network (อธิบายไว้ในส่วนที่ 3) การวิเคราะห์ชุดขนาดใหญ่กว่าของในเกมเมื่อเทียบกับงาน esports ก่อนหน้านี้

การออกแบบการทดลองเบื้องหลังคือการฝึกแบบจำลอง neural network เพื่อทำนายว่าฮีโร่จะตายใน 5 วินาทีถัดไปหรือไม่ ภายใน 5 วินาทีให้เวลามากพอที่จะก่อนจะเกิดการสังหารเกิดขึ้น ดังนั้นนักวิจารณ์สามารถใช้ประโยชน์จากข้อมูลการทำนายใด ๆ ที่เครือข่ายให้ไว้ เราได้ทำการทดลองเบื้องต้นจาก 5 วินาที จนถึง 20 วินาทีโดยความแม่นยำของโมเดลลดลงอย่างมากเมื่อเวลาเพิ่มขึ้น

มีข้อจำกัดบางอย่างทันทีที่การที่ทุกจุดข้อมูลต้องการแม้แต่ข้อมูลฮีโร่ที่ไม่ชัดเจนเครื่องมือจะจำกัดการวิเคราะห์หลังเกมจบหรือในสภาพแวดล้อมที่นักพัฒนา ผู้ประกาศสามารถรับตัวแปรที่ต้องการได้ (ยกตัวอย่าง การถ่ายทอดสดเกมระดับมืออาชีพ) ข้อจำกัดอีกอย่างคือเกมมีการเปลี่ยนแปลงตลอดเวลาหมายความว่าแบบจำลองเราอาจจะใช้ได้ไม่ดีกับเกมรุ่นใหม่

โครงสร้างของวิจัยนี้มีดังนี้ งานที่เกี่ยวข้องจะถูกกล่าวถึงในหัวข้อที่ 2 ต่อมาการวิเคราะห์เชิงลึกของชุดข้อมูลรวมถึงการทำความสะอาดและคุณลักษณะตามขอบเขตการศึกษาในหัวข้อที่ 3 และในหัวข้อที่ 3 เราจะทำการวิเคราะห์แบบจำลอง neural network ที่ใช้จากข้อมูลที่ไม่สมดุลจนถึงการวิเคราะห์ในเชิงลึกสำรวจไฮเปอร์พารามิเตอร์จากนั้นเราจะสรุปผลลัพธ์ของแบบจำลองที่ถูกฝึกมาแล้วและวิธีการที่นำไปใช้กับข้อมูลทดสอบและได้เลือกการแข่งขัน DOTA2 แบบเรียลไทม์ในหัวข้อที่ 4 ส่วนในหัวข้อที่ 5 กล่าวถึงประสิทธิภาพการทำงาน เราได้ให้ข้อสรุปในส่วนที่ 5 มีซอร์สโค้ดที่ใช้สำหรับสร้างชุดข้อมูลและฝึกแบบจำลองที่ https://github.com/adam-katona/dota2_death_prediction

บทที่ 2

งานที่เกี่ยวข้อง

2.1 งานที่เกี่ยวข้อง

โดยรวมแล้วอุตสาหกรรม esports สร้างความรู้ที่เพิ่มขึ้นอย่างรวดเร็วแม้ว่าความรู้ส่วนใหญ่จะไม่สามารถเปิดเผยต่อสาธารณชนได้เนื่องจากมีเนื้อหาเชิงพาณิชย์ ดังนั้นการสร้างความรู้ที่ทันสมัยจึงเป็นสิ่งที่ท้าทายใน esports การวิจัย esports ก่อนหน้านี้มีการตีพิมพ์ในหลายสาขาวิชาที่แตกต่างกันรวมถึง ปัญญาประดิษฐ์ การวิเคราะห์หัตถ์วิทยา จิตวิทยา การศึกษา การทำให้เห็นภาพ ชาติพันธุ์วิทยา การตลาด การจัดการและธุรกิจ

ภายใน esports การวิเคราะห์ esports โดยทั่วไปมักจะใช้เพื่อแสดงว่างานข่าวกรองธุรกิจมีศูนย์กลางที่ esports เช่น การวิเคราะห์ข้อมูลที่มาจากการวัดพฤติกรรม ข้อมูลการขายและอื่น ๆ การวิเคราะห์เหล่านี้เกิดขึ้นในช่วงของสาขาวิชาที่เกี่ยวข้องและอธิบายถึงขอบเขตการทำงาน งานของเราเน้นไปที่การวิเคราะห์ข้อมูลที่มาจากการวัดพฤติกรรมจาก esports game

ส่วนใหญ่งานวิชาการได้เน้นการสร้างแบบจำลองการเรียนรู้ของเครื่องจักรเพื่อทำนายผลลัพธ์ของ esports หรือการทำนายอื่น ๆ งานอื่นที่เกี่ยวข้องได้แก่ Schubert และคณะ ผู้ที่อธิบายวิธีตรวจสอบเวลาและสถานที่การเผชิญหน้าทีมใน DOTA2 Summerville และคณะ ใช้การเรียนรู้เครื่องจักรเพื่อทำนายการเลือกตัวฮีโร่ใน DOTA2 Rioult ได้พูดคุยเกี่ยวกับการใช้ทั่วไปของแท็กผู้เล่นจากการทำเหมืองจาก esports games Drachen และคณะ ใช้การจำแนกประเภทเพื่อการศึกษาแบบการเคลื่อนไหวของทีม DOTA2 ผ่านระดับทักษะ Gao และคณะ จำแนกประเภทฮีโร่ DOTA2 ตามการวัดประสิทธิภาพ Eggert และคณะ ใช้การถดถอยโลจิสติกส์เพื่อจำแนกผู้เล่นออกเป็นบทบาทที่กำหนดไว้ล่วงหน้าโดยใช้ตัวชี้วัดประสิทธิภาพ

งานที่เกี่ยวข้องที่ใกล้เคียงที่สุดคืองานของ Clegghem และคณะ ผู้ที่ทำนายพลังชีวิตฮีโร่ใน DOTA2 โดยใช้เทคนิคต่าง ๆ แบบจำลองค่าเฉลี่ยเคลื่อนที่แบบถดถอยอัตโนมัติเพื่อทำนายการเปลี่ยนแปลงเล็กน้อย ในพลังชีวิตและแบบจำลองการประมาณทางสถิติ ซึ่งการทำนายเปลี่ยนแปลงพลังชีวิตและการทำงานร่วมกับการถดถอยโลจิสติกส์เชิงเส้นที่ทำนายสัญญาณและขนาดของการเปลี่ยนแปลงตามลำดับ ผู้เขียนระบุถึงความแม่นยำสูงสุดร้อยละ 77.2 สำหรับแบบจำลอง ARMA เพื่อทำนายการเปลี่ยนแปลงพลังชีวิตเล็กน้อย (น้อยกว่า 100) ใน 5 วินาที แบบจำลองการประมาณทางสถิติทำนายว่าการเปลี่ยนแปลงพลังชีวิตครั้งใหญ่จะเกิดขึ้นในอนาคต 10 วินาที อย่างไรก็ตามแบบจำลองมีประสิทธิภาพต่ำในการทำนายจุดที่เปลี่ยนแปลงพลังชีวิตแม้ว่าจะมีความแม่นยำอยู่ที่ประมาณร้อยละ 80 สำหรับทิศทางและขนาดของการเปลี่ยนแปลงพลังชีวิต แบบจำลองของ Clegghem และคณะ ใช้แค่ข้อมูลพลังชีวิตเท่านั้นต่างจากงานวิจัยในปัจจุบันที่ใช้คุณลักษณะหลายตัวมาก นอกจากนี้ชุดข้อมูลของเขาประกอบด้วย 542 การแข่งขัน และไม่ได้ระบุถึงระดับของการแข่งขันไว้ในงานวิจัย ดังนั้นจึงไม่สามารถทราบผลลัพธ์ได้ว่าจะใช้กับการแข่งขันระดับมืออาชีพหรือมือใหม่ได้

โดยสรุปแม้ว่างานวิจัยก่อนหน้านี้มุ่งเน้นไปที่ macro-predictions (ทำนายทีมที่ชนะ) มีงานจำนวนเล็กน้อย เช่น การเผชิญหน้าและการเปลี่ยนแปลงพลังชีวิตของฮีโร่แต่มีแค่เพียงงานเดียวเท่านั้นที่มุ่งเน้นไปที่การพยากรณ์ในเกมด้วยความสำเร็จที่หลากหลาย ไม่มีงานที่ทำอยู่ในขณะนี้ที่มุ่งเน้นไปที่ระดับมืออาชีพ ระดับมือใหม่และการทำนายตายของฮีโร่ เหตุการณ์การตายของฮีโร่นั้นเป็นส่วนสำคัญในการบรรยายเกมและมีความสำคัญพอๆกับผู้บรรยาย นักวิจารณ์ ผู้ชมเกี่ยวกับสถานการณ์ในเกมและกลยุทธ์ของผู้เล่นและการวิเคราะห์ จากความรู้ที่ทันสมัยเราสามารถสรุปได้อีกว่าต่อไปการวิเคราะห์ esports จะมีความซับซ้อนและต้องการความรอบคอบจากข้อมูลและแบบจำลองที่ใช้หลายครั้ง

บทที่ 3

ขั้นตอนและวิธีการดำเนินการ

3.1 ขั้นตอนและวิธีการดำเนินการ

ชุดข้อมูล

จำเป็นต้องมีการเข้าถึงข้อมูลเป็นจำนวนมากเพื่อฝึกแบบจำลองจำแนกประเภททำนายการสังหารได้สำเร็จ ในการสร้างแบบจำลองนี้เราได้ใช้ไฟล์ DOTA2 ย้อนหลัง เป็นไบนารีไฟล์โดย วาล์วคอร์ปอเรชัน สำหรับทุกการแข่งขัน DOTA2 ที่เล่น

วาล์วคอร์ปอเรชันมีเหตุการณ์เกมระดับมือใหม่จากการแข่งขัน เป็นการนำเสนอที่ครอบคลุมการแข่งขันและใช้โดย DOTA2 engines เพื่อสร้างการแข่งขันใหม่สำหรับการดูซ้ำและการวิเคราะห์ ดังนั้นจึงเป็นแหล่งข้อมูลที่ดีของการฝึกแบบจำลองของเรา

ก.การเก็บรวบรวมข้อมูล

เพื่อให้เราเข้าถึงไฟล์เล่นซ้ำเราจำเป็นต้องมี Universal Resource Locator (URL) ของไฟล์แต่ละไฟล์ OpenDota ได้จัดทำ Application Programming Interface (API) สำหรับการเข้าถึง URL ไฟล์เล่นซ้ำ DOTA2 ที่อนุญาตให้เราดาวน์โหลดการแข่งขันจากเซิร์ฟเวอร์ของวาล์ว เราได้รวบรวมการแข่งขันระดับมืออาชีพ 5000 (ทัวร์นาเมนต์ใหญ่) และ การแข่งขันระดับมือถึงมืออาชีพ 5000 (ทัวร์นาเมนต์รองและลีก) ที่เล่นก่อนวันที่ 5 ธันวาคม 2561 บัญชีสำหรับข้อผิดพลาดในการดาวน์โหลดข้อมูลเกมมีทั้งหมด 9883 การแข่งขันในการดึงข้อมูลจากไบนารีไฟล์เราได้ใช้ตัวแยกโอเพนซอร์ส parser Clarity Clarity ได้ให้ฟังก์ชันเพื่อตรวจสอบลำดับวัตถุในเกมและเข้าถึงคุณลักษณะของวัตถุในเกม

เมื่อเปิดไฟล์เล่นซ้ำให้เป็นอนุกรมเวลาเราได้บันทึกค่าของชุดของคุณลักษณะ(คุณในหัวข้อถัดไป)ด้วยตัวอย่างที่กำหนดจุดข้อมูลที่อยู่ใกล้กันมีความสัมพันธ์กันสูง ดังนั้นการลดระยะตัวอย่างที่กำหนดส่งผลให้ข้อมูลมากขึ้นและมีความหลากหลายน้อยลง เพื่อให้ได้ประโยชน์สูงสุดที่สุดจากไฟล์เล่นซ้ำขณะที่รักษารายละเอียดของชุดข้อมูลที่จัดการได้เราเลือกตัวอย่างการสุ่มให้เป็น 4 เกมซึ่งตรงกับ 0.133 วินาทีของในเกม ในการคำนวณเวลาที่แน่นอนของการตายเราใช้ข้อมูลความละเอียดสูงซึ่งส่งผลให้การจำแนกประเภทมีความแม่นยำมากขึ้น มีค่าคุณลักษณะสำคัญหายไปจำนวนหนึ่งในไฟล์เล่นซ้ำตัวอย่างเช่น คุณลักษณะเกี่ยวกับสถิติฮีโร่ หลังจากยกเลิกไฟล์พวกนี้ไปแล้วเราจะยังคงเหลือไฟล์เล่นซ้ำอยู่ 7311 ไฟล์ เท่าที่เราทราบไม่มี bias ที่แนะนำให้ทั้งไฟล์พวกนี้

ข.การประมวลผลข้อมูล

ขั้นตอนการประมวลผลข้อมูลรวมถึงการทำความสะอาดข้อมูล สร้างคุณลักษณะเพิ่มเติม calculating labels และ normalization

เพื่อให้ข้อมูลอนุกรมเวลาตรงกันในเกมเราได้ทำการลบ ticks ทั้งหมดที่เกิดขึ้นระหว่างการหยุดในเกม ในเกม DOTA2 ผู้เล่นทุกคนสามารถหยุดเกมชั่วคราวได้ซึ่งจะเกิดขึ้นเมื่อมีปัญหาทางเทคนิคใด ๆ ในเกมเกิดขึ้นทำให้ผู้ตัดสินมีโอกาสแก้ไขปัญหาในเกมได้ เนื่องจากการทำนายความตายในระหว่างเกมหยุดชั่วคราวไม่สำคัญจุดของข้อมูลหยุดใด ๆ จึงถูกลบออกจากชุดข้อมูล

เพราะธรรมชาติของการทำนาย เราพยายามที่จะทำให้บรรลุผลเช่น การทำนายความตาย เราจำเป็นต้องสร้างชุดคุณลักษณะที่จะรวบรวมข้อมูลที่เกี่ยวข้อง เราเริ่มต้นด้วยการดูสองปัจจัยที่สำคัญที่สุดที่ทำให้ผู้เล่นถูกสังหาร ความแข็งแกร่งของฮีโร่และตำแหน่งปัจจุบันของพวกมันบนแผนที่ในเวลาที่กำหนด การสร้างจากคุณลักษณะน้อย ๆ เพื่อเป็นตัวแทนจนจบลงด้วยชุดคุณลักษณะที่มี 287 คุณลักษณะต่อฮีโร่ คุณลักษณะบางอย่างเป็นค่าของคุณลักษณะวัตถุในเกม เช่นพลังชีวิตอื่น ๆ ได้มาจากค่าที่ใกล้เคียงกับผู้เล่นอื่น ภาพรวมด้านล่างนี้สำหรับคุณลักษณะแต่ละประเภทและทำไมถึงรวบรวม รายละเอียดเพิ่มเติมอยู่ในตารางที่ 1 เนื่องจากจำนวนของคุณลักษณะและข้อจำกัดของหน้า ไม่สามารถให้รายละเอียดทั้งหมดของแต่ละคุณสมบัติแต่สามารถขอได้โดยติดต่อผู้เขียนคนแรก

- เวลา (1 คุณลักษณะ) เวลาของเกมอาจเป็นคุณลักษณะที่มีประโยชน์ด้วยเหตุผลสองข้อ หนึ่งคือมีเหตุการณ์เกิดขึ้นตามเวลาที่กำหนด (ครีปเกิด) สองเวลาผ่านไปโดยทั่วไปผู้เล่นจะแข็งแกร่งมากขึ้นส่งผลให้ผู้เล่นทำตามวัตถุประสงค์แตกต่างกันไป การเปลี่ยนแปลงการเล่นเกมนี้จะเรียกว่า ดันเกม กลางเกมและท้ายเกม

- สถานะปัจจุบันของฮีโร่ (21 คุณลักษณะ) รวมถึงคุณลักษณะพลังชีวิต ความแข็งแกร่ง ความคล่องตัว อื่น ๆ รายการทั้งหมดอยู่ในตารางที่ 1

- สถิติที่เกี่ยวกับฮีโร่ (17 คุณลักษณะ) อธิบายถึงสิ่งที่ฮีโร่ได้ทำมาแล้ว ยกตัวอย่างเช่น จำนวนการสังหาร จำนวนการตาย จำนวนสังหารครีป ค่าประสบการณ์ อื่น ๆ รายการทั้งหมดอยู่ในตารางที่ 1

- ไอเท็มที่ใช้งานได้ (34 คุณลักษณะ) ผู้เล่นสามารถซื้อไอเท็มที่สามารถกดใช้งานได้เพื่อให้ความสามารถของไอเท็มได้ทำงานและมีผลอย่างมากต่อการปะทะกัน ยกตัวอย่างไอเท็มสามารถทำให้ผู้เล่นเคลื่อนย้ายออกได้จากจุดอันตราย แต่ละไอเท็มมีค่าคุณลักษณะ 2 ค่าแทนว่าผู้เล่นมีไอเท็มหรือไม่และคุณลักษณะคลูดาวน์ซึ่งแสดงเวลาจนกระทั่งใช้งานได้อีกครั้ง เพื่อให้การใช้พื้นที่คุณลักษณะน้อยที่สุดจึงได้เลือก 17 คุณลักษณะที่มีผลมากที่สุดโดยเลือกตามความคิดเห็นของผู้เชี่ยวชาญ รายการทั้งหมดอยู่ในตารางที่ 1

- ความสามารถของฮีโร่ (48 คุณลักษณะ) รวมถึงคุณลักษณะความสามารถของฮีโร่ เช่น คลูดาวน์ เลเวล มานาที่ต้องใช้ อื่น ๆ (รายการทั้งหมดอยู่ในตารางที่ 1) สำหรับฮีโร่ที่ความสามารถน้อยกว่า 8 ค่าจะเป็น 0

- ฮีโร่ ID (130 คุณลักษณะ) ข้อมูลประจำตัวของฮีโร่แทนด้วยเวกเตอร์ที่ใช้ one hot encoded คุณลักษณะนี้ให้ข้อมูลที่เป็นประโยชน์เนื่องจากฮีโร่แต่ละตัวมีจุดแข็งจุดอ่อนแตกต่างกันไปอย่างไรก็ตามด้วยจำนวนฮีโร่มีมากจึงเป็นส่วนที่สำคัญของคุณลักษณะนี้

- ตำแหน่งของฮีโร่ (4 คุณลักษณะ) การเปลี่ยนตำแหน่ง (4 คุณลักษณะ) ตำแหน่งจะแสดงเป็นพิกัด x และ y

- ความใกล้เคียงของฮีโร่ การเปลี่ยนแปลงความใกล้เคียง (18 คุณลักษณะ) แทนระยะห่างระหว่างผู้เล่นปัจจุบันกับผู้เล่นคนอื่นทุกคนและอัตราการเปลี่ยนแปลงของระยะห่าง

- ระยะความใกล้ของศัตรูที่ยังมีชีวิตอยู่และป้อมของฝั่งเราและการเปลี่ยนแปลงระยะห่าง (4 คุณลักษณะ) ป้อมสามารถสร้างความเสียหายได้เป็นจำนวนมาก

- ประสิทธิภาพมองเห็น (10 คุณลักษณะ) 2 คุณลักษณะแทนด้วยผู้เล่นมองเห็นทีมศัตรูในช่วง 10 วินาทีอย่างละเอียดและ 1 วินาทีอย่างละเอียด

Hero statistics	Hero state	Positions/Proximities	Ability	Item list
FristBloodClaimed	Agility	Hero position x	Level	Blink dagger
TeamFightParticipation	AgilityTotal	Hero position x change	CastRange	Black king bar
Level	Intellect	Hero position y	ManaCost	Magic wand
Kills	IntellectTotal	Hero position y change	Cooldown	Quelling blade
Deaths	Strength	Ally proximity 1-4	Activated	Power treads
Assists	StrengthTotal	Ally proximity 1-4 change	ToggleState	Hand of midas
ObserverWardPlaced	MagicalResistanceValue	Enemy proximity 1-5		Hurricane pike
SentryWardPlaced	PhysicalArmorValue	Enemy proximity 1-5 change		Force Staff
CreepsStacked	Mana	Closet ally tower proximity		Abyssal blade
CampsStacked	MaxMana	Closet ally tower proximity change		Mask of madness
RunePickups	TauntCooldown	Closet enemy tower proximity		Nullifier
TowerKills	BKBChargesUsed	Closet enemy tower proximity change		Travel boots
RoshanKills	AbilityPoints			Dagon 5
TotalEarnedGold	PrimaryAttribue			Lotus orb
LastHitCount	MoveSpeed			Tpscroll
TotalEarnedXP	Health			Smoke of deceit
Stuns	MaxHealth			Clarity
	DamageMax			
	DamageMin			
	lifeState			
	TaggedAsVisibleByTeam			

ตารางที่ 1 รายละเอียดความแตกต่างชนิดของคุณลักษณะ

เกมนี้สามารถสังเกตการณ์ได้อย่างเต็มที่ ไฟล์การเล่นซ้ำจะมีสถานะทุกอย่างในเกม อย่างแรกการเพิ่มคุณลักษณะขึ้นอยู่กับการเล่นอาจจะดูไม่จำเป็นแล้ว อย่างไรก็ตามส่วนที่สำคัญของระบบก็คือผู้เล่นและพฤติกรรมของผู้เล่นขึ้นอยู่กับเกมที่ผ่านมา ยกตัวอย่าง หากศัตรูเพิ่งหายไปจากสายตาของผู้เล่นผู้เล่นยังสามารถรู้ได้ว่าศัตรูยังอยู่ในพื้นที่นั้น ในทางกลับกันหากศัตรูหายไปเมื่อไม่กี่นาทีก่อนศัตรูสามารถอยู่ที่ไหนก็ได้ในแผนที่จากมุมมองของผู้เล่น จึงเป็นสาเหตุที่เพิ่มคุณลักษณะประวัติการมองเห็น

แม้ว่าข้อมูลจำนวนมากที่มีอยู่ในคุณลักษณะแต่ก็ยังขาดข้อมูลที่สำคัญไปบนมนุษย์ที่เป็นผู้สังเกตการณ์ได้ให้ความสนใจ ยกตัวอย่างเช่น การไม่มีข้อมูลเกี่ยวกับ projectiles หรือ ผู้เล่นที่ไม่ใช่มนุษย์

การจำแนกประเภทเป็นเลขฐานสองแสดงให้เห็นว่าผู้เล่นจะเล่นถูกสังหารภายใน 5 วินาที

การNormalizing คุณลักษณะผ่านตัวผู้เล่น สำหรับแต่ละคุณลักษณะเราได้คำนวณค่าสูงสุดและต่ำสุดของผู้เล่นคนใดที่การแข่งขันใด ๆ และใช้ค่าพวกนี้ปรับขนาดของคุณลักษณะให้อยู่ในช่วง 0 ถึง 1

แบบจำลอง

เราได้ใช้ Deep feedforward neural network ด้วย weight sharing ของแบบจำลองเรา อินพุตของ neural network ประกอบด้วยคุณลักษณะ(ที่กล่าวไว้ในการประมวลผลข้อมูล)สำหรับแต่ละอีโรว์ ในชุดคุณลักษณะที่ใหญ่ที่สุดอีโรว์มีคุณลักษณะ 287 คุณลักษณะ และมี 10 อีโรว์ดังนั้นเครือข่ายมีอินพุตทั้งหมด 2870 เครือข่ายมีเอาต์พุตที่ต่อเนื่องอิสระ 10 ตัว ความน่าจะเป็นของแต่ละผู้เล่นที่กำลังจะถูกสังหารในอีก 5 วินาที ในส่วนนี้เราจะพูดถึง weight sharing และ สถาปัตยกรรมของระบบเครือข่าย (network architecture) เราจะนำเสนอกระบวนการวิธีของเราสำหรับการปรับสมดุลข้อมูล การฝึกข้อมูล และการสำรวจไฮเพอร์พารามิเตอร์

ก.Weight sharing

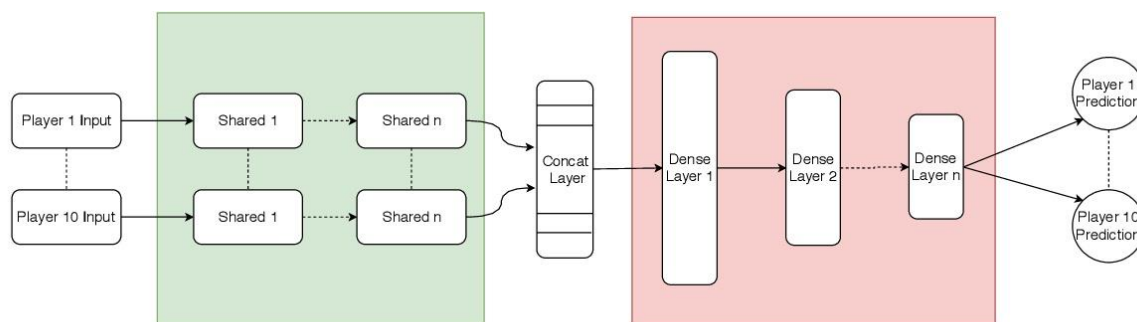
Weight sharing สามารถลดจำนวน weight ของเครือข่ายได้อย่างมากโดยไม่ต้องเสียกำลังมากมาย สิ่งนี้สามารถทำได้โดยใช้ความสมมาตรโดยธรรมชาติของขอบเขต ในกรณีของเรานั้นสมมาตรมาจากอีโรว์ทุกตัวที่มีคุณลักษณะเหมือนกันและลำดับของอีโรว์ไม่เกี่ยวข้องกับการเล่นเกม Weight sharing มักถูกนำไปใช้กับขอบเขตต่าง ๆ จากภาพและเสียง การจำแนกประเภทไปจนถึงการประมวลผลภาษาธรรมชาติ(Natural language processing) หนึ่งในตัวอย่างที่ประสบความสำเร็จและรู้จักมากที่สุดของ Weight sharing คือ convolution neural network (CNN) ในกรณีของการตรวจสอบและจดจำรูปภาพ (image recognition) เป้าหมายคือการเรียนรู้ตำแหน่งคงที่และไม่เปลี่ยนแปลง (invariant representations) เนื่องจากเราต้องรู้จักวัตถุที่อยู่ที่ไหนก็ได้ของภาพ ในการทำนายปัญหาที่กล่าวถึงในวิจัยนี้เราต้องการเรียนรู้ช่องของอีโรว์คงที่และไม่เปลี่ยน

ข.สถาปัตยกรรมของเครือข่าย

ภาพที่ 1 ได้แสดงภาพรวมของโครงสร้างเครือข่ายแม้ว่าจำนวนชั้นและความลึกของเครือข่ายจะไม่แสดงให้เห็นแต่ค่าพวกนี้สามารถดูได้ในตารางที่2 ซึ่งยังแสดงไฮเพอร์พารามิเตอร์ของเครือข่ายในรูปแบบที่ 1 เห็นได้ว่าอินพุตเข้าสู่โครงสร้างเครือข่ายที่ใช้ร่วมกันได้อย่างไร

ทั้งหมดนี้คือ weights shared โดยพื้นฐานแล้วการสร้างเครือข่ายย่อยที่เรียนรู้ใหม่ ที่หนาแน่นขึ้นสำหรับคุณลักษณะอีโรว์ Concatenation layer จะต่อเติมเอาต์พุตจากแต่ละ layer ที่ใช้ร่วมกันสร้าง tensor ของอินพุตไปยังส่วนสุดท้ายของโครงสร้างเครือข่ายที่เชื่อมต่ออย่างสมบูรณ์

Rectified linear unit (RELU) [24]ถูกใช้สำหรับเอาต์พุตของทุก hidden layer ในขณะที่เราใช้ Sigmoid function สำหรับการเปิดใช้งานเครือข่าย



รูปภาพที่ 1 ภาพรวมของสถาปัตยกรรมของระบบเครือข่าย คุณลักษณะจากแต่ละฮีโร่ผ่าน shared layer ซึ่งทำการคำนวณเหมือนกันสำหรับฮีโร่แต่ละตัว ตัวอย่างเหล่านี้เชื่อมต่อกับเวกเตอร์ขนาดใหญ่อันหนึ่งซึ่งถูกส่งไปผ่านยัง fully connected network เครือข่ายมีเอาต์พุต 10 ตัว ซึ่งแต่ละตัวจะสอดคล้องกับความน่าจะเป็นของฮีโร่แต่ละตัวที่จะตายภายใน 5 วินาที

ค.การจัดการข้อมูลที่ไม่สมดุล

ชุดข้อมูลเดิมในชุดต่อเนื่องไม่สมดุลอย่างมาก เนื่องจากเวลาส่วนใหญ่ในระหว่างการแข่งขันผู้เล่นจะไม่ค่อยตายมีจำนวนคะแนนมากขึ้นสำหรับการฝึกที่ “ไม่ตาย” กลุ่ม โดยมีเพียงร้อยละ 1 ของข้อมูลการจำแนกประเภทเป็นกลุ่ม positive “ตาย” เพื่อความสมดุลของการฝึกข้อมูลเราใช้ประโยชน์จากการสุ่มตัวอย่างบนกลุ่ม negative กระบวนการนี้แบ่งออกเป็นสองขั้นตอน

หลังจากวิเคราะห์ไฟล์เล่นซ้ำอย่างต่อเนื่องที่สามารถอ่านได้เราจะสุ่มทั้งร้อยละ 50 ของกลุ่ม negative จากชุดข้อมูล (ซึ่งไม่มีผู้เล่นถูกสังหารอย่างน้อย 5 วินาที) สิ่งนี้มันทำให้เรามีพื้นที่เก็บข้อมูลจำนวนมากและทำการโหลดชุดข้อมูลเร็วขึ้น

ขั้นตอนที่สองของการจัดสมดุลข้อมูลเกิดขึ้นเมื่อไหร่ที่เราประกอบ minibatch สำหรับ gradient descent เนื่องจากมีบางคนตายคนอื่นก็ไม่ว่าเป็นที่จะต้องตายในเวลาเดียวกันมันเป็นไปได้ที่จะได้ batches สมดุลสำหรับผู้เล่นแต่ละคนแต่เราปรับสมดุล batch สำหรับผู้เล่นเพียงคนเดียว ในการทำซ้ำแต่ละครั้งผู้เล่นจะถูกเลือกแบบสุ่มและชุดข้อมูลถูกสุ่มตัวอย่างในจำนวนของกลุ่ม positive และ negative มีค่าเท่ากับสำหรับผู้เล่นจากนั้นเราจะใช้ backpropagated ของค่าผิดพลาดสำหรับผู้เล่นที่ถูกเลือก

ง.ชุดของคุณลักษณะ

เนื่องจากว่าความซับซ้อนของเกมเราจึงไม่แน่ใจว่าเครือข่ายสามารถใช้ประโยชน์ได้ทุกคุณลักษณะหรือไม่ เพื่อให้ได้ประโยชน์มากขึ้นเราได้สร้างชุดคุณลักษณะแยกไว้สามชุดและเรียกใช้ฝึกด้วยการค้นหาไฮเพอร์พารามิเตอร์แบบแยกสามประเภท

•ชุดคุณลักษณะต่ำ (15คุณลักษณะต่อฮีโร่) มีเพียงแค่ คุณลักษณะพลังชีวิตปัจจุบัน เงินทั้งหมด ตำแหน่ง และฮีโร่และป้อมที่ใกล้เคียง เราได้เลือกคุณลักษณะเหล่านี้เพราะเราคิดว่ามันมีข้อมูลมากที่สุด เงินทั้งหมดอธิบายถึงความแข็งแกร่งของฮีโร่ ในขณะที่ความใกล้เคียงอธิบายถึงมีใครที่อยู่ในระยะโจมตีหรือช่วยเหลือฮีโร่ได้

•ชุดคุณลักษณะกลาง (109คุณลักษณะต่ออีโรว์) มีทุกอย่างยกเว้นคุณลักษณะอีโรว์ ID และความสามารถ คุณลักษณะทั้งสองนี้ใช้พื้นที่มากกว่าร้อยละ60ของพื้นที่คุณลักษณะ ในขณะที่พลังความสามารถสามารถเปลี่ยนกระแสน้ำของการปะทะ การใช้ประโยชน์จากคุณลักษณะเหล่านี้แตกต่างกันเนื่องจากมีความสามารถมากกว่าร้อยละความสามารถในเกม

•ชุดคุณลักษณะใหญ่มีคุณลักษณะทั้งหมด (287คุณลักษณะต่ออีโรว์)

จ.การฝึกเครือข่าย

ชุดข้อมูลถูกแบ่งออกก่อนการฝึกเป็นร้อยละ80ต่อร้อยละ10แบ่งเป็นการฝึกและตรวจสอบความสมบูรณ์ อีกร้อยละ10สำหรับการทดสอบ ชุดฝึกประกอบด้วย 5848 การแข่งขันและชุดตรวจสอบความสมบูรณ์ประกอบด้วย 732 การแข่งขัน หลังจากใช้การปรับสมดุลของกลุ่ม ชุดฝึกมีจุดข้อมูล 57.6ล้านจุดในขณะที่ชุดตรวจสอบความสมบูรณ์ประกอบด้วยจุดข้อมูล 7.2ล้านจุด ร้อยละ9.2ของกลุ่มที่เป็นpositive

เครือข่ายที่ฝึกมาแล้วได้ประเมินในชุดทดสอบที่มี 731 การแข่งขัน ไม่มีการปรับสมดุลของกลุ่มที่ใช้กับชุดทดสอบ เช่น ข้อมูลยังคงตรงกับจุดข้อมูลอย่างต่อเนื่องซึ่งแสดงให้เห็นว่ากระจายข้อมูลอย่างแท้จริง ข้อมูลการทดสอบประกอบด้วยจุดข้อมูล 14 ล้านจุด คิดเป็นร้อยละ1.04ของ กลุ่มที่เป็นpositive

เนื่องจากข้อมูลมีจำนวนมากจึงไม่สามารถโหลดข้อมูลทั้งหมดไว้ในหน่วยความจำได้ ดังนั้นข้อมูลจึงถูกบันทึกลงในไฟล์ขนาดเล็กซึ่งแต่ละไฟล์มีจุดข้อมูล 4000 จุด ส่งผลให้มีไฟล์ประมาณ 16200ไฟล์ สำหรับชุดข้อมูลทั้งหมด สำหรับแต่ละการฝึกทำซ้ำมีชุดข้อมูล 128 จุดข้อมูลจากไฟล์สุ่ม

สำหรับแต่ละชุดคุณลักษณะเราใช้การค้นหาแบบสุ่ม การสำรวจเป็นหลักเพื่อกำหนดพารามิเตอร์ที่ดีที่สุดสำหรับใช้งานต่อไปนี้ จำนวนเลเยอร์ จำนวน neurons ต่อเลเยอร์ อัตราการเรียนรู้และขนาด batch คำอธิบายโดยละเอียดของไฮเพอร์พารามิเตอร์สามารถดูได้ที่ตารางที่ 2

Hyperparameter	Minimal Feature	Medium Feature	All Feature
Learning rate	3.06e-5	Hero position x	Level
Shared layers	[200,100,60,20]	[256,184,64]	[256,128,64]
Final layers	[150,75]	[1024,512,256,128,64,32]	[1024,512,256,128,64,32]
Batch Size	128	128	128
Optimizer	Adam	Adam	Adam

ตารางที่ 2 ไฮเพอร์พารามิเตอร์สำหรับแต่ละชุดคุณลักษณะ สถาปัตยกรรมถูกกำหนดโดยรายการที่ความยาวของรายการคือจำนวนชั้นของเลเยอร์และค่าของจำนวน neurons ในแต่ละเลเยอร์ ดูได้จากภาพที่ 1 สำหรับคำอธิบายของ shared layers และ final layers

บทที่ 4

ผลการทดลอง

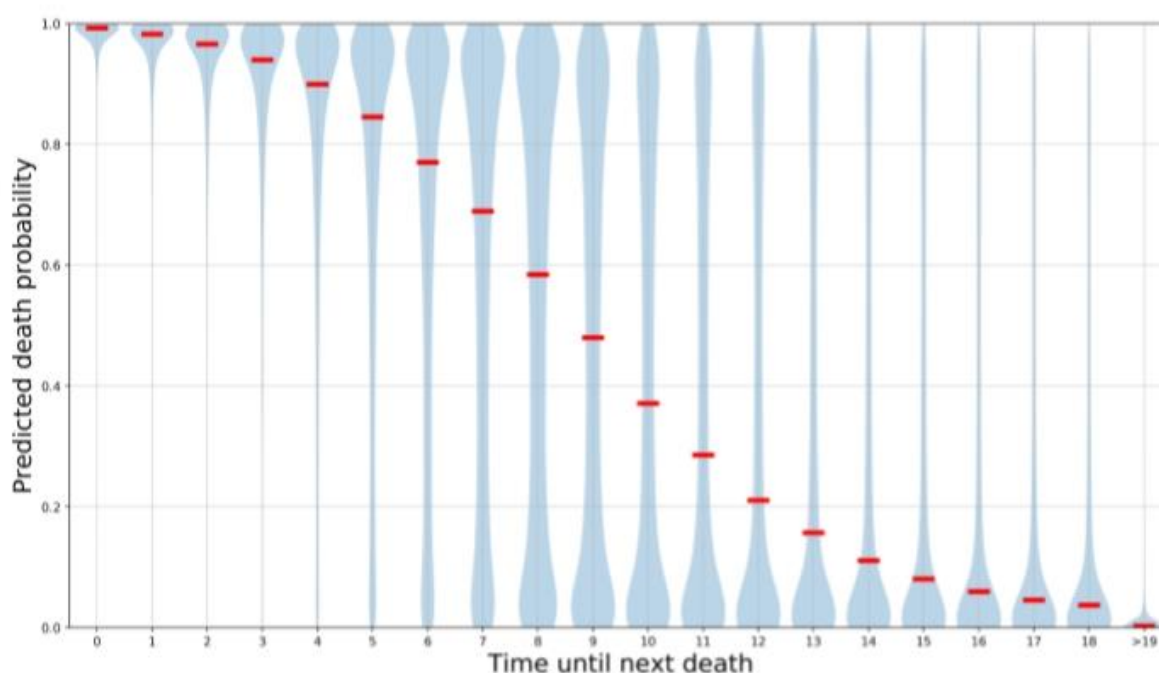
4.1 การประเมินผล

เนื่องจากการสังหารเกิดขึ้นน้อยครั้งในการแข่งขัน DOTA2 อัตราส่วนของกลุ่ม positive ในข้อมูลทดสอบมีประมาณร้อยละ 1 ในข้อมูลที่ไม่สมดุล ความแม่นยำ (accuracy) ไม่ใช่ตัวชี้วัดประสิทธิภาพที่ดี แต่เราใช้เส้นโค้ง precision-recall และค่าเฉลี่ยของ precision ซึ่งเป็นพื้นที่ใต้เส้นโค้ง precision-recall

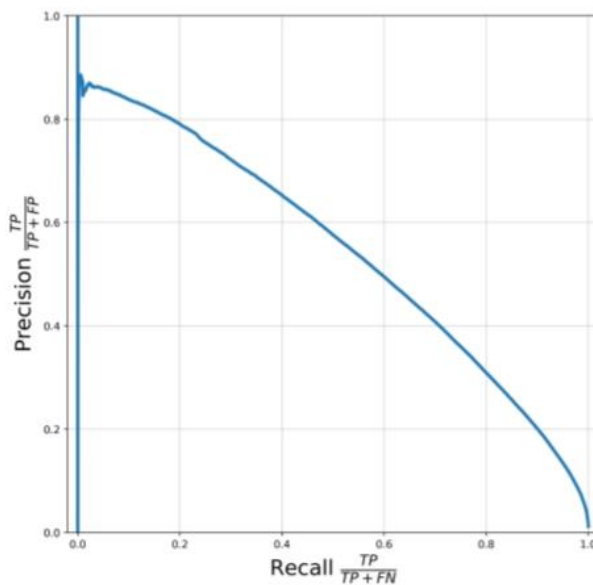
Feature Set	Average Precision
Minimal	0.5001
Medium	0.5365
All	0.5447

ตารางที่ 3 ค่าเฉลี่ย Precision (พื้นที่ใต้เส้นโค้ง Precision-Recall) สำหรับชุดคุณลักษณะของตัวแบบจำลองที่แม่นยำที่สุด

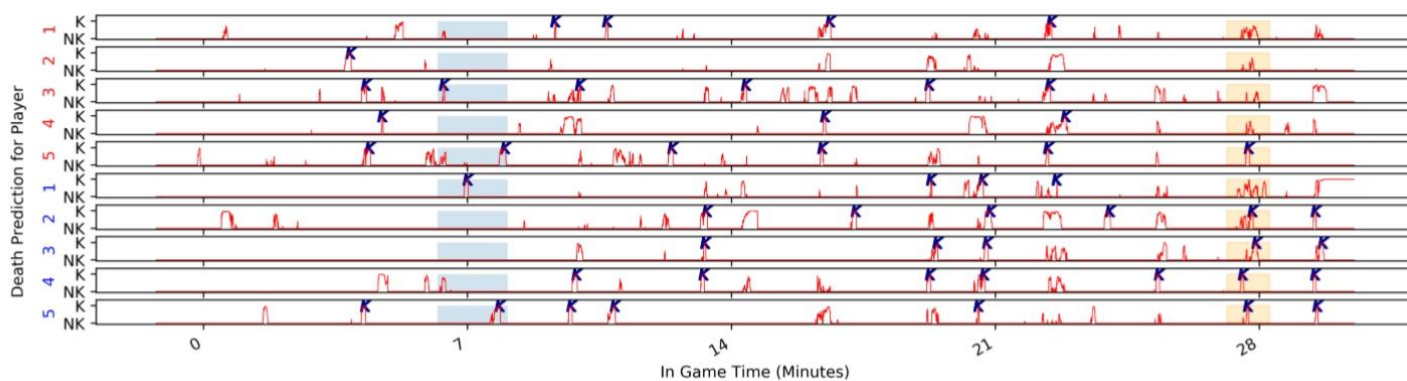
จากตารางที่ 3 แสดงให้เห็นประสิทธิภาพของแบบจำลองที่ดีที่สุดที่ทดสอบสำหรับแต่ละชุดคุณลักษณะ ผลการวิจัยพบว่าเครือข่ายสามารถใช้ประโยชน์จากคุณลักษณะได้ ข้อดีประสิทธิภาพของชุดคุณลักษณะกลางที่มีน้อยมาก ข้อดีของชุดคุณลักษณะใหญ่มีมากกว่าคุณลักษณะกลาง



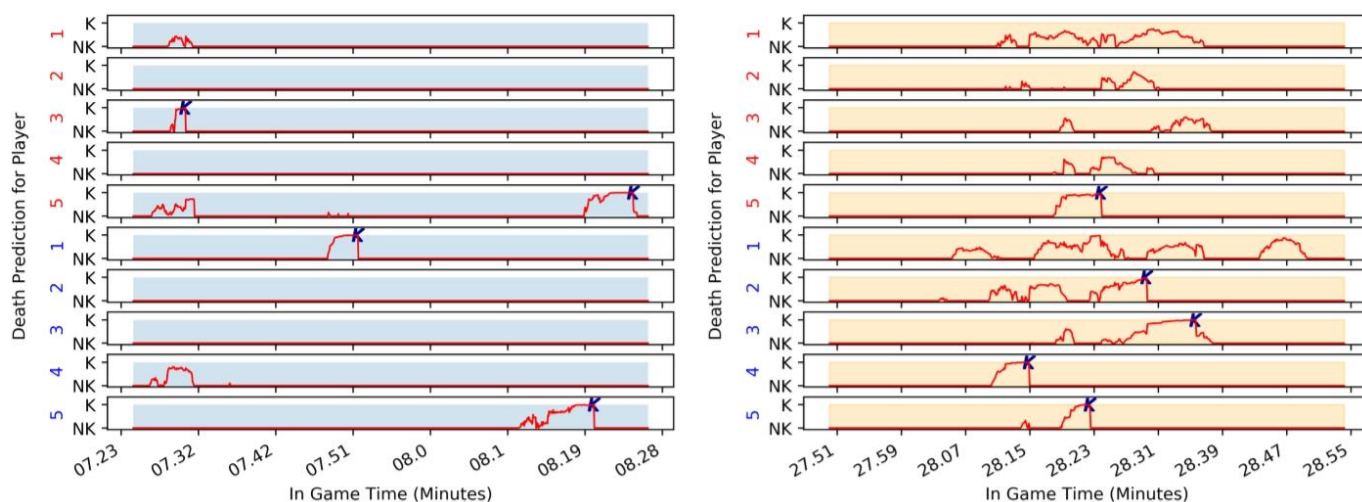
รูปภาพที่ 2 การกระจายของการทำนายสำหรับจุดข้อมูลด้วยเวลาที่แตกต่างกันจนกระทั่งผู้เล่นตายในชุดข้อมูลทดสอบ เส้นสีแดงแสดงถึงค่ามัธยฐาน เครือข่ายได้รับการฝึกฝนเพื่อทำนายว่าผู้เล่นจะตายภายใน 5 วินาทีหรือไม่



รูปภาพที่ 3 เส้นโค้งของ Precision-Recall สำหรับตัวแบบจำลองที่แม่นยำที่สุด ดำเนินการกับชุดข้อมูลทดสอบทั้งหมด



รูปภาพที่ 4 แสดงผลลัพธ์ของตัวอย่างเกมแบบสุ่ม (ID 2857921062) จากชุดทดสอบเรียกใช้แบบจำลองการทำนายของเรา โดยตั้งค่า threshold ไว้ที่ 0.5 ภาพได้บ่งบอกความแน่นอนของการสังหารซึ่งสัญลักษณ์ที่ดูกว้าง เช่น 'K' กับความน่าจะเป็น เส้นสีแดงแสดงถึงความแน่นอนของการตายสำหรับผู้เล่นแต่ละคน โดยที่ที่มีคำอธิบายประกอบเป็นสีแดงและสีน้ำเงิน



รูปภาพที่ 5 และ 6 แสดงให้เห็นถึงส่วนที่แรเงาของรูปภาพที่ 4

แบบจำลองที่มีประสิทธิภาพดีที่สุด มีค่าเฉลี่ย precision อยู่ที่ 0.5447 เราใช้แบบจำลองนี้สำหรับการวิเคราะห์เพิ่มเติม เส้นโค้ง precision-recall แสดงให้เห็นในรูปที่ 3 ในข้อมูลการทดสอบการทำนายความน่าจะเป็นที่มีไม่มาก ได้บวกที่ผิดพลาด (false positive) ค่าปกติเหล่านี้สร้างเส้นแคบในจุดเริ่มต้นของเส้นโค้ง precision-recall precision-recall ได้แสดงให้เห็นถึงความลำบากระหว่างค่า threshold ที่สูงกับค่า precision ที่สูงและ ค่า recall ที่น้อยหรือเลือกค่า threshold กับค่า precision ที่น้อยและ ค่า recall ที่สูง ในตัวอย่าง ค่าthreshold 0.9 แบบจำลองมีค่า precision 0.377 กับ recall 0.725 โดยการส่งข้อมูลอย่างต่อเนื่องจากการแข่งขันเราสามารถสร้างกราฟของการทำนายคะแนนในการแข่งขันทั้งหมดได้ ตัวอย่างของกราฟคาดคะเนเกมโดยสามารถเห็นได้ในรูปที่ 4(a) ซึ่งเราได้วางสัญลักษณ์ของการสังหาร (K) จากเกมเพื่อทำนายเอาท์พุทของแบบจำลอง สิ่งนี้สามารถทำให้เห็นภาพได้ว่ามี true positives และ false positives เกิดขึ้น

นอกจากนี้ยังเป็นที่น่าสนใจแม้ว่าเราจะทำการทดสอบสำหรับการสังหารก่อน 5 วินาทีในบางครั้งโอกาสสังหารจะยังคงถูกทำนายก่อนเครื่องหมาย 5 วินาทีแม้ว่าจะลดความสับสนในการคำนวณสิ่งเหล่านี้จะยังนับเป็นความผิดพลาดการทำนาย หากใช้เครื่องมือนี้ในเกมที่ถ่ายทอดสดความแม่นยำของการทำนายจะสูงกว่าผลการรายงานนี้ เรื่องนี้จะกล่าวไว้ในบทที่ 6 ภาพรวมทั่วไปของเวลาการทำนายจากทั้งที่ถูกฝึก 0-5 วินาที และ ขยายเป็น 20 วินาทีผลลัพธ์ความน่าจะเป็นคู่ได้ในรูปภาพที่ 2

เพื่อพิสูจน์แนวคิดใด ๆ ที่เกี่ยวข้องพียงพามากในบางความสัมพันธ์คุณลักษณะกับการตาย เราเลือกที่จะวิเคราะห์คุณลักษณะพลังชีวิตของแต่ละผู้เล่นสู่การทำนายจากแบบจำลองเราวิเคราะห์จับข้อมูลทั้งหมดเพื่อดำเนินการคำนวณค่าความน่าจะเป็นที่มีค่าเกี่ยวข้องกัน โดยการกำหนดตัวแปรทั้งสองบนแผนภาพกระจาย (scatter plot) และดำเนินการ Spearman

สัมประสิทธิ์สหสัมพันธ์ (correlation coefficient) ได้ให้ค่า -0.09 กับ $P < 0.001$ อย่างมีประสิทธิภาพ หักล้างปัญหาเกี่ยวกับการเรียนรู้ด้วยข้อมูลของเครือข่ายบนตัวแปรหนึ่งตัวแปร

บทที่ 5

สรุปผลการทดลอง

5.1 การอภิปรายผล

ฟังก์ชันการเล่นซ้ำของ DOTA2 ได้ให้ความสามารถในการเล่นซ้ำได้อย่างแน่นอนในขณะที่การเล่นซ้ำในรูปภาพที่ 4 การแข่งขันพร้อมโดยเครื่องมือการทำนายเราสามารถติดตามการทำนายที่ผิดพลาดจากเครือข่ายได้มากกว่าที่จะเห็นในเชิงสถิติว่าเป็นการจำแนกประเภทการฆ่าที่ผิดพลาด สิ่งที่เราพบบ่งชี้ว่าแม้ว่าเครือข่ายจะแน่ใจว่าผู้เล่นจะตาย แต่ปัจจัยภายนอกมากมายเข้ามาในการเล่น ยกตัวอย่างเช่น ผู้เล่นจะทำการสังหารโดยที่ไม่มีข้อมูลที่สมบูรณ์เกี่ยวกับทีมศัตรู หรือ การคาดคะเนความเสียหายผิดพลาดไป การศึกษาเพิ่มเติมจะต้องมีการดำเนินการเพื่อทำความเข้าใจกับสิ่งที่เกิดขึ้นภายในช่วงนี้อาจจะต้องมีความช่วยเหลือจากผู้เชี่ยวชาญ สิ่งที่น่าแน่นอนในแต่ละครั้งของการทำนายการตายนั้นคือการปะทะกันของฮีโร่ที่มากกว่าสองคนหรือมากกว่านั้น

ในการพิจารณาว่าความมีประโยชน์ของแบบจำลองเช่น เป็นเครื่องมือช่วยนักวิจารณ์ การใช้ตัวชี้วัดที่มีประสิทธิภาพ เช่น precision หรือ recall ไม่น่าเพียงพอต่อวัตถุประสงค์เนื่องจากมีความผิดพลาดชนิดต่าง ๆ ของการทำนายมีผลกระทบต่อประโยชน์ของแบบจำลองเราสามารถจำแนกความผิดพลาดการทำนายได้เป็นสี่ประเภท

- False negative ฮีโร่จะตาย แต่ แบบจำลองผิดพลาดในการทำนาย
- False positives ฮีโร่กำลังจะตาย แต่ไม่ใช่ในวินาที
- False positives ฮีโร่ไม่ได้จะตายแต่ผู้เล่นระดับมืออาชีพประเมินว่าเป็นสถานการณ์อันตราย
- False positives ฮีโร่ที่ไม่ได้จะตายแต่ผู้เล่นระดับมืออาชีพประเมินว่าไม่เป็นสถานการณ์อันตราย

แบบจำลองจำนวนมากของประเภทที่ 1 มีข้อผิดพลาดก็ยังคงสามารถช่วยเหลือนักวิจารณ์แม้ว่าความช่วยเหลือจะให้น้อยลงก็ตาม ในทางกลับกันข้อผิดพลาดการทำนายของประเภทที่ 4 จำนวนมากมีผลเสียต่อประโยชน์ของแบบจำลอง ถ้าแบบจำลองเตือนนักวิจารณ์อย่างต่อเนื่องเพื่อให้เป็นเหตุการณ์ที่น่าสนใจมันจะกลายเป็นอุปสรรคแทนความช่วยเหลือ ข้อผิดพลาดการทำนายประเภทที่ 2 และ 3 ทำให้แบบจำลองมีประโยชน์และเป็นที่ต้องการ การทำนายผิดพลาดประเภทที่ 2 สามารถแจ้งเตือนนักวิจารณ์ถึงบางคนที่จะตายในเวลาข้างหน้า สถานการณ์ของเกมเมื่อมีผู้เล่นหนีออกจากปะทะแทบจะไม่สนุกเหมือนตอนที่ผู้เล่นบางคนตาย ความสามารถในการทำนายสถานการณ์ดังกล่าวทำให้แบบจำลองมีประโยชน์ทำให้การทำนายผิดพลาดประเภทที่ 3 เป็นที่ต้องการ โชคร้ายที่การแยกความแตกต่างของประเภทที่ 3 กับประเภทที่ 4 ต้องใช้ผู้เชี่ยวชาญด้วยเหตุนี้การประเมินประสิทธิภาพจึงแตกต่างกันและต้องทำงานเพิ่มเติม

5.2 การสรุปผล

ในข้างต้นเราได้สรุปงานวิจัยของเราเกี่ยวกับ micropredictions ในเกม DOTA2 การใช้เพอร์เซ็ปตรอนหลายชั้น (multilayered perceptron) การฝึกในคุณลักษณะที่หลากหลาย การใช้ shared weights ภายในครั้งแรกของเครือข่ายความสามารถในการฝึกทำให้แบบจำลองสามารถทำนายสถานการณ์ตายของผู้เล่นทุกคนได้อย่างแม่นยำภายในเกมที่ถ่ายทอดสดหรือเกมที่ถูกบันทึกของ DOTA2 ในขณะที่การจัดการอินพุตแบบสมมาตรเป็นของชุดข้อมูล MOBA เราอธิบายกระบวนการประมวลผลข้อมูล esports ล่วงหน้าจาก DOTA2 และวิธีจำลองการทดลองที่น่าเสนออย่างครบถ้วนรวมถึง

ผลลัพธ์ขนาดใหญ่ของการสำรวจพารามิเตอร์บนเครือข่ายและสถาปัตยกรรมคุณลักษณะการร่วมมือกันนี้จะช่วยการทดลองในอนาคตเป้าหมายคือ micropredictions ไปยัง esports การสนับสนุนของงานวิจัยนี้คือการสำรวจของแบบจำลองการเรียนรู้เชิงลึกที่ใช้กับ DOTA2 ได้ที่ micro level (หรือระดับเหตุการณ์)เมื่อเทียบกับการแข่งขัน เราได้ทำแบบจำลองสำหรับการทำนายการตายของผู้เล่นที่เป็นมืออาชีพด้วยความแม่นยำสูงภายใน 5 วินาที พร้อมตัวอย่างในรูปภาพที่ 4 แม้ว่าเราจะได้ฝึกสำหรับ 5 วินาทีของการทำนายแบบจำลองได้เรียนรู้คุณสมบัติโดยธรรมชาติของคุณลักษณะที่ประกอบไปด้วยการสังหารซึ่งในบางสถานการณ์เครือข่ายสามารถทำนายได้ก่อน 5 วินาที การแบ่งความน่าจะเป็นเหล่านี้สามารถดูได้ในรูปภาพที่ 2 ตัวแบบจำลองสามารถพัฒนาเป็นเครื่องมือสำหรับนักวิจารณ์เกมที่มุ่งเน้นและเล่าเรื่องมากกว่า 5 วินาทีก่อนจะเกิดการสังหารภายในเกม

เอกสารอ้างอิง

- [1] J. Hamari and M. Sj  blom, “What is esports and why do people watch it?,” *Internet research*, vol. 27, no. 2, pp. 211–232, 2017.
- [2] Superdata Research, “European esports conference brief (<http://strivesponsorship.com/wp-content/uploads/2017/04/superdata2017-esports-market-brief.pdf>),” 2017.
- [3] M. Schubert, et al., “Esports analytics through encounter detection,” in *Proceedings of the MIT Sloan Sports Analytics Conference*, 2016.
- [4] F. Block, et al., “Narrative bytes : Data-driven content production in esports,” in : *TVX ’18, Procs of ACM Int’l Conf. on Interactive Experiences for TV and Online Video*, pp. 29–41, ACM, 2018.
- [5] F. Block and A. Drachen, “The case for data in esports,” *Develop*, vol. 185, p. 10, 2017.
- [6] G. Yannakakis, “Game AI Revisited,” in *Proc. of ACM Computing Frontiers Conference*, pp. 285–292, 2012.
- [7] P. Yang, et al., “Identifying patterns in combat that are predictive of success in moba games,” in *Procs of the 9th Int’l Conf. on Foundations of Digital Games, FDG ’14*, ACM, 2014.
- [8] Y. Seo, “Electronic sports: A new marketing landscape of the experience economy,” *Journal of Marketing Management*, vol. 29, no. 2, pp. 1542– 1560, 2013.
- [9] H. Xue, et al., “E-sports management? institutional logics, professional sports,emergingesportsfield,”in2016NorthAmericanSocietyforSport Management Conference, Orlando, FL, June 2016.
- [10] S. Demediuk, et al., “Player retention in league of legends: a study using survival analysis,” in *ACSW ’18 Proceedings of the Australasian Computer Science Week Multiconference*, 2018.
- [11] W. Wang, “Predicting multiplayer online battle arena (moba) game outcome based on hero draft data,” *Master’s thesis*, Masters thesis, Dublin, National College of Ireland, 2016.
- [12] V. Hodge, et al., “Win prediction in esports: Mixed-rank match prediction in multi-player online battle arena games,” *ArXiv e-prints (CS:AI)*, vol. : 1711.06498, Nov. 2017.
- [13] Y. Yang, T. Qin, and Y.-H. Lei, “Real-time eSports Match Result Prediction,” *ArXiv e-prints*, Dec. 2016.
- [14] G. Synnaeve and P. Bessiere, “A bayesian model for plan recognition in rts games applied to starcraft,” in *Seventh Artificial Intelligence and Interactive Digital Entertainment Conference*, 2011.
- [15] E. Bursztein, “I am a legend: Hacking hearthstone using statistical learning methods,” in *2016 IEEE conference on computational intelligence and games (CIG)*, pp. 1–8, IEEE, 2016.
- [16] A. Summerville, M. Cook, and B. Steenhuisen, “Draft-analysis of the ancients: predicting draft picks in dota 2 using machine learning,” in *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2016.

- [17] F. Rioult, et al., “Mining tracks of competitive video games,” AASRI Procedia, vol. 8, pp. 82–87, 2014.
- [18] A. Drachen, et al., “Skill-based differences in spatio-temporal team behavior in defence of the ancients 2 (dota 2),” in Procs of IEEE Games Media Entertainment Conf., 2014.
- [19] L. Gao, et al., “Classifying dota 2 hero characters based on play style and performance,” University of Utah Course on ML, 2013.
- [20] C. Eggert, et al., “Classification of player roles in the team-based multi-player game dota 2,” in Int’l Conf. on Entertainment Computing, pp. 112–125, Springer, 2015.
- [21] Z. Cleghern, et al., “Predicting future states in dota 2 using value-split models of time series attribute data,” in Pros of the 12th Int’l Conf. on Foundations of Digital Games, FDG ’17, ACM, 2017.
- [22] Steam, “Opendota - dota 2 statistics <https://www.opendota.com/>.”
- [23] M. Schrodt, “Clarity parser for dota 2 and csgo replay files <https://github.com/skadistats/clarity>.”
- [24] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. MIT press, 2016.
- [25] Y. LeCun and Y. Bengio, “Convolutional networks for images, speech, and time series,” in The Handbook of Brain Theory and Neural Networks (M. A. Arbib, ed.), pp. 255–258, MIT Press, 1998.
- [26] Y. LeCun, et al., “Gradient-based learning applied to document recognition,” in Proceedings of the IEEE, pp. 2278–2324, 1998.
- [27] H. Lee, et al., “Unsupervised feature learning for audio classification using convolutional deep belief networks,” in Advances in neural information processing systems, pp. 1096–1104, 2009.
- [28] J.-T. Huang, et al., “Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers,” in IEEE Int’l Conf. on Acoustics, Speech and Signal Processing, pp. 7304–7308, 2013.
- [29] N. V. Chawla et al., “Special issue on learning from imbalanced data sets,” ACM Sigkdd Explorations Newsletter, vol. 6, no. 1, pp. 1–6, 2004.
- [30] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” Journal of Machine Learning Research, vol. 13, no. Feb, pp. 281–305, 2012.

ภาคผนวก

Time to Die: Death Prediction in Dota 2 using Deep Learning

Adam Katona*, Ryan Spick*, Victoria J. Hodge*, Simon Demediuk*,
Florian Block[†], Anders Drachen* and James Alfred Walker*, *Senior Member, IEEE*

*Department of Computer Science, University of York, Heslington, York, YO10 5GE, UK

[†]Department of Theatre, Film and Television, University of York, Heslington, York, YO10 5GE, UK

Email: {ak1774, rjs623, victoria.hodge, simon.demediuk, florian.block, anders.drachen, james.walker}@york.ac.uk

Abstract—Esports have become major international sports with hundreds of millions of spectators. Esports games generate massive amounts of telemetry data. Using these to predict the outcome of esports matches has received considerable attention, but micro-predictions, which seek to predict events inside a match, is as yet unknown territory. Micro-predictions are however of perennial interest across esports commentators and audience, because they provide the ability to observe events that might otherwise be missed: esports games are highly complex with fast-moving action where the balance of a game can change in the span of seconds, and where events can happen in multiple areas of the playing field at the same time. Such events can happen rapidly, and it is easy for commentators and viewers alike to miss an event and only observe the following impact of events. In Dota 2, a player hero being killed by the opposing team is a key event of interest to commentators and audience. We present a deep learning network with shared weights which provides accurate death predictions within a five-second window. The network is trained on a vast selection of Dota 2 gameplay features and professional/semi-professional level match dataset. Even though death events are rare within a game (1% of the data), the model achieves 0.377 precision with 0.725 recall on test data when prompted to predict which of any of the 10 players of either team will die within 5 seconds. An example of the system applied to a Dota 2 match is presented. This model enables real-time micro-predictions of kills in Dota 2, one of the most played esports titles in the world, giving commentators and viewers time to move their attention to these key events.

Index Terms—Esports, Dota 2, Deep Learning, Micro Prediction

I. INTRODUCTION

The term esports describes video games that are played competitively and usually watched by large audiences [1]. Superdata [2] predict 330 million esports spectators by 2019. Esports have become an important research field across academia and industry [3] due to the availability of high-dimensional, high-volume data from virtually every match. This has introduced the field of esports analytics: “the process of using esports related data, [...], to find meaningful patterns and trends in said data, and the communication of these

patterns using visualization techniques to assist with decision-making processes” [3]. This definition highlights a fundamental challenge in esports: making the matches comprehensible to the audience. Many esports are complex and fast-paced, making it hard to fully unpack the live action with the naked eye.

Esports analytics has to a significant extent focused on the Multi-player Online Battle Arena (MOBA) genre [3]. As a typical MOBA, Dota 2 by Valve Corp. is a 10 player game with two teams of five players (taking typically between 30-40 minutes [4]). Dota 2 is a very complex game where each player selects one of many unique ‘heroes’ to play. Each player uses their hero’s spells and abilities coupled with items they purchase in-game to gather resources across the 3-lane map. The heroes play different roles where they aim to generate resources via fights against the rival team to progress through hero levels and become more powerful. Each team’s ultimate aim is to destroy the opposing team’s base. There are multiple strategies and winning a game requires coordination within the team and the ability to react to the opposition’s tactics and behavior. The game is real-time with hidden information and has deep strategic gameplay.

During a Dota 2 game, the players must closely monitor their hero’s status, in particular, the likelihood of dying. Thus, a method of predicting deaths within a game of Dota 2 can be beneficial on multiple levels of both professional and amateur game analysis. In current esports analytics approaches, many of the performance metrics (data variables) used are correlated directly to team success rather than necessarily to a player’s likelihood to die. In particular, in Section V, we show that the hero health variable which may seem an obvious indicator of likelihood to die is not correlated to hero deaths. Some heroes have abilities which allow them to heal themselves or their team-mates while heroes can purchase items in-game that allow them to heal or teleport away from danger. Hence, we cannot simply use player health for predicting deaths as the prediction is complex and requires careful investigation. For the performance of players and heroes to be accurately analysed, the historical data needs to be carefully considered and tailored to the specific task. Applying performance metrics without careful selection introduces noise and leads to biased or skewed analyses.

This work was supported by the EPSRC Centre for Doctoral Training in Intelligent Games & Games Intelligence (IGGI) [EP/L015846/1] and the Digital Creativity Labs (digitalcreativity.ac.uk), jointly funded by EPSRC/AHRC/Innovate UK under grant no. EP/M023265/1.

This paper introduces the first step towards the larger challenge of *micro-predictions* in esports. Micro-predictions are granular predictions about what might happen in the near future in the game, as opposed to predicting the outcome of the match itself. The concept is known from other sports, e.g. predicting which football player that will score the next goal. Understanding the likelihood of dying in the near future can be vital in determining what tactics a player should use within post-game analysis. The use of a prediction tool within the professional game broadcast scene would provide far more readability for the audience on why and how players are making mistakes and what tactics they should use individually and as a team, allowing commentators to simplify complex build ups to when a player dies. Live broadcasts would benefit from feeding this information back to the audience, alongside automated camera movement to direct the focus when these predictions occur to a high enough degree of accuracy, making both the commentator's job and the users' viewing experience a far smoother process. Furthermore, post-game data analysis forms a large part of professional sport players' training schedule [5]. Coaches currently have to select (by hand) points of the game that led to deaths or where a team's advantage snowballed. With access to our tool, automated points could be created where the model would predict a player's death, whether they did die or not allowing a further investigation as to why the model assumed they were in danger. We introduce a novel prediction framework using a shared memory neural network (described in Section IV) and analyse a much larger set of in-game features compared to previous esports work.

The experimental design behind this paper was to train a neural network model to predict whether a hero will die in the next 5 seconds. A 5-second window provides a large enough period before a kill occurs, such that commentators would be able to make use of any prediction information the network provides. We ran preliminary experiments exploring larger window sizes from 5 seconds up to 20 seconds, with the precision of the model drastically dropping as the window size increased.

There are certain immediate limitations, being that every data point is required, even that of unseen hero data. With this in mind, the tool would be limited to post-game analyses, or in a developer/broadcaster environment where the required variables can be obtained (an extension of a professional game broadcast for example). Another limitation is that the game is constantly changing, meaning our model might not generalize well to newer game versions.

The structure of the paper is as follows; Related work will be discussed in Section II. Followed by an in-depth analysis of the data set, including the cleaning and feature selection based on domain knowledge in Section III. In Section IV we then explore the neural network model used, from dealing with the imbalance data, to in-depth hyperparameter exploration. We then summarise results of the fully trained model and how it applies to the test data and a selected Dota 2 match in real time in Section V. Section VI discusses the difficulties of performance elevation. We provide

concluding remarks in Section VII. The source code used for creating the dataset and training the model is available at https://github.com/adam-katona/dota2_death_prediction.

II. RELATED WORK

Collectively, the esports industry is generating knowledge at a rapidly increasing pace. However, much of this knowledge is not publicly available due to commercial confidentiality, so establishing the state-of-the-art in esports is challenging. Research into esports has previously been published across a number of different disciplines, these include; AI, analytics, psychology, education, visualization, ethnography, marketing, management and business (e.g. [1], [3], [6]–[9]).

Within esports, “esports analytics” is generally used to denote business intelligence work centred on esports, i.e., the analysis of data such as behavioural telemetry, sales data, etc.. These analyses occur across the range of disciplines and describe a broad area of work. Our work focuses on the analysis of behavioural telemetry data from esports games.

The majority of academic work in this specific area has focused on building machine learning models to predict the outcome of esports matches or other prediction tasks [7], [9]–[15]. Other work includes Schubert et al. [3], who described a method for detecting spatio-temporally bounded team encounters in Dota 2. Summerville et al. [16] used machine learning to predict draft picks in Dota 2. Rioult [17] discussed some of the general applications of mining player tracks from esports games. Drachen et al. [18] used classification for studying the movement patterns of Dota 2 teams across skill levels. Gao et al. [19] classified Dota 2 heroes based on performance metrics. Eggert et al. [20] used logistic regression to classify players into pre-determined roles using performance metrics.

The most closely related work is Cleghern et al. [21] who predicted hero health in Dota 2 using a combination of techniques: an auto-regressive moving average model to predict small changes in health and a statistical estimation model (see [21]) which predicts large changes in health and works in conjunction with logistic and linear regression that predict the sign and magnitude of that change respectively. The authors note a peak accuracy of 77.2% for the ARMA model to predict small changes in health (less than 100 points) five seconds into the future. The statistical estimation model predicts large health changes 10 seconds into the future. However, the model has poor performance in terms of predicting the points of major health changes, although when it does find them, the accuracy is around 80% for the direction and magnitude of the health change. The models of Cleghern et al. [21] only use health data, unlike the present work which uses a large number of features. Furthermore, their dataset comprises just 542 matches, and the level of the matches is not described in the paper, so it is unknown whether the results would work for professional matches and/or amateur levels.

In summary, while much previous work focuses on macro-predictions (predicting the winning team), there has been a small amount of work on micro-level events like encounters and hero health changes [3], [17], [18], [20], [21] but only

one previous work has focused on in-game forecasting with mixed success. None of the existing work has focused on professional/semi-professional levels or predicted hero deaths. Death events are an important part of the game narrative and equally important for broadcasters' commentaries, spectators' understanding of the game state and players' tactics and analyses. From the state-of-the-art, we can further conclude that esports analytics is complex, requires careful consideration of the data and models used and, frequently, no one technique excels.

III. DATASET

Access to large amounts of data is required in order to successfully train a classification model to predict kills. To generate this we use Dota 2 replay files; these are binary files produced by Valve Corp. for every Dota 2 match played.

They contain all low-level game events from the match. They are a comprehensive representation of the matches and are used by Dota 2 engines to recreate entire matches for re-watching and analysis, so are an ideal source of data to train our models. We explain how we obtain these data and describe the format in Section III-A and III-B, respectively.

A. Data collection

To allow us to access the replay files, we need the URL of each file. OpenDota [22] provides an API for accessing Dota 2 replay URLs that then allows us to download matches from Valve's servers. We collected the last 5000 Professional (major tournaments) and 5000 Semi-Professional (minor tournaments and leagues) games that were played prior to the 5th of December 2018. Accounting for errors in downloaded game data, this resulted in a total collection of 9883 matches. To extract the data from the binary files, we used the open source parser Clarity [23]. Clarity provides functions to traverse the game object hierarchy and to access the attributes of game objects.

To turn the replay into a time-series, we recorded values of a set of attributes (see next section for details) with a given sampling period. Neighboring data points are highly correlated, thus lowering the sampling period results in more data with less diversity. To get most out of our replay files while keeping the size of the dataset manageable, we chose the sampling period to be 4 game ticks, which correspond to 0.133 seconds in game time. To calculate the exact time of death we used the full resolution data however, which resulted in more accurate classification labels. There were some missing key feature values within a number of replay files, for example, the attributes about hero statistics. After discarding these files 7311 replay files remained. To the best of our knowledge, there was no bias introduced by discarding these files.

B. Data processing

The data processing stage included cleaning, creating additional features, calculating labels and normalization.

In order to keep the time series data consistent in the game, we removed all ticks that occurred during a pause in the game.

In a Dota2 game, a pause can be initiated by any player, this occurs when technical faults happen within the game, giving referees a chance to fix any issues. Since predicting death during a paused game is trivial, any pause data points were removed from the dataset.

Due to the nature of the predictions we are trying to achieve, i.e. death prediction, we needed to build a feature set that would capture the relevant data. We began by looking at the two most important factors that result in a player's death, relative strength of the heroes and their current location on the map at a given time. Building from an initially small number of features to represent this we ended with a final feature set of 287 features per hero. Some of these features were the values of game object attributes like hero health. Other features were derived values like proximity to other players. Below an overview is given for each category of features, and why they were included. Further details are contained in Table I. Due to the number of features and the page limit, a full description of each feature is not given, but is available on request by contacting the first author.

- Time (1 feature): The game time might be a useful feature for two reasons. One is that there are events which happen on given times (creature spawns), the other is that as time progresses, the players generally get more powerful, which results in the players pursuing different objectives. This gameplay change is often referred to as transitioning from early-game to mid-game and then late-game.
- Current state of the hero (21 features): Includes attributes like health, strength, agility, etc. Full list in Table I.
- Statistics about the hero (17 features): Describes what the hero did in the past. For example the number of kills, deaths, last hits, experience points etc. Full list in Table I.
- Activatable items (34 features): Players can buy items which can be activated to provide a powerful effect. These items can have a drastic effect on the outcome of an encounter, for example, they can make the player teleport away from danger. Each item has a binary feature value representing whether the player has the item or not, and a cooldown feature, which represents the time until it can be activated again. To keep the feature space low, the 17 most powerful items were selected based on expert opinion. A full list is given in Table I.
- Hero abilities (48 features): Include the attributes of hero abilities like cooldown, level, mana cost, etc. (full list in Table I). For heroes with fewer than eight abilities, values are zero padded.
- Hero ID (130 features): The identity of the hero is represented as a one hot encoded vector. This feature provides useful information since different heroes have different strengths and weaknesses. However, since the number of heroes is high, this represents a significant portion of the feature space.
- Hero position, position change (4 features). The position is represented as x and y coordinates.

TABLE I: Details of different categories of features.

Hero statistics	Hero state	Positions / Proximities	Ability	Items list
FirstBloodClaimed	Agility	Hero position x	Level	Blink dagger
TeamFightParticipation	AgilityTotal	Hero position x change	CastRange	Black king bar
Level	Intellect	Hero position y	ManaCost	Magic wand
Kills	IntellectTotal	Hero position y change	Cooldown	Quelling blade
Deaths	Strength	Ally proximity 1-4	Activated	Power treads
Assists	StrengthTotal	Ally proximity 1-4 change	ToggleState	Hand of midas
ObserverWardsPlaced	MagicalResistanceValue	Enemy proximity 1-5		Hurricane pike
SentryWardsPlaced	PhysicalArmorValue	Enemy proximity 1-5 change		Force staff
CreepsStacked	Mana	Closest ally tower proximity		Abyssal blade
CampsStacked	MaxMana	Closest ally tower proximity change		Mask of madness
RunePickups	TauntCooldown	Closest enemy tower proximity		Nullifier
TowerKills	BKBCChargesUsed	Closest enemy tower proximity change		Travel boots
RoshanKills	AbilityPoints			Dagon 5
TotalEarnedGold	PrimaryAttribute			Lotus orb
LastHitCount	MoveSpeed			Tpscroll
TotalEarnedXP	Health			Smoke of deceit
Stuns	MaxHealth			Clarity
	DamageMax			
	DamageMin			
	lifeState			
	TaggedAsVisibleByTeam			

- Hero proximity, proximity change (18 features): Represent the distance between the current player to every other player, and the rate of change of the distances.
- Distance to closest alive enemy and ally tower, and distance change (4 features): Towers are destroyable structures which can deal a large amount of damage.
- Visibility history (10 features): Binary features representing whether the player was visible for the enemy team in the past 10 seconds with 1-second resolution.

The game is fully observable, the replay file contains the whole game state. At first glance adding features depending on the past might seem unnecessary. However an important part of the system are the players, and the behavior of the players does depend on the past. For example, if an enemy just went out of sight, the player still knows that the enemy is in the area. On the other hand, if the enemy disappeared minutes ago, that enemy can be anywhere from the player's perspective. This was our motivation for adding the visibility history features.

Even though a lot of information is present in these features, there is still important information missing which a human observer pays attention to. For example, there is no information about projectiles or non-player creatures.

The classification label is a binary number representing whether or not the player will die within 5 seconds.

Normalizing the features was done across players. For each feature, we calculated the maximum and minimum values that any player had in any match, and used these values to scale the features to the range between zero and one.

IV. THE MODEL

We used a deep, feedforward neural network with weight sharing as our model [24], [25]. The input of the neural network consists of the features (discussed in the III-B) for each hero. In the largest feature set, a hero has 287 features and there are 10 heroes so the whole network has 2870 inputs. The network has 10 independent continuous outputs representing

the probability of each player dying in the next 5 seconds. In this section, we first discuss weight sharing and our network architecture. Then we present our methods for data balancing, the training procedure and the hyperparameter exploration.

A. Weight sharing

Weight sharing can drastically reduce the number of weights of a network without sacrificing much of the representational power [25]. This can be achieved by using the inherent symmetries of the domain. In our case, the symmetry comes from all heroes having exactly the same kind of features, and the hero slot order being irrelevant to the gameplay. Weight sharing has been applied to various domains; from image [26] and audio [27] classification to natural language processing [28]. One of the most successful and best-known example of weight sharing is the convolutional neural network [26]. In the case of image recognition, the goal is to learn positional invariant representations, since we want to recognize objects which can be anywhere in the image. In the prediction problem discussed within this paper, we want to learn hero slot invariant representations.

B. Network architecture

Fig. 1 shows a brief abstract overview of the network structure, though the number of layers and depth of the network were not shown. Instead, these values can be seen within Table II, which also shows the hyperparameters of the network. The main takeaway from the network Fig. 1, visualises how the inputs entered into the shared network structure.

Here all the weights are shared, essentially creating a sub-network that learns a new, denser representation for the hero features. The concatenation layer appends the outputs from each shared layer creating a tensor of inputs to the last part of the network structure, which is a fully-connected network.

Rectified linear unit (RELU) [24] was used for the output of every hidden layer, while we used the Sigmoid function [24] for the final activation of the network.

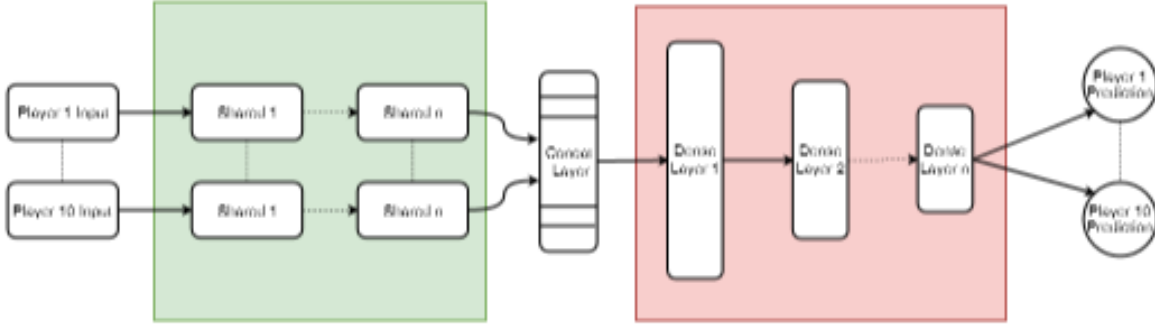


Fig. 1: Overview of the network architecture. The features from each hero flow through the shared layers, which calculate the same representation for each hero. These representations are concatenated to one big vector, which is passed to a standard fully connected network. The network has 10 outputs, each corresponding to the probability of each hero dying within 5 seconds.

C. Dealing with imbalanced data

The original data-set in its continuous form is extremely imbalanced. Since most of the time during a match the players are not about to die there are a significant amount more points for training on "no death" labels, with only around 1 % of the classification data being positive "death" labels. To balance the training data, we utilised under sampling [29] on the negative labels. This process was split into two steps;

After parsing the replays files into their readable continuous format, we randomly discarded around 50% of the negative labels from the data set (where no player died for at least 5 seconds). This saved us a considerable amount of storage space, and made loading the dataset far faster.

The second step of balancing the data took place when we assembled our minibatch for gradient descent [24]. Since when someone dies, others do not necessarily die around the same time it is not possible to get batches where the labels are balanced for each player. Instead, we balanced the batch for only one player. At each iteration, a player was selected randomly and the dataset was sampled in a way that the number of negative and positive labels were equal for this player. Then we only backpropagated the error for the selected player.

D. Feature sets

Because of the complexity of the game, we were not sure if the network can make use of all the feature categories. To gain insight we created three separate feature sets, and run three training procedure with three separate hyperparameter searches.

- The minimal feature set (15 features per hero) only contains the current health, total gold, position, and the hero and tower proximity features. We selected these features because we think these are the most informative. The total gold describes how powerful the hero is, while the proximities describe who is in range to attack or help the hero.
- The medium feature set (109 features per hero) contains everything except for the hero ID and the abilities features. These two feature categories occupy more than 60% of the feature space. While a powerful ability can

change the tide of the battle, making use of these features is difficult, since there are over four hundred unique abilities in the game.

- The large feature set contains all features (287 features per hero).

E. Training the Network

The data set was divided prior to training into an 80%/10% training and validation split, with 10% left over for testing. The training set consisting of 5848 matches, and the validation set consisting of 732 matches. After applying label balancing, the training set contained 57.6 million datapoints, while the validation set contained 7.2 million datapoints. 9.2% of the labels were positive.

The trained network was evaluated on the test set containing 731 matches. There was no label balancing applied to the test set, such that the data remained a continuous match of data points representing the real distribution of data. The test data consisted of 14 million data points, 1.04% percent of the labels were positive.

Due to the sheer amount of data, it was impossible to load the entire data set into memory. Therefore the data was shuffled and saved into small files each containing 4000 data points. This resulted in roughly 16,200 files for the entire data set. For each training iteration, a batch of 128 data points was assembled from a random file.

For each feature set, we used a random search [30] based exploration to determine the best parameters to use for the following; the number of layers, number of neurons per layer, learning rate and batch size. A detailed description of the determined hyperparameters can be seen in Table II.

V. RESULTS

Due to kills occurring rarely within a Dota 2 match, the ratio of positive labels in the test data is around 1%. In such imbalanced data, accuracy is not a good measure of performance. Instead, we used the precision-recall curve, and the average precision, which is the area under the precision-recall curve.

Table III shows the performance of the best model on the test set for each feature set. The results indicate that

TABLE II: Hyperparameters for each feature set. The architecture is given by lists, where the length of the lists are the number of layers, and the values are the number of neurons in each layer. See Fig. 1 for an explanation of shared and final layers.

Hyperparameter	Minimal Features	Medium Features	All Features
Learning rate	3.06 e-5	7.48 e-5	6.15 e-5
Shared layers	[200, 100, 60, 20]	[256, 128, 64]	[256, 128, 64]
Final layers	[150, 75]	[1024, 512, 256, 128, 64, 32]	[1024, 512, 256, 128, 64, 32]
Batch Size	128	128	128
Optimizer	Adam	Adam	Adam

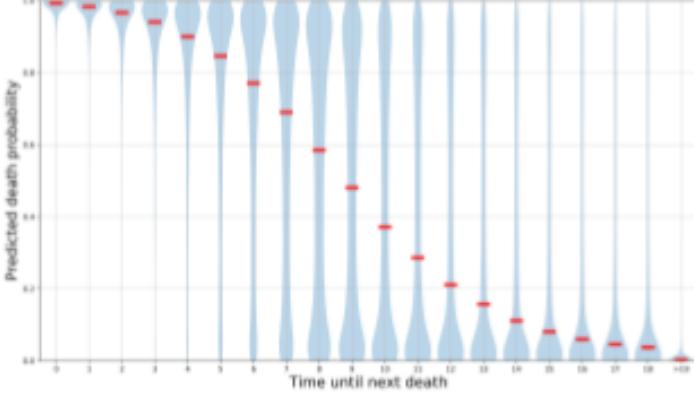


Fig. 2: Distribution of predictions for data points with different time until the player dies on the test data. The red line represents the median. The network is trained to predict whether a player will die within 5 seconds.

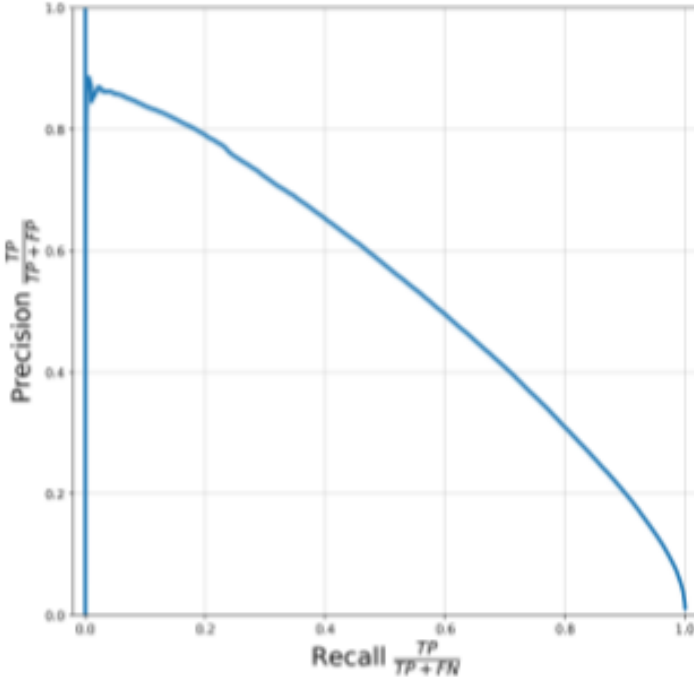


Fig. 3: Precision-Recall curve for the most accurate model, run against the entire test set.

TABLE III: Average precision (Area under the Precision-Recall curve) for feature sets for most accurate model.

Feature Set	Average Precision
Minimal	0.5001
Medium	0.5365
All	0.5447

the network is able to make use of the extra features. The performance advantage of the medium feature set to the minimal one is substantial. The advantage of the largest feature set over the medium one is more subtle.

The best performing model achieved an average precision of 0.5447. We used this model for further analysis. The precision-recall curve of this model is shown in Fig. 3. On the test data, a few of the highest predicted probability were actually false positives. These outliers are creating a narrow drop at the beginning of the precision-recall curve. The precision-recall shows the dilemma between choosing a high threshold with high precision, and low recall, or choosing a low threshold with low precision and high recall. For example, at the threshold of 0.9 the model has 0.377 precision with 0.725 recall. By passing data in a continuous manner from a match we can construct a graph of prediction points on our test matches over the course of the entire match file. An example of a typical game predictions graph can be seen in Fig. 4a, in which we plot the actual labels of kills (K) from a game against the prediction output of the model. This can visualize how many true positives and false positives occur.

It is also worth noting that even though we are testing for kills before 5 seconds, on some occasions kills would still be predicted before the 5-second mark, though to reduce any confusion in the calculations these would still be counted as mispredictions. Therefore if this tool would be used in a live game the accuracy of predictions could be higher than what is reported in this section. This is further discussed in Section VI. A generalised overview of prediction times from both the trained 0-5 seconds and an extended window to 20 seconds, and their probability outputs can be seen in Fig 2

To disprove any notion that the network relied heavily on certain feature correlations with death, we chose to analyse the feature of each player's health relevant to the prediction from the model. We analyzed an entire match to calculate the probability that these values were related. By plotting both of these variables on a scatter plot and conducting a Spearman

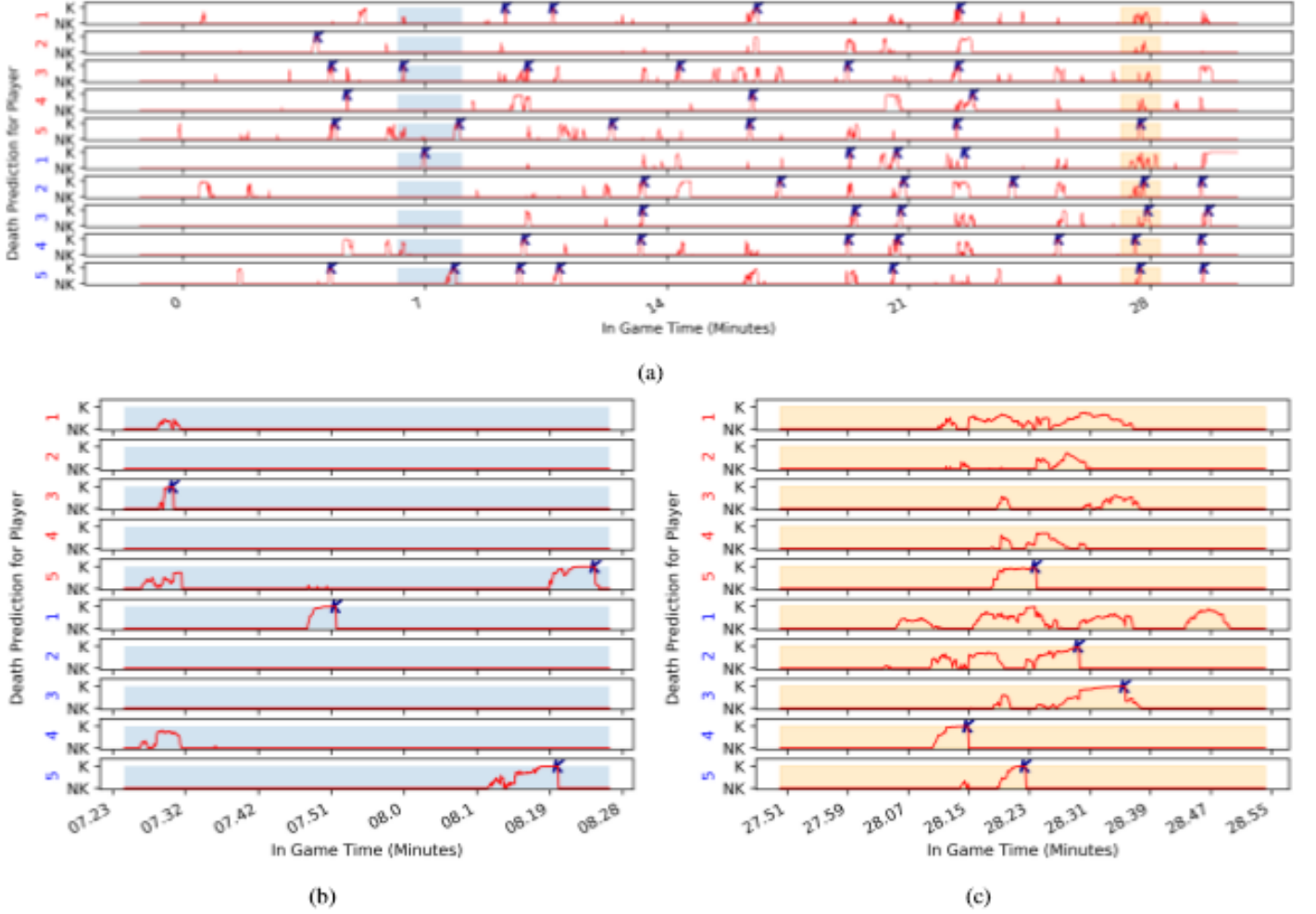


Fig. 4: (a) shows the results of a randomly sampled game (ID 2857921062) from the test set, run through our prediction model with our threshold set to 0.5. Visual indication of certainty of kills which are labelled as ‘K’ with red probability line showing certainty of the death for each individual player, with teams annotated as red and blue. (b) and (c) show smaller blown-up sections of the game, highlighting the two respective shaded regions in (a).

rank correlation coefficient, giving a value of -0.09 with $P < 0.001$. Effectively, heavily disproving any issues regarding the learning of the network regressing on one variable.

VI. DISCUSSION

Dota 2’s replay function provides the ability to replay matches exactly. While delving in to the replay for Fig. 4a’s match accompanied by our prediction tool, we could more closely monitor false predictions from the network, rather than witnessing them statistically as mis-classifying a kill. What we found indicated that although the network would be certain a player would die, a myriad of external factors would come into play. For example whether a player would commit to a kill while not having full information on the enemy team, or simply miscalculating their damage. A further study would need to be conducted to understand what is exactly happening within these moments, perhaps with professional domain assistance. What is certain is that at each spike of kill prediction an encounter between two or more heroes occurs.

To determine the usefulness of the model as a commentator assisting tool, the ability to predict interesting situations must be evaluated. Using standard performance metrics such as precision or recall are not satisfactory for this purpose, since different kinds of mispredictions have different impact on the usefulness of the model. We can classify mispredictions into four categories:

- 1) False negatives. The hero will die, but the model failed to predict it.
- 2) False positives where hero is about to die, but not within 5 seconds.
- 3) False positives where hero is not about to die, but professional players would evaluate the situation as dangerous.
- 4) False positives where hero is not about to die, and professional players would evaluate the situation as not dangerous.

A model with a large number of category 1 errors can still provide assistance to a commentator, even though this

assistance is provided less often. On the other hand a large number of category 4 errors have a catastrophic effect on the usefulness of the model. If the model constantly alerts the commentator to uninteresting events, it becomes a hindrance instead of a help.

Category 2 and 3 mispredictions arguably make the model more useful and thus they are desirable. Category 2 mispredictions can alert the commentator to fights where someone will die further ahead in time. A game situation when someone barely escaped can be just as interesting or entertaining as when someone died. The ability to predict such situations makes the model more useful, thus category 3 mispredictions are desirable. Unfortunately, to differentiate between category 3 and 4 mispredictions, human expert input is needed. For this reason evaluating the performance is difficult and requires further work.

VII. CONCLUSION

In the above we have outlined our research on micro-predictions in the esports game Dota 2. Using a multilayered perceptron, trained on a vast array of features, that utilizes shared weights within the first half of the network, the ability to train a model that can accurately predict the death-state of every player within a live or recorded game of Dota 2, while efficiently dealing with the symmetric inputs of a MOBA data set. We further describe the process of pre-processing esports data from Dota 2 and how to replicate the experiments presented in their entirety, including the results of a large scale parameter exploration on network and feature architectures. Jointly, this will assist future experiments targeting micro-predictions into esports. The novel contribution of this paper is the exploration of deep learning models applied to Dota 2 at a micro level (or event level) as compared to match level. We provide a model for predicting the death of professional Dota 2 players with a high degree of accuracy within a 5 second window, with an example shown in Fig. 4a. Though we trained for a 5 second window of prediction, the model has learned inherent properties of features that constitute kills such that in certain situation the network can predict even prior to the 5 second window. The distribution of these probabilities can be seen in the violin plot in Fig. 2.

The model can also be developed into a focus tool for game commentators that can direct their focus and storytelling upwards of 5 seconds before kills occur within the game.

REFERENCES

- [1] J. Hamari and M. Sjöblom, "What is esports and why do people watch it?" *Internet research*, vol. 27, no. 2, pp. 211–232, 2017.
- [2] Superdata Research, "European esports conference brief (<http://strivesponsorship.com/wp-content/uploads/2017/04/superdata-2017-esports-market-brief.pdf>)," 2017.
- [3] M. Schubert, et al., "Esports analytics through encounter detection," in *Proceedings of the MIT Sloan Sports Analytics Conference*, 2016.
- [4] F. Block, et al., "Narrative bytes : Data-driven content production in esports," in : *TVX '18, Proc of ACM Int'l Conf. on Interactive Experiences for TV and Online Video*, pp. 29–41, ACM, 2018.
- [5] F. Block and A. Drachen, "The case for data in esports," *Develop*, vol. 185, p. 10, 2017.
- [6] G. Yannakakis, "Game AI Revisited," in *Proc. of ACM Computing Frontiers Conference*, pp. 285–292, 2012.
- [7] P. Yang, et al., "Identifying patterns in combat that are predictive of success in moba games," in *Proc of the 9th Int'l Conf. on Foundations of Digital Games*, FDG '14, ACM, 2014.
- [8] Y. Seo, "Electronic sports: A new marketing landscape of the experience economy," *Journal of Marketing Management*, vol. 29, no. 2, pp. 1542–1560, 2013.
- [9] H. Xue, et al., "E-sports management? institutional logics, professional sports, emerging esports field," in *2016 North American Society for Sport Management Conference, Orlando, FL*, June 2016.
- [10] S. Demediuk, et al., "Player retention in league of legends: a study using survival analysis," in *ACS'18 Proceedings of the Australasian Computer Science Week Multiconference*, 2018.
- [11] W. Wang, "Predicting multiplayer online battle arena (moba) game outcome based on hero draft data," Master's thesis, Masters thesis, Dublin, National College of Ireland, 2016.
- [12] V. Hodge, et al., "Win prediction in esports: Mixed-rank match prediction in multi-player online battle arena games," *ArXiv e-prints (CS:AI)*, vol. : 1711.06498, Nov. 2017.
- [13] Y. Yang, T. Qin, and Y.-H. Lei, "Real-time eSports Match Result Prediction," *ArXiv e-prints*, Dec. 2016.
- [14] G. Synnaeve and P. Bessière, "A bayesian model for plan recognition in rts games applied to starcraft," in *Seventh Artificial Intelligence and Interactive Digital Entertainment Conference*, 2011.
- [15] E. Bursztein, "I am a legend: Hacking hearthstone using statistical learning methods," in *2016 IEEE conference on computational intelligence and games (CIG)*, pp. 1–8, IEEE, 2016.
- [16] A. Summerville, M. Cook, and B. Steenhuisen, "Draft-analysis of the ancients: predicting draft picks in dota 2 using machine learning," in *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2016.
- [17] F. Rioult, et al., "Mining tracks of competitive video games," *AASRI Procedia*, vol. 8, pp. 82–87, 2014.
- [18] A. Drachen, et al., "Skill-based differences in spatio-temporal team behavior in defence of the ancients 2 (dota 2)," in *Proc of IEEE Games Media Entertainment Conf.*, 2014.
- [19] L. Gao, et al., "Classifying dota 2 hero characters based on play style and performance," *University of Utah Course on ML*, 2013.
- [20] C. Eggert, et al., "Classification of player roles in the team-based multi-player game dota 2," in *Int'l Conf. on Entertainment Computing*, pp. 112–125, Springer, 2015.
- [21] Z. Cleghern, et al., "Predicting future states in dota 2 using value-split models of time series attribute data," in *Proc of the 12th Int'l Conf. on Foundations of Digital Games*, FDG '17, ACM, 2017.
- [22] Steam, "Opendota - dota 2 statistics <https://www.opendota.com/>."
- [23] M. Schrodt, "Clarity parser for dota 2 and csgo replay files <https://github.com/skadi/stats/clarity>."
- [24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [25] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks* (M. A. Arbib, ed.), pp. 255–258, MIT Press, 1998.
- [26] Y. LeCun, et al., "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, pp. 2278–2324, 1998.
- [27] H. Lee, et al., "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Advances in neural information processing systems*, pp. 1096–1104, 2009.
- [28] J.-T. Huang, et al., "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *IEEE Int'l Conf. on Acoustics, Speech and Signal Processing*, pp. 7304–7308, 2013.
- [29] N. V. Chawla et al., "Special issue on learning from imbalanced data sets," *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 1–6, 2004.
- [30] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 281–305, 2012.