# **บทที่ 4 ผลการดำเนินงาน**

สำหรับผลการดำเนินงานการพัฒนาระบบรู้จำท่าทางภาษามือไทยด้วยโครงข่ายประสาทเทียมแบบวนกลับ แบ่งออกได้เป็น 4 ส่วน ได้แก่

4.1 ผลการฝึกฝนโมเดล

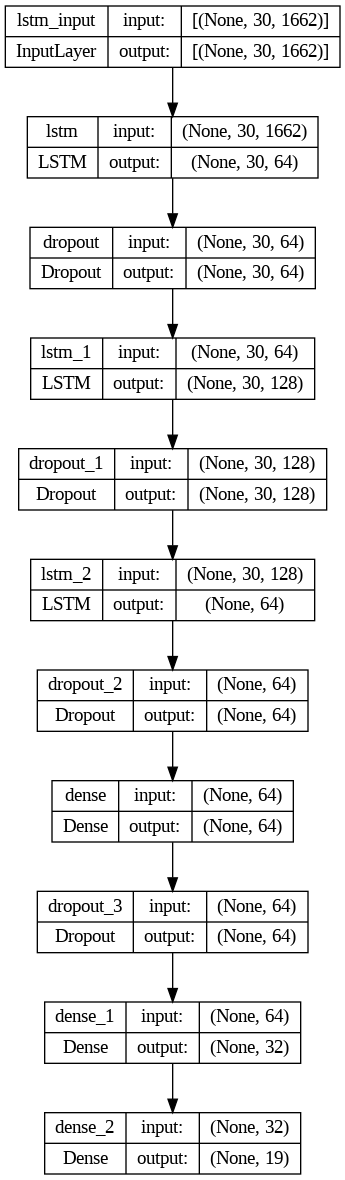
4.2 ผลการวัดประสิทธิภาพโมเดล

4.3 ผลการเปรียบเทียบประสิทธิภาพโมเดล

4.4 ผลการทดสอบใช้งานระบบรู้จำภาษามือ

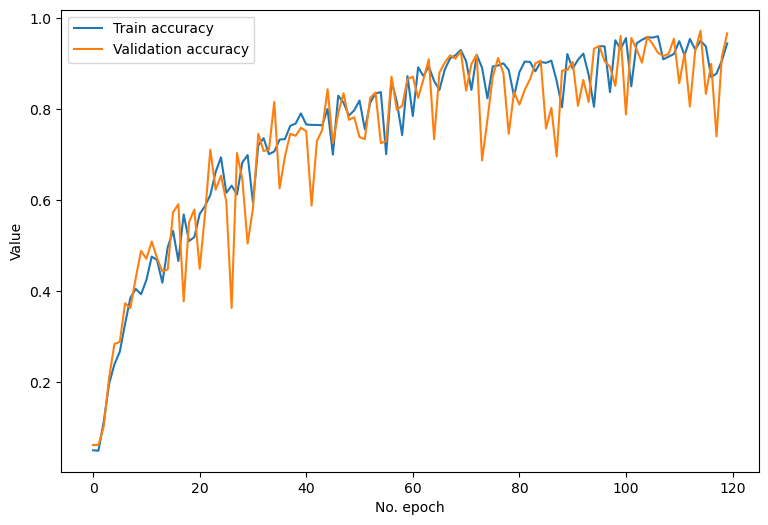
## **4.1 ผลการฝึกฝนโมเดล**

4.1.1 โมเดล LSTM

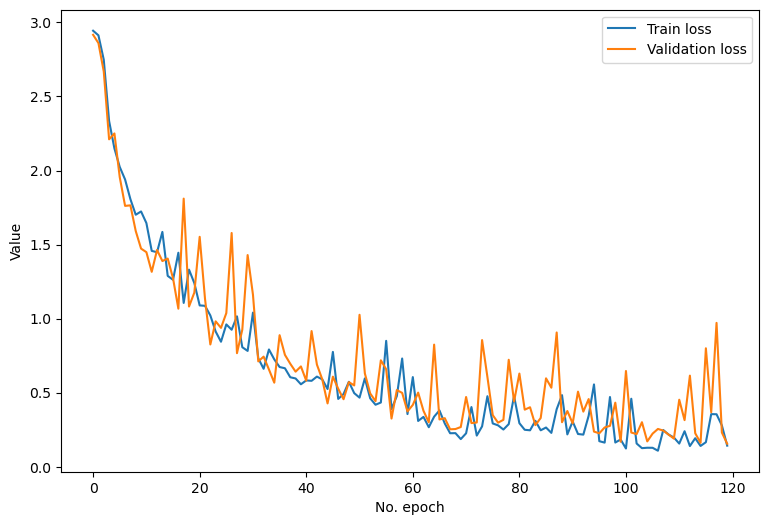


**ภาพที่ 4.1** โครงสร้างโมเดล LSTM

ผลจากการพัฒนาระบบรู้จำท่าทางภาษามือไทยด้วย RNN พบว่า การกำหนดค่าอัตราการเรียนรู้ Epochs เท่ากับ 120 และ Batch Size เท่ากับ 32 ของโมเดล LSTM ให้ค่า Accuracy สูงถึง 0.9 และค่า Loss ต่ำกว่า 0.2 ดังภาพด้านล่าง

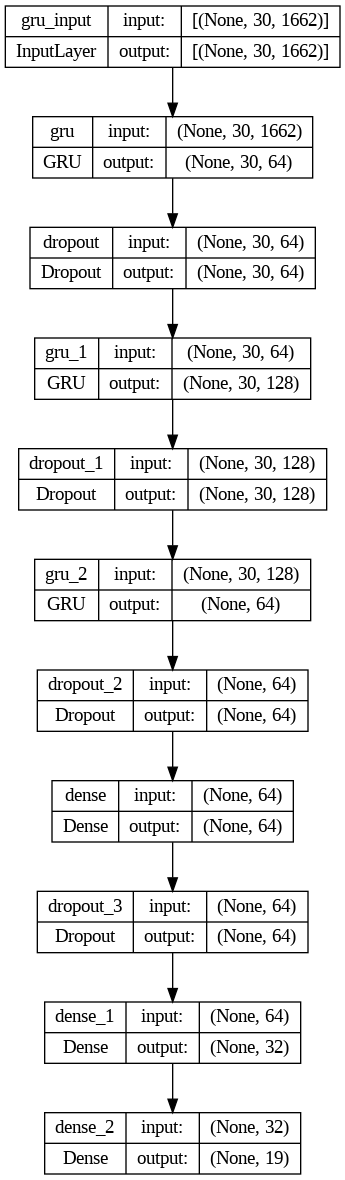


**ภาพที่ 4.2** กราฟแสดงจำนวนรอบการเทรนและค่าความถูกต้องของโมเดล LSTM



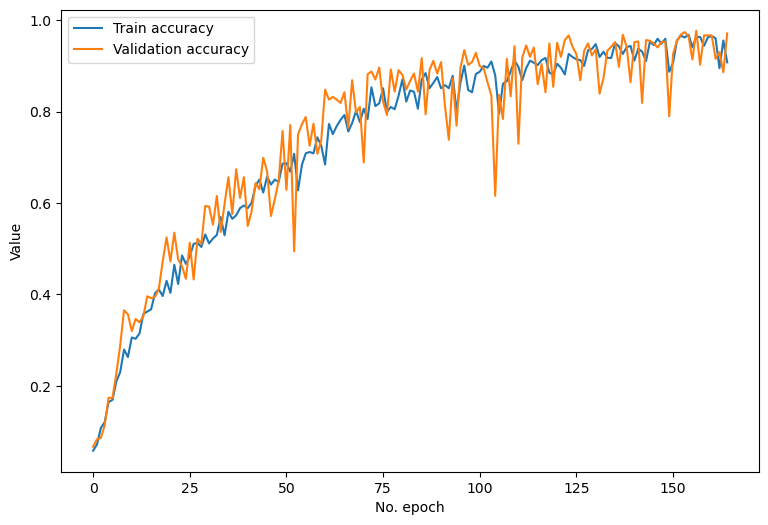
**ภาพที่ 4.3** กราฟแสดงรอบการเทรนและค่าความผิดพลาดของโมเดล LSTM

4.1.2 โมเดล GRU

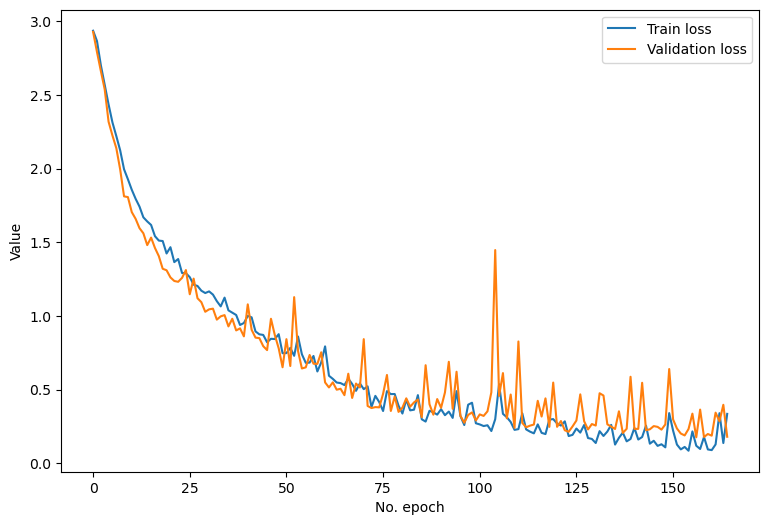


**ภาพที่ 4.4** โครงสร้างโมเดล GRU

ผลจากการพัฒนาระบบรู้จำท่าทางภาษามือไทยด้วย RNN พบว่า การกำหนดค่าอัตราการเรียนรู้ Epochs เท่ากับ 165 และ Batch Size เท่ากับ 32 ของโมเดล GRU ให้ค่า Accuracy สูงถึง 0.9 และค่า Loss ต่ำกว่า 0.2 ดังภาพด้านล่าง

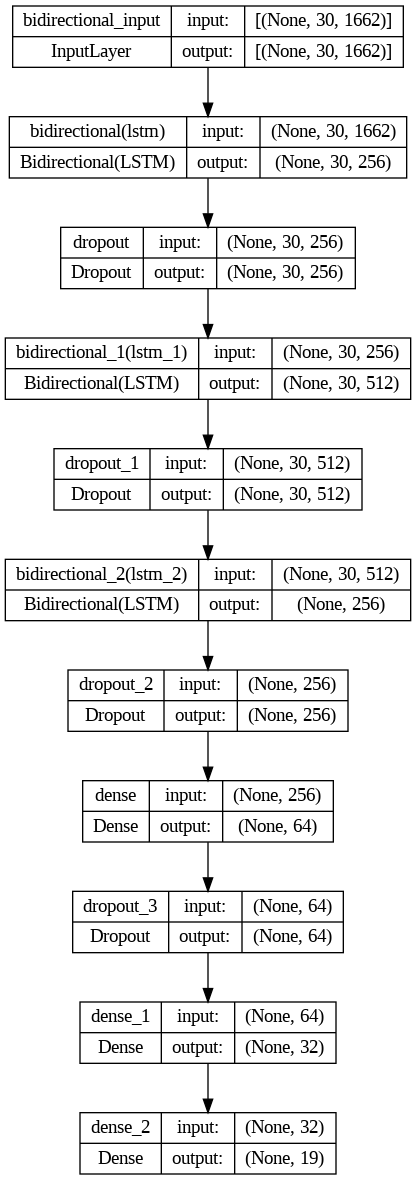


**ภาพที่ 4.5** กราฟแสดงจำนวนรอบการเทรนและค่าความถูกต้องของโมเดล GRU



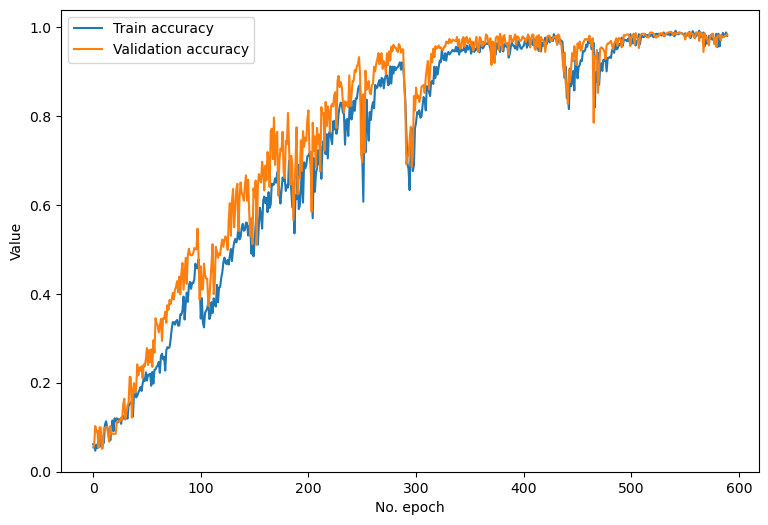
**ภาพที่ 4.6** กราฟแสดงรอบการเทรนและค่าความผิดพลาดของโมเดล GRU

4.1.3 โมเดล BiLSTM

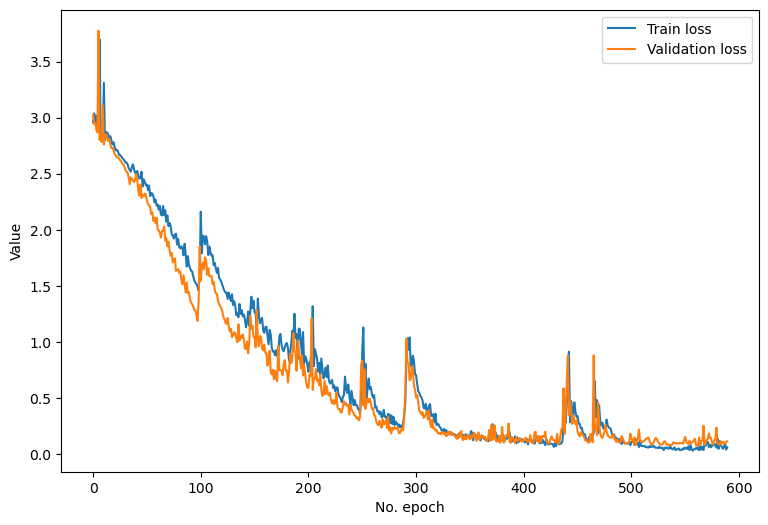


**ภาพที่ 4.7** โครงสร้างโมเดล BiLSTM

โมเดล BiLSTM มีการกำหนดค่าอัตราการเรียนรู้ Epochs เท่ากับ 590 และ ไม่มีใส่ Batch Size ในการเทรน เนื่องจากผู้วิจัยทดลองใส่ค่า Batch Size เท่ากับ 32 เหมือนกับโมเดลอีก 2 โมเดลก่อนหน้าในการการเทรนทำให้เกิด Exploding Gradient จึงปรับค่า Batch Size เป็นค่าเริ่มต้นเพื่อแก้ปัญหา แต่ขึ้นจำเป็นต้องใช้จำนวนรอบการเทรนที่มากกว่า 2 โมเดลก่อนหน้านี้หลายเท่าจึงจะให้ค่า Accuracy สูงถึง 0.9 และค่า Loss ต่ำกว่า 0.2 ดังภาพด้านล่าง



**ภาพที่ 4.8** กราฟแสดงจำนวนรอบการเทรนและค่าความถูกต้องของโมเดล BiLSTM



**ภาพที่ 4.9** กราฟแสดงรอบการเทรนและค่าความผิดพลาดของโมเดล BiLSTM

จากภาพของโครงสร้างโมเดลทั้ง 3 โมเดล ผู้วิจัยได้มีการใช้ Dense และ Dropout ซึ่ง Dense layer เป็นชั้นของโครงข่ายประสาทเทียมที่มีการเชื่อมต่อ (connection) ระหว่างโนด (node) ในชั้นก่อนหน้ากับชั้นต่อไป โดยทุกโนดในชั้นจะมีการเชื่อมต่อกับโนดทุกตัวในชั้นต่อไป ชั้น Dense จะมีการปรับค่าน้ำหนัก (weight) และค่าไบแอส (bias) ของโนดในแต่ละชั้น ซึ่งช่วยให้โมเดลสามารถเรียนรู้และสร้างรูปแบบ (pattern) ที่ซับซ้อนได้มากขึ้น

Dropout layer เป็นชั้นที่มีไว้เพื่อลดการเกิด overfitting ในโมเดล โดย Dropout layer จะสุ่มตัดการเชื่อมต่อระหว่างโนดในชั้นก่อนหน้ากับชั้นต่อไป โดยตัดการเชื่อมต่อเหล่านี้ด้วยการกำหนดค่าเป็นศูนย์ (zero) โดยสุ่มตัดบางโนดออกจากการคำนวณในแต่ละรอบการฝึกฝน การทำ Dropout จะช่วยให้โมเดลสามารถเรียนรู้และสร้างรูปแบบที่เหมาะสมกับข้อมูลได้ดีขึ้นโดยไม่เกิดการเรียนรู้ที่ผิดพลาดจาก overfitting

การใช้ Dense layer และ Dropout layer ในโมเดลประสาทเทียมช่วยให้โมเดลมีความแม่นยำในการทำนายข้อมูลมากขึ้น โดย Dense layer เป็นชั้นหลักในการสร้างโมเดลและ Dropout layer เป็นชั้นช่วยลดการเกิด overfitting ในโมเดลให้มีประสิทธิภาพสูงขึ้น การใช้ Dense layer และ Dropout layer นั้นเป็นเพิ่มความแม่นยำมากขึ้นนั่นเอง

## **4.2 ผลการวัดประสิทธิภาพโมเดล**

4.2.1 ผลการวัดประสิทธิภาพโมเดล LSTM

จากการเทรนโมเดลด้วยชุดข้อมูลสำหรับเทรนและข้อมูลสำหรับทดสอบได้ค่า Accuracy และค่า Loss ดังนี้

**ตารางที่ 4.1** ผลการวัดประสิทธิภาพโมเดล LSTM

|  |  |  |
| --- | --- | --- |
| **Data** | **Accuracy** | **Loss** |
| Train-Data | 0.98% | 0.06 |
| Test-Data | 0.96% | 0.15 |

และผลจากการประเมินด้วย Confusion Matrix ของแต่ละท่าภาษามือไทยได้ดังนี้

**ตารางที่ 4.2** Confusion Matrix โมเดล LSTM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **คำศัพท์** | **Confusion Matrix** | | | | **Accuracy** |
| **TP** | **FP** | **TN** | **FN** |
| ขอบคุณ | 650 | 0 | 32 | 2 | 0.997% |
| ขอโทษ | 646 | 5 | 33 | 0 | 0.993% |
| ไม่เป็นไร | 645 | 0 | 38 | 1 | 0.999% |
| สบายดี | 647 | 0 | 37 | 0 | 1.000% |
| โชคดี | 646 | 3 | 32 | 3 | 0.991% |
| คิดถึง | 643 | 1 | 39 | 1 | 0.997% |
| น่ารัก | 649 | 0 | 35 | 0 | 1.000% |
| สวย | 647 | 2 | 35 | 0 | 0.997% |
| ชอบ | 642 | 0 | 36 | 6 | 0.991% |
| ไม่ชอบ | 649 | 5 | 29 | 1 | 0.991% |
| รัก | 655 | 0 | 28 | 1 | 0.999% |
| เก่ง | 647 | 2 | 34 | 1 | 0.996% |
| ฉลาด | 644 | 0 | 39 | 1 | 0.999% |
| ไม่สบาย | 648 | 2 | 34 | 0 | 0.997% |
| เศร้า | 653 | 0 | 30 | 1 | 0.999% |
| เสียใจ | 643 | 1 | 40 | 0 | 0.999% |
| หิว | 649 | 0 | 34 | 1 | 0.999% |
| อิ่ม | 637 | 2 | 45 | 0 | 0.997% |
| เข้าใจ | 649 | 0 | 33 | 2 | 0.997% |
| **รวม** | **12289** | **23** | **663** | **21** | **0.997%** |

4.2.2 ผลการวัดประสิทธิภาพโมเดล GRU

จากการเทรนโมเดลด้วยชุดข้อมูลสำหรับเทรนและข้อมูลสำหรับทดสอบได้ค่า Accuracy และค่า Loss ดังนี้

**ตารางที่ 4.3** ผลการวัดประสิทธิภาพโมเดล GRU

|  |  |  |
| --- | --- | --- |
| **Data** | **Accuracy** | **Loss** |
| Train-Data | 0.98% | 0.06 |
| Test-Data | 0.97% | 0.17 |

และผลจากการประเมินด้วย Confusion Matrix ของแต่ละท่าภาษามือไทยได้ดังนี้

**ตารางที่ 4.4** Confusion Matrix โมเดล GRU

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **คำศัพท์** | **Confusion Matrix** | | | | **Accuracy** |
| **TP** | **FP** | **TN** | **FN** |
| ขอบคุณ | 645 | 0 | 39 | 0 | 1.000% |
| ขอโทษ | 656 | 0 | 28 | 0 | 1.000% |
| ไม่เป็นไร | 653 | 0 | 31 | 0 | 1.000% |
| สบายดี | 656 | 0 | 28 | 0 | 1.000% |
| โชคดี | 656 | 3 | 23 | 2 | 0.993% |
| คิดถึง | 638 | 7 | 39 | 0 | 0.990% |
| น่ารัก | 643 | 1 | 39 | 3 | 0.994% |
| สวย | 640 | 0 | 41 | 3 | 0.996% |
| ชอบ | 649 | 1 | 33 | 1 | 0.997% |
| ไม่ชอบ | 639 | 1 | 43 | 1 | 0.997% |
| รัก | 648 | 0 | 36 | 0 | 1.000% |
| เก่ง | 641 | 1 | 39 | 3 | 0.994% |
| ฉลาด | 647 | 0 | 30 | 7 | 0.990% |
| ไม่สบาย | 641 | 1 | 41 | 1 | 0.997% |
| เศร้า | 641 | 1 | 42 | 0 | 0.999% |
| เสียใจ | 658 | 0 | 26 | 0 | 1.000% |
| หิว | 647 | 0 | 36 | 1 | 0.999% |
| อิ่ม | 644 | 2 | 38 | 0 | 0.997% |
| เข้าใจ | 650 | 0 | 33 | 1 | 0.999% |
| **รวม** | **12292** | **18** | **665** | **23** | **0.999%** |

4.2.3 ผลการวัดประสิทธิภาพโมเดล BiLSTM

จากการเทรนโมเดลด้วยชุดข้อมูลสำหรับเทรนและข้อมูลสำหรับทดสอบได้ค่า Accuracy และค่า Loss ดังนี้

**ตารางที่ 4.5** ผลการวัดประสิทธิภาพโมเดล BiLSTM

|  |  |  |
| --- | --- | --- |
| **Data** | **Accuracy** | **Loss** |
| Train-Data | 0.99% | 0.01 |
| Test-Data | 0.98% | 0.11 |

และผลจากการประเมินด้วย Confusion Matrix ของแต่ละท่าภาษามือไทยได้ดังนี้

**ตารางที่ 4.6** Confusion Matrix โมเดล GRU

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **คำศัพท์** | **Confusion Matrix** | | | | **Accuracy%** |
| **TP** | **FP** | **TN** | **FN** |
| ขอบคุณ | 643 | 40 | 1 | 0 | 0.999% |
| ขอโทษ | 641 | 43 | 0 | 0 | 1.000% |
| ไม่เป็นไร | 647 | 36 | 0 | 1 | 0.999% |
| สบายดี | 646 | 38 | 0 | 0 | 1.000% |
| โชคดี | 650 | 32 | 2 | 0 | 0.997% |
| คิดถึง | 656 | 26 | 2 | 0 | 0.997% |
| น่ารัก | 648 | 35 | 0 | 1 | 0.999% |
| สวย | 644 | 37 | 2 | 1 | 0.996% |
| ชอบ | 649 | 34 | 0 | 1 | 0.999% |
| ไม่ชอบ | 643 | 40 | 1 | 1 | 0.997% |
| รัก | 650 | 34 | 0 | 0 | 0.000% |
| เก่ง | 662 | 22 | 0 | 0 | 1.000% |
| ฉลาด | 646 | 37 | 0 | 1 | 0.999% |
| ไม่สบาย | 653 | 31 | 0 | 0 | 1.000% |
| เศร้า | 637 | 45 | 0 | 2 | 0.997% |
| เสียใจ | 650 | 34 | 0 | 0 | 1.000% |
| หิว | 647 | 35 | 0 | 2 | 0.997% |
| อิ่ม | 646 | 35 | 3 | 0 | 0.996% |
| เข้าใจ | 643 | 39 | 0 | 2 | 0.997% |
| **รวม** | **12301** | **673** | **11** | **12** | **0.998%** |

## **4.3 ผลการเปรียบเทียบประสิทธิภาพโมเดล**

ผลจากการพัฒนาและวัดประสิทธิภาพของโมเดลทั้ง 3 แบบได้แก่ LSTM, GRU และ BiLSTM ที่ใช้ในการพัฒนาระบบรู้จำท่าทางภาษามือไทยด้วย RNN ซึ่งแสดงได้ดังต่อไปนี้

**ตารางที่ 4.7** เปรียบประสิทธิภาพของโมเดลทั้ง 3 โมเดล

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Models** | **Epochs** | **Accuracy** | | **Loss** | | **Confusion Matrix** |
| Train-Data | Test-Data | Train-Data | Test-Data |
| LSTM | 120 | 0.98% | 0.96% | 0.06 | 0.15 | 0.997% |
| GRU | 165 | 0.98% | 0.97% | 0.06 | 0.17 | 0.999% |
| BiLSTM | 590 | 0.99% | 0.98% | 0.01 | 0.11 | 0.998% |

จากตารางเปรียบเทียบประสิทธิภาพโมเดลทั้ง 3 โมเดล BiLSTM มีค่าความถูกต้องสูงอยู่ที่ 0.99% ซึ่งมากที่สุดในโมเดลทั้ง 3 และมีค่า Loss อยู่ที่ 0.1 แต่มีจำนวนรอบการในการเทรนมากที่สุดเหมือนกันอยู่ที่ 590 รอบ ซึ่งแตกต่างจากโมเดล LSTM และโมเดล GRU ที่มีจำนวนการเทรนตั้งแต่ 120 – 165 แต่ให้ค่า Accuracy ใกล้เคียงกับโมเดล BiLSTM นอกจากความแตกต่างของจำนวนในการเทรนของโมเดลแล้วยังมีเรื่องของเวลาที่ใช้ในการเทรนที่แตกต่างระหว่างโมเดล LSTM และโมเดล GRU แม้ว่าจำนวนในการเทรนของโมเดล LSTM จะน้อยกว่าโมเดล GRU แต่ใช้เวลาในการเทรนมากกว่าเนื่องจากจำนวนพารามิเตอร์ของโมเดล GRU นั้นน้อยกว่าจึงทำให้ใช้เวลาในการเทรนน้อยกว่าแม้จำนวนรอบที่ใช้ในการเทรนจะมากกว่า

## **4.4 ผลการทดสอบใช้งานระบบรู้จำภาษามือ**

โดยจะเป็นทดสอบทำท่าทางภาษามือไทยด้วยโมเดล GRU ทั้ง 19 คำ คำละ 10 ครั้ง เพื่อทดสอบว่าโมเดลทำนายได้ดีแค่ไหน ผลได้ดังตาราง