การทำนายการฉ้อโกงบัตรเครดิตด้วยแบบจำลองปัญญาประดิษฐ์

Credit Card Fraud Prediction using Artificial Intelligence

Chansing Sem[1], ธนพร สุเตนัน[2], ปวาริษา รัตนเทียนทอง[3] , ปัญญนัท อ้นพงษ์[4*]

[1, 2, 3, 4]**สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร**

Emails: [1]sem_c@silpakorn.edu, [2]sutenan_t@silpakorn.edu,

[3]rattanathiantho_p@silpakorn.edu, [4*]aonpong_p@silpakorn.edu

*** อาจารย์ที่ปรึกษา, ผู้ประพันธ์บรรณกิจ**

บทคัดย่อ

เนื่องจากบัตรเครดิตช่วยให้ง่ายและสะดวกในการชำระเงินเพราะผู้ใช้ไม่จำเป็นต้องพกเงินสดติดตัว ทำให้การใช้จ่ายทางบัตรเครดิตมีการเพิ่มสูงขึ้นแต่ก็มีปัญหาในการฉ้อโกงบัตรที่ส่งผลกระทบต่อผู้ใช้มาด้วย เป้าหมายของงานคือการพัฒนาเทคนิคที่สามารถป้องกันและตรวจจับการทำนายการฉ้อโกงบัตรเครดิตแบบจำลองปัญญาประดิษฐ์ที่สามารถช่วยเหลือธนาคารหรือสถาบันทาง การเงินในการตรวจจับธุรกรรมที่ผิดปกติ ในการทดสอบ มี 3 เมธอด และ เมธอดที่เหมาะกับชุดข้อมูล แล้วให้ผลความถูกต้องแม่นยำที่สุด คือเมธอดที่ 3 ซึ่งเป็นการนำชุดข้อมูลมาทำการลดมิติ (PCA) การเลือกฟีเจอร์ (Feature Selection) การใช้เทคนิค SMOTE ในการฝึกฝนโมเดลจากการทดลองแสดงว่าต้องใช้ ANOVA ในการเลือกฟีเจอร์ แล้วโมเดลที่ดีที่สุด คือ Random Forest

คำสำคัญ--การเรียนรู้ของเครื่อง, ปัญญาประดิษฐ์, การทำนาย, การฉ้อโกงบัตรเครดิต

Abstract

Credit card fraud detection is dataset which is focuses on detecting in credit card transaction using machine learning to identify whether it is fraudulent or not. There are small number of fraud. Due to the imbalance, Smote technique is applied. Feature selection methods such as Recursive Feature Elimination (RFE), ANOVA, and Lasso are applied to identify the most relevant features improving model performance by reducing dimensionality and focusing on key variables. Various algorithms like training models such as SVM, kNN, logistic regression, random forest, decision trees and ANN are evaluated using performance metrics such as accuracy, precision, recall, and F1-score. The goal is to identify the best model for detecting fraud while minimizing errors and false positives.

Keywords -- machine learning, artificial intelligent, prediction, fraudulent transactions

1. บทนำ

การฉ้อโกงบัตรเครดิตในยุคดิจิทัลทำให้ธุรกิจและผู้ใช้ได้รับความเสียหายมากมายซึ่งความสำคัญ ในการตรวจจับการฉ้อโกงบัตรเครดิต ในยุคดิจิทัลทำให้ธุรกรรมออนไลน์เพิ่มประสิทธิภาพมากขึ้น จึงเป็นสิ่งที่จำเป็นพัฒนา เพื่อเพิ่มความมั่นใจ ปลอดภัยในการใช้บัตรเครดิตให้ปลอดภัยยิ่งขึ้น ในช่วง 4 ปีที่ผ่านมา นักวิจัยในประเทศไทยได้พัฒนาวิธีการต่างๆ เพื่อเพิ่มประสิทธิภาพในการตรวจจับการฉ้อโกงบัตรเครดิต เช่น การศึกษาโดย ธนกร และคณะ ในปี 2564 ที่นำเสนอการการเรียนรู้เชิงลึก (Deep Learning) กับการวิเคราะห์ฟีเจอร์เบื้องต้นพบว่าสามารถตรวจจับธุรกรรมที่น่าสงสัยได้ แต่ยังมีข้อจำกัดด้านการประมวลผลข้อมูลขนาดใหญ่ [1]. ต่อมาในปี 2565 ศิริกาญจน์ และคณะ ได้แนะนำเทคนิคการเรียนรู้เชิงลึกแบบผสม (Hybrid Deep Learning) ที่ช่วยเพิ่มความแม่นยำในการตรวจจับแต่ยังพบปัญหาด้านการจัดการข้อมูลที่มีความซับซ้อนสูง [2]. นอกจากนี้ ในปี 2566 จิรวัฒน์ และคณะ ได้นำเสนอการใช้การวิเคราะห์ฟีเจอร์แบบ ANOVA ร่วมกับเทคนิค XGBoost ที่ทำให้การตรวจจับฉ้อโกงมีความสามารถมากขึ้นในการจัดการกับข้อมูลที่ไม่สมดุล แต่ยังพบความท้าทายในการจัดการข้อมูลขนาดใหญ่ [3]. จากข้อจำกัดในงานวิจัยก่อนหน้า งานวิจัยนี้จึงนำเสนอแนวทางใหม่ในการตรวจจับการฉ้อโกงบัตรเครดิตโดยใช้เทคนิคการเลือกฟีเจอร์แบบ ANOVA ที่เกี่ยวกับข้อมูลที่สุดร่วมกับการเรียนรู้ของเครื่องแบบป่าสุ่ม (Random Forest) และการปรับสมดุลข้อมูลด้วยการ SMOTE เพื่อเพิ่มความแม่นยำ และลดข้อจำกัดของข้อมูลที่มีความซับซ้อน

2. งานวิจัยและทฤษฎีที่เกี่ยวข้อง

โดยที่งานวิจัยที่เกี่ยวข้องที่นำมาใช้ มีจุดเด่นและข้อจำกัดที่ต่างกัน โดยที่งานวิจัยเรื่องตรวจจับการฉ้อโกงบัตรเครดิตด้วย Logistic Regression โดยใช้เทคนิค และการวิเคราะห์ใน Machine Learning โดยที่จุดเด่นของงานวิจัยนี้ คือการใช้เทคนิคที่เข้าใจง่าย และเลือกใช้โมเดล Logistic Regression ซึ่งโมเดลนี้เหมาะกับข้อมูลของงานวิจัยนี้ที่ใช้ข้อมูลธุรกรรมที่มีข้อมูลมีขนาดไม่ใหญ่มากและไม่ซับซ้อนเหมือนกับโมเดล XGBoost หรือ Random Forest แต่ข้อจำกัด คือการใช้โมเดล Logistic Regression อาจจะไม่สามารถจับความสัมพันธ์แบบเชิงเส้นได้ดีอาจจะทำให้ความแม่นยำลดลงถ้าในกรณีข้อมูลมีความซับซ้อนมาก และการจะนำข้อมูลมาวิเคราะห์แต่ใช้โมเดลเพียงอันเดียว อาจจะไม่เพียงพอหาเปรียบเทียบกับการใช้โมเดลอื่นที่มีความหลากหลายมากกว่า โดยที่ผลการศึกษาพบว่าโมเดลที่สร้างมีความถูกต้อง 92% โดยดูจากค่า accuracy, precision, recall, f1 score และ ROC Curve

ถัดมา [4] งานวิจัยเกี่ยวข้องกับเรื่อง การตรวจจับการฉ้อโกงประกันภัยรถยนต์โดยใช้การวิเคราะห์ข้อความและการเรียนรู้ของเครื่อง โดยที่จุดเด่น คือการใช้โมเดลที่หลากหลาย เช่น Logistic Regression, XGBoostClassifier, K-nearest Neighbors, Random Forest , Support Vector Classifier (SVC), Gradient Boosting กับข้อมูลที่มีขนาดใหญ่ และมีความซับซ้อนมากสามารถเปรียบเทียบได้ดีกว่า การเลือกใช้โมเดล Logistic Regression เพียงโมเดลเดียว จึงทำให้สามารถนำมาเปรียบเทียบหาโมเดลที่มีความเหมาะสมกับข้อมูลที่ ผู้วิจัยเลือกนำมาใช้ได้ แต่ข้อจำกัด คือการเลือกใช้โมเดลหลากหลายข้อมูลที่มีขนาดใหญ่ และมีความซับซ้อนมาก อาจจะต้องการทรัพยากรคอมพิวเตอร์ที่สูงและใช้เวลาในการประมวลผลที่นานกว่า และยากต่อการอธิบายผลลัพธ์ เมื่อเทียบกับการเลือกใช้เพียงโมเดล Logistic Regression เพียงโมเดลเดียว โดยที่ผลการศึกษาพบว่าการพัฒนาแบบจำลอง โดยการใช้เทคนิค

วิธี XGBoost ให้ค่าความไว (Recall) ที่มากที่สุด ซึ่งมีค่าเท่ากับ 0.97 มีค่าความถูกต้อง (Accuracy) เท่ากับ 0.37 แต่เทคนิควิธี K-Nearest Neighbors (KNN) ให้ค่าความไว (Recall) ที่น้อยที่สุด ซึ่งมีค่าเท่ากับ 0.57 มีค่าความถูกต้อง (Accuracy) เท่ากับ 0.55 ดังนั้นผู้วิจัยจึงนำจุดเด่น และข้อจำกัดของงานวิจัยที่เกี่ยวข้อง นำมาประยุกต์ใช้ในงานของผู้วิจัย [5].

2.1 Support Vector Machine (SVM) [6]

เป็นวิธีการเรียนรู้ที่ใช้สำหรับการจำแนกการวิเคราะห์ถดถอย และการตรวจจับข้อมูลผิดปกติมีประสิทธิภาพในข้อมูลที่มีมิติสูง มีความยืดหยุ่น สามารถทำงานได้ดีโดยเฉพาะอย่างยิ่งเมื่อข้อมูลมีความซับซ้อน แต่จำนวนตัวอย่างไม่มาก จึงไม่สามารถให้ค่าความน่าจะเป็นโดยตรงได้

2.2 K-Nearest Neighbour (K-NN) [7]

คือเพื่อนบ้านที่ใกล้เคียงที่สุดเป็นวิธีการในการทำนายคลาสของข้อมูลใหม่ โดยที่จะหาข้อมูลเก่าที่ใกล้เคียงที่สุดจำนวน k ชุด เพื่อช่วยในการตัดสินใจว่าข้อมูลใหม่ควรอยู่ในคลาสไหน และวิธีการทำงาน คือ โปรแกรมจะวัดระยะห่างระหว่างข้อมูลใหม่กับข้อมูลเก่าโดยทั่วไปใช้วิธีการวัดระยะทาง แบบยูคลิด (Euclidean) หรือ แบบแมนฮัตตัน (Manhattan)

2.3 Logistic Regression [8]

การถดถอยโลจิสติก (Logistic Regression) เป็นเทคนิคที่ใช้กันอย่างแพร่หลายในปัญหาการจำแนกประเภทมีที่รองรับการจำแนกหลายคลาส เนื่องจากมีความง่ายในการตีความ และการใช้งานโดยที่มีการตั้งค่าของ multi_class ให้เป็น ovr เพื่อใช้อัลกอริธึม one-vs-rest [9] หรือ multinomial สำหรับการใช้ cross-entropy loss (รองรับเฉพาะ solver เช่น lbfgs, sag, saga, และ newton-cg) และโมเดลนี้ยังมีการปรับ regularization เพื่อช่วยลดปัญหา overfitting และสามารถตั้งค่าเพิ่มเติมได้

2.4 Artificial Neural Network (ANN) [10]

เป็นโมเดลเครือข่ายประสาทเทียม โดยเฉพาะ Multi-layer Perceptron (MLP) เป็นโมเดลของการเรียนรู้แบบ Supervised learning ที่ใช้ในงาน Machine Language ได้หลากหลายแบบ เช่น การจำแนกประเภท (Classification) และการถดถอย (Regression) และการทำงานของโมเดล MLP (Multi-layer Perceptron) ในการเรียนรู้ฟังก์ชัน จากชุดข้อมูลที่กำหนดซึ่งมี hidden layers, ระหว่าง input layer และ output layer

2.5 Decision Tree [11]

เป็นโมเดล Machine Learning ที่ใช้ในการสร้างการตัดสินใจในรูปแบบของโครงสร้างต้นไม้ โดยมีส่วนประกอบ คือ

1.Root Node คือ โหนดเริ่มต้นของต้นไม้ เพื่อใช้แยกข้อมูลตามคุณสมบัติที่สำคัญที่สุด

2.Internal Nodes คือ โหนดกลาง เพื่อใช้ในการตัดสินใจเพิ่มเติมจากข้อมูลที่แตกแขนงมาจาก Root Node

3.Leaf Nodes คือ โหนดปลายทางที่ไม่มีการแยกย่อยต่อแสดงผลลัพธ์ของการตัดสินใจหรือการคาดการณ์

2.6 Random Forest [12]

เป็นโมเดล Machine Learning ที่ใช้วิธีรวมผลการตัดสินใจจากหลาย Decision Trees เพื่อเพิ่มความแม่นยำในการทำนายทำงานได้ดีทั้งในกรณีที่ข้อมูลมีความซับซ้อน หรือมีฟีเจอร์จำนวนมาก และสรุปคือ Random Forest เป็น Ensemble model ซึ่งจะประกอบไปด้วยหลายๆ Decision Trees และซึ่งแต่ละต้นไม้จะได้รับการฝึกด้วยข้อมูลที่แตกต่างกันโดยสุ่มเลือกบางฟีเจอร์ และข้อมูลบางส่วน

3. ชุดเครื่องมือและชุดข้อมูล

3.1 ชุดเครื่องมือ

3.1.1 Python

Python เป็นภาษาที่นิยมในเชิง Machine Language เนื่องจากมีไลบรารี และเครื่องมือที่ช่วยให้การพัฒนาโมเดลเป็นไปได้อย่างสะดวก และชุดข้อมูลที่มีอยู่แล้ว เช่น iris dataset , breast cancer dataset หรือดึงข้อมูลจากแหล่งที่น่าเชื่อถือ เช่น Kaggle

3.1.2 Library

Pandas เพื่อจัดการและประมวลผลข้อมูลในรูปแบบ DataFrame

sklearn.model_selection ใช้ train_test_split เพื่อแบ่งข้อมูลเป็นชุดฝึกและชุดทดสอบ

โมเดลจาก sklearn ที่นำใช้ได้แก่ SVC (SVM), KNeighborsClassifier (KNN), LogisticRegression, RandomForestClassifier, DecisionTreeClassifier และ MLPClassifier (ANN)

Sklearn.metrics ใช้วัดประสิทธิภาพของโมเดล เช่น accuracy_score, precision_score, recall_score, f1_score, classification_report, confusion_matrix imblearn.over_sampling

ใช้ SMOTE เพื่อปรับสมดุลข้อมูลโดยเพิ่มตัวอย่างสังเคราะห์สำหรับกลุ่มที่มีจำนวนน้อย

matplotlib.pyplot และ seaborn ใช้สร้างภาพข้อมูลเพื่อการวิเคราะห์

sklearn.preprocessing ใช้ StandardScaler เพื่อที่จะปรับมาตรฐานข้อมูลให้มีค่าเฉลี่ยเป็น 0 และส่วนเบี่ยงเบนมาตรฐานให้เป็น 1

sklearn.feature_selection ใช้ RFE เพื่อเลือกคุณลักษณะโดยลบคุณลักษณะที่สำคัญน้อยออกทีละตัว

SelectKBest และ f_classif เพื่อเลือกคุณลักษณะด้วยการวิเคราะห์ และใช้ ANOVA เพื่อเลือกคุณลักษณะที่มีความสัมพันธ์สูงสุดกับเป้าหมาย

3.1.3 Google Colab [13]

Google Colab (ย่อมาจาก Colaboratory) เป็นบริการคลาวด์ (Cloud) ที่ผู้ใช้สามารถเขียน และรันโค้ด Python ผ่านเว็บเบราว์เซอร์ได้ โดยไม่ต้องติดตั้งซอฟต์แวร์ใดๆบนเครื่องคอมพิวเตอร์ของตนเอง โดยใช้รูปแบบของ Jupyter Notebook ซึ่งเป็นสภาพแวดล้อมการเขียนโปรแกรมแบบโต้ตอบที่ได้รับความนิยมในชุมชนวิทยาศาสตร์ข้อมูล และการเรียนรู้ของเครื่อง

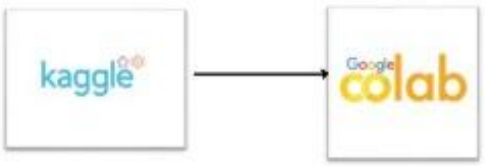
3.2 ชุดข้อมูล

ข้อมูลที่ใช้ในการศึกษาได้มาจาก ชุดข้อมูล ‘Credit Card Fraud Detection’ ที่สามารถดาวน์โหลดได้จากเว็บไซต์ Kaggle [14] ภายในชุดข้อมูลประกอบด้วยการทำธุรกรรมทั้งหมด 284,807 รายการ โดยมีคอลัมน์ที่บ่งบอกถึงรายละเอียดการทำธุรกรรม เช่น จำนวนเงินที่ใช้เวลาในการทำธุรกรรม และคุณสมบัติที่ถูกแปลงด้วยเทคนิค PCA เพื่อรักษาความเป็นส่วนตัวของผู้ใช้

4. วิธีการดำเนินการวิจัย

วิธีการดำเนินงานวิจัยสำหรับการตรวจจับการฉ้อโกงบัตรเครดิตกระบวนการแบ่งออกเป็น 3 ขั้นตอน ดังนี้

4.1 ขั้นตอนการดึงข้อมูลจากเว็บไซต์ (Kaggle)



ภาพที่ 1 การดึงข้อมูลจาก Kaggle

ในขั้นตอนการดึงข้อมูลสามารถดำเนินการ โดยการค้นหาข้อมูล 'Credit Card Fraud Detection' จาก Kaggle ผู้ใช้สามารถ Download dataset มาเป็นไฟล์ CSV และนำ dataset มาใช้ทำการทดลองผ่าน Google Colab โดยมีขั้นตอนการดำเนินงาน ดังกระบวนการใน **ภาพที่ 1**

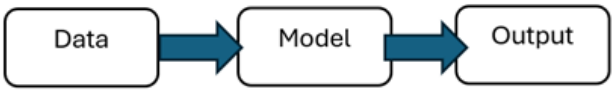
4.2 ขั้นตอนการจำแนกประเภท

ในการจำแนกประเภทผู้วิจัยได้แบ่งการทดลองออกเป็น 3 เมธอด โดยแยกจากกัน เพื่อนำมาเปรียบเทียบประสิทธิภาพ ได้แก่

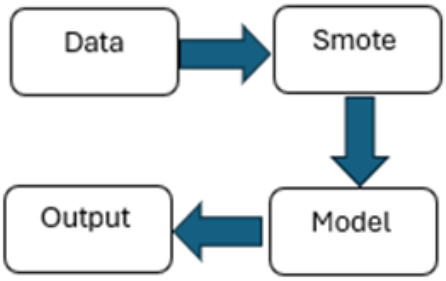
เมธอดที่ 1 คือการนำข้อมูลที่ได้ไปใช้ในแบบจำลองการเรียนรู้ของเครื่องโดยตรง เนื่องจากข้อมูลที่ได้จากแหล่งข้อมูลเป็นข้อมูลที่ถูกจัดการและดำเนินการ PCA เพื่อลดมิติข้อมูลไปแล้ว จึงทำให้ข้อมูลมีความพร้อมในการดำเนินการผ่านกระบวนการ การเรียนรู้ของเครื่องระดับหนึ่งผู้วิจัยจึงนำข้อมูลนี้ไปใช้โดยตรง เพื่อสร้างผลลัพธ์อ้างอิงโดยผลลัพธ์อ้างอิงนี้จะเป็นผลจากการดำเนินการ การเรียนรู้ของเครื่องโดยตรง ดังภาพที่แสดงใน **ภาพที่ 2** อย่างไรก็ตามผู้วิจัยนำเสนอ

เมธอดที่ 2 ซึ่งเป็นการเพิ่มประสิทธิภาพของการเรียนรู้ของเครื่อง โดยการใช้เทคนิคการเลือกฟีเจอร์ **(Feature Selection) [15]** โดยที่ผู้วิจัยจะเลือกใช้เทคนิคการเลือกฟีเจอร ได้แก่ RFE และ ANOVA เพื่อทำการเลือกฟีเจอร์ด้วยเมธอดที่แตกต่างกัน และนำมาทำการทดสอบกับแบบจำลองการเรียนรู้ของเครื่องที่แตกต่างกัน ดังแสดงภาพรวมกระบวนการใน **ภาพที่ 3** นอกจากนี้ ผู้วิจัยยังเพิ่มประสิทธิภาพของแบบจำลองอีกขั้น โดยนำเสนอ

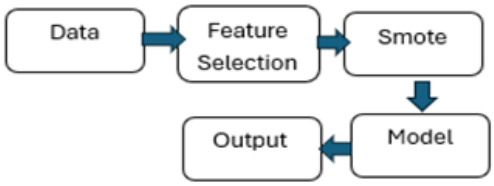
เมธอดที่ 3 ใช้หลักการ Data Balancing ให้กับข้อมูล เนื่องจากข้อมูลการฉ้อโกงผ่านระบบบัตรเครดิตแบบข้อมูลที่มีความไม่สมดุลเป็นอย่างมาก ผู้วิจัยจึงนำข้อมูลแบบเดียวกับที่สามารถคัดเลือกได้ด้วย **เมธอดที่ 2** ไปผ่านที่กระบวนการ Data Balancing ชื่อว่า SMOTE [16] ก่อนจะนำเข้าสู่แบบจำลองการเรียนรู้ของเครื่อง ดังแสดงใน**ภาพที่ 4** โดยคาดหวังว่า **เมธอดที่ 2** และ **เมธอดที่ 3** จะสามารถเพิ่มประสิทธิภาพให้แบบจำลองการเรียนรู้ของเครื่องได้



ภาพที่ 2 กระบวนการแบ่งข้อมูลออกเป็น 2 ส่วน เพื่อใช้ในการฝึกฝน และทดสอบโมเดล



ภาพที่ 3 กระบวนการแบ่งข้อมูลออกเป็น 2 ส่วน เพื่อใช้ในการฝึกฝน และทดสอบโมเดล สำหรับชุดข้อมูลนี้โมเดลแบ่งเป็น 80 กับ 20



ภาพที่ 4 กระบวนการแบ่งข้อมูลออกเป็นส่วนเพื่อใช้ในการฝึก และทดสอบโมเดล สำหรับชุดข้อมูลนี้โมเดลแบ่งเป็น 80 กับ 20

ชุดข้อมูล: ดึงข้อมูลมาจาก Kaggle การเตรียมข้อมูล: แบ่งชุดข้อมูลออกเป็นข้อมูลฝึก (Training set) และ ข้อมูลทดสอบ (Test set) 80:20 การเลือกฟีเจอร์ (Feature Selection): ใช้เทคนิค ANOVA, RFE ในการเลือกฟีเจอร์ที่สำคัญต่อการตรวจจับการฉ้อโกงการปรับสมดุลข้อมูลด้วย SMOTE (Synthetic Minority Over-sampling Technique):

สำหรับการจัดการกับข้อมูลที่ไม่สมดุล โดยทำการสร้างตัวอย่างข้อมูลการฉ้อโกงเพิ่มขึ้นในลักษณะการสุ่ม เพื่อช่วยปรับสมดุลระหว่างข้อมูลการฉ้อโกง และข้อมูลปกติ การเลือกโมเดล และการตั้งค่า: ทดสอบโมเดลที่หลากหลาย เช่น Decision Tree, Random Forest, Logistic Regression, KNN, SVM, และ ANN โดยตั้งค่าพารามิเตอร์ของแต่ละโมเดลให้เหมาะสมกับชุดข้อมูลที่ใช้เพื่อให้ได้ผลลัพธ์ที่มีความแม่นยำสูงสุด การประเมินผลการทดลอง: ใช้เมทริกซ์ต่างๆในการวัดผลการทดลอง เช่น Accuracy, Precision, Recall และ F1-score เพื่อเปรียบเทียบประสิทธิภาพของแต่ละวิธีการตั้งค่า และเลือกวิธีที่ใช้ผลลัพธ์ที่ดีที่สุดในแง่ของความแม่นยำในการตรวจจับการฉ้อโกงการเตรียมข้อมูลเริ่มจากการจัดการกับ Missing Values และการทำ Standardization เพื่อให้ข้อมูลมีสเกลที่เหมาะสมสำหรับการเรียนรู้ของเครื่อง เนื่องจากข้อมูลมีความไม่สมดุล จึงใช้เทคนิคการสุ่มแบบ Oversampling ด้วยการ SMOTE เพื่อเพิ่มจำนวนข้อมูล ที่เป็นการฉ้อโกงในการทดลองผู้วิจัยทำการแบ่งชุด ข้อมูลเป็น 80:20 โดย 80 คือข้อมูลที่ใช้ในการฝึกฝน โมเดล ซึ่งต้องใช้ข้อมูลส่วนใหญ่ในการฝึกโมเดลเพื่อให้โมเดลเรียนรู้จากรูปแบบและข้อมูลที่มีอย่างเพียงพอ ต่อมาการทดสอบโมเดลส่วน 20% ถูกกันไว้เพื่อตรวจสอบความแม่นยำของโมเดลโดยไม่ได้ใช้ในการฝึกซึ่งจะช่วยให้การประเมินโมเดลสะท้อนถึงความสามารถของโมเดลในการทำนายข้อมูลใหม่ได้ดียิ่งขึ้นและในการทดสอบโมเดลแสดงผลลัพธ์ค่าความแม่นยำและ Confusion Metrix เพื่อวัดประสิทธิภาพ

5. Experimental Setup

ชุดข้อมูล: ดึงข้อมูลจาก Kaggle การเตรียมข้อมูลออกเป็นข้อมูลฝึก (Training set) และข้อมูลทดสอบ (Test set) 80:20 การเลือกฟีเจอร์ (Feature Selection): ใช้เทคนิค ANOVA, RFE ในการเลือกฟีเจอร์ที่สำคัญต่อการตรวจจับการฉ้อโกง การปรับสมดุลข้อมูลด้วย SMOTE (Synthetic Minority Over-sampling Technique): สำหรับการจัดการกับข้อมูลที่ไม่สมดุล โดยทำการสร้างตัวอย่างข้อมูลการฉ้อโกงเพิ่มขึ้นในลักษณะการสุ่ม เพื่อช่วยปรับสมดุลระหว่างข้อมูลปกติ การเลือกโมเดล และการตั้งค่าทดสอบโมเดลที่หลากหลาย เช่น Decision Tree, Random Forest, Logistic Regression, KNN, SVM, และ ANN โดยตั้งค่าพารามิเตอร์ของแต่ละโมเดลให้เหมาะสมกับชุดข้อมูลที่ใช้ เพื่อให้ได้ผลลัพธ์ที่มีความแม่นยำสูงสุด การประเมินผลการทดลอง: ใช้เมทริกซ์ต่างๆ ในการวัดผลการทดลอง เช่น Accuracy, Precision, Recall และ F1-score เพื่อเปรียบเทียบประสิทธิภาพของแต่ละวิธีการตั้งค่า และเลือกวิธีที่ให้ผลลัพธ์ที่ดีที่สุด ในแง่ของความแม่นยำ ในการตรวจจับการฉ้อโกงบัตรเครดิต การเตรียมข้อมูลเริ่มจากการจัดการกับ Missing Values และการทำ Standardization เพื่อให้ข้อมูลมีสเกลที่เหมาะสมสำหรับการเรียนรู้ของเครื่อง เนื่องจากข้อมูลมีความไม่สมดุลจึงใช้เทคนิคการสุ่มแบบ Oversampling ด้วยการ SMOTE เพื่อเพิ่มจำนวนข้อมูลที่เป็นการฉ้อโกงในการทำการทดลอง ผู้วิจัยทำการแบ่งชุดข้อมูลเป็น 80:20 โดย 80 คือข้อมูลที่ใช้ในการฝึกฝนโมเดลซึ่งต้องใช้ข้อมูลส่วนใหญ่ในการฝึกฝนโมเดล เพื่อให้โมเดลเรียนรู้จากรูปแบบ และข้อมูลที่มีอย่างเพียงพอ

การทดสอบโมเดลส่วน 20% ถูกกันไว้เพื่อตรวจสอบความแม่นยำของโมเดล โดยไม่ได้ใช้ในการฝึกซึ่งจะช่วยให้การประเมินโมเดลสะท้อนถึงความสามารถของโมเดลในการทำนายข้อมูลใหม่ได้ดียิ่งขึ้น และการทดสอบโมเดลที่แสดงผลลัพธ์ค่าความแม่นยำ และ Confusion Matrix เพื่อวัดประสิทธิภาพ

6. ผลการทดสอบประสิทธิภาพของระบบ

โดยคำนึงถึงค่าความแม่นยำ และวัดค่าประสิทธิภาพ Confusion Matrix ในเมธอดต่างๆ

6.1 การวัดประสิทธิภาพของโมเดลจากค่าความแม่นยำ (Precision, Recall, F1-Score)

	Accuracy	Precision	Recall	F1-score
SVM	0.9982795547909132	0.50	0.50	0.50
KNN	0.9984375548611355	1.00	0.55	0.58
Logistic Regression	0.9986306660580738	0.81	0.78	0.79
Decision Tree	0.9991748885221726	0.87	0.90	0.89
Random Forest	0.9995962220427653	0.99	0.89	0.93
ANN	0.9984024437344194	0.81	0.59	0.64
SVM + SMOTE	0.9956826055607337	1.00	1.00	1.00
KNN + SMOTE	0.9708597858009602	0.97	0.97	0.97
Logistic Regression + SMOTE	0.9732690853454795	0.97	0.97	0.97
Decision Tree + SMOTE	0.9984348346024656	1.00	1.00	1.00
Random Forest + SMOTE	0.9998681040395336	1.00	1.00	1.00
ANN + SMOTE	0.979696815152208	0.98	0.98	0.98
SVM + SMOTE + RFE	0.9884107416070204	0.99	0.99	0.99
KNN + SMOTE + RFE	0.9993317271336369	1.00	1.00	1.00
Logistic Regression + SMOTE + RFE	0.9786416474684768	0.98	0.98	0.98
Decision Tree + SMOTE + RFE	0.998188628809595	1.00	1.00	1.00
Random Forest + SMOTE + RFE	0.9998593109755025	1.00	1.00	1.00
ANN + SMOTE + RFE	0.9996131051826319	1.00	1.00	1.00
SVM + SMOTE + ANOVA	0.9895098746109069	0.99	0.99	0.99
KNN + SMOTE + ANOVA	0.9995163814782899	1.00	1.00	1.00
Logistic Regression + SMOTE + ANOVA	0.9780876844345181	0.98	0.98	0.98
Decision Tree + SMOTE + ANOVA	0.9982413871937815	1.00	1.00	1.00
Random Forest + SMOTE + ANOVA	0.9998856901675958	1.00	1.00	1.00
ANN + SMOTE + ANOVA	0.9995339676063522	1.00	1.00	1.00

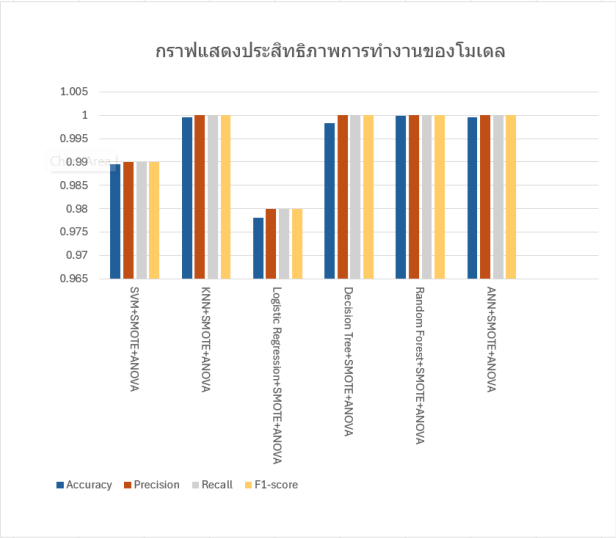
ตารางที่ 1 สรุปผลการทดสอบ

7. สรุปผลการวิจัย และข้อเสนอแนะ

7.1 สรุปผลการวิจัย

จาก **ตารางที่ 1** เมธอดที่เหมาะกับข้อมูลนี้ที่สุดคือ**เมธอดที่ 3** และทำ feature selection คือ ANOVA และทำการ SMOTE จากนั้นนำไปเข้าโมเดล Random Forest ได้ค่า Accuracy สูงสุด คือ Random

Forest + SMOTE + ANOVA เท่ากับ 0.9998 และค่า Precision เท่ากับ 1.00 ค่า Recall เท่ากับ 1.00 และค่า F1-Score เท่ากับ 1.00 เนื่องจากชุดข้อมูลมีความแตกต่างกันไป ผู้วิจัยควรเลือกเมธอดที่เหมาะสมกับข้อมูลของผู้วิจัยมากที่สุด ถ้าข้อมูลไม่สมดุลกันก็ควรเลือก**เมธอดที่ 3** แต่ถ้าข้อมูลมีความสมดุลกันก็สามารถเลือกใช้**เมธอดที่ 2** เพื่อลดขั้นตอนให้น้อยลง และจาก**ภาพที่ 5** ซึ่งแสดงการเปรียบเทียบแต่ละโมเดลจากการทำ**เมธอดที่ 3** สามารถสรุปได้ว่า โมเดลที่ดีที่สุด คือ Random Forest มีค่า Accuracy เท่ากับ 0.9998 ซึ่งมีค่าสูงกว่าโมเดลทั้งหมด



ภาพที่ 5 กราฟสรุปผลเมธอดที่ 3

7.2 ข้อเสนอแนะ

เนื่องจากโมเดลที่ผู้วิจัยใช้มีความเรียบง่ายสำหรับการศึกษาเบื้องต้น จึงสามารถนำไปประยุกต์ใช้ต่อยอดได้ โดยการทำการทดลองเพิ่มหรือเปลี่ยนโมเดลต่างๆ เพื่อผลลัพธ์ที่มีประสิทธิภาพมากยิ่งขึ้น ในอนาคตการศึกษานี้สามารถนำไปสร้าง และพัฒนาเป็นแอปพลิเคชัน หรือเว็บไซต์ โดยสามารถประเมินหรือ ตรวจสอบการฉ้อโกงได้ เพื่อหาวิธีป้องกันการฉ้อโกงในอนาคต และสามารถนำไปประยุกต์ใช้ในธุรกิจ หรือสถาบันธนาคารต่อไปได้

เอกสารอ้างอิง

[1] ธนกร, พ., ศรีสวัสดิ์, ว., & ศิริวรรณ, จ. (2564). การใช้การเรียนรู้เชิงลึกและการวิเคราะห์ฟีเจอร์เบื้องต้นในการตรวจจับการฉ้อโกงบัตรเครดิต. วารสารการจัดการเทคโนโลยีสารสนเทศ, 10(1), 35-50.

[2] ศิริกาญจน์, ส., ชัยยศ, ธ., & วรลักษณ์, อ. (2565). การใช้การเรียนรู้เชิงลึกแบบผสมในการตรวจจับการฉ้อโกงบัตรเครดิต. วารสารเทคโนโลยีสารสนเทศและนวัตกรรม, 15(1), 45-56.

[3] จิรวัฒน์, ภ., สมหมาย, ก., & พรศักดิ์, ม. (2566). การตรวจจับการฉ้อโกงโดยการวิเคราะห์ฟีเจอร์แบบ ANOVA และ XGBoost. วารสารวิทยาศาสตร์และเทคโนโลยี, 21(2), 100-113.

[4] มารีอา อูกนิช. (2565). “ตรวจจับการฉ้อโกงบัตรเครดิตด้วย Logistic Regression ใน Machine Learning.” [ออนไลน์]. สืบค้นวันที่ 3 พฤศจิกายน 2567. จาก https://devjourneys.com/2022/02/08/งานวจย-data-sci-report-ตรวจจบการฉอโกง/?path=thesis/it/0738/01title-illustrations.pdf

[5] เครือวัลย์ เนตรพนา และ ศิริสรรพ เหล่าหะเกียรติ. (2566). “การวิเคราะห์ความเสี่ยงในการผิดนัดชำระของลูกหนี้บัตรเครดิต โดยการใช้อัลกอริทึมการเรียนรู้ของเครื่อง.” [ออนไลน์]. สืบค้นวันที่ 3 พฤศจิกายน 2567. จาก https://msds.science.swu.ac.th/wp-content/uploads/2023/04/4_64199130036_Kruewan_Netphana_50_66.pdf

[6] scikit-learn developers. (2567). “Support Vector Machines (SVM).” [ออนไลน์]. สืบค้นวันที่ 3

พฤศจิกายน 2567. จาก https://scikit-learn.org/1.5/modules/svm.html

[7] scikit-learn developers. (2567). “sklearn.neighbors.KNeighborsClassifier.” [ออนไลน์]. สืบค้นวันที่ 3 พฤศจิกายน 2567. จาก https://scikit-learn.org/dev/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

[8] Scikit-Learn. “Logistic Regression.” [ออนไลน์]. สืบค้นวันที่ 3 พฤศจิกายน 2567. จาก https://scikit-learn.org/0.16/modules/generated/sklearn.linear_model.LogisticRegression.html

[9] scikit-learn developers. (2566). “OneVsRestClassifier.” [ออนไลน์]. สืบค้นวันที่ 3 พฤศจิกายน 2567. จาก https://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OneVsRestClassifier.html

[10] scikit-learn developers. (2567). “Neural Networks for Supervised Learning.” [ออนไลน์]. สืบค้นวันที่ 3 พฤศจิกายน 2567. จาก https://scikit-learn.org/1.5/modules/neural_networks_supervised.html

[11] scikit-learn developers. (2567). “Tree-based Models.” [ออนไลน์]. สืบค้นวันที่ 3 พฤศจิกายน 2567. จาก https://scikit-learn.org/1.5/modules/tree.html

[12] scikit-learn developers. (2567). “Random Forest Classifier.” [ออนไลน์]. สืบค้นวันที่ 3 พฤศจิกายน 2567. จาก https://scikit-learn.org/1.5/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[13] NovelBiz Co., Ltd. (2567). “การใช้งาน Google Colab.” [ออนไลน์]. สืบค้นวันที่ 3 พฤศจิกายน 2567. จาก https://www.novelbiz.co.th/google-colab/

[14] Chanchal24. (2566). “Credit Card Fraud Detection.” [ออนไลน์]. สืบค้นวันที่ 3 พฤศจิกายน 2567. จาก https://www.kaggle.com/code/chanchal24/credit-card-fraud-detection

[15] scikit-learn developers. (2567). “Feature Selection.” [ออนไลน์]. สืบค้นวันที่ 3 พฤศจิกายน 2567. จาก https://scikit-learn.org/1.5/modules/feature_selection.html

[16] GeeksforGeeks. “Handling Imbalanced Data with SMOTE and NearMiss Algorithm in Python.” [ออนไลน์]. สืบค้นวันที่ 20 กันยายน 2567. จาก https://www.geeksforgeeks.org/ml-handling-imbalanced-data-with-smote-and-near-miss-algorithm-in-python/

https://www.ibm.com/topics/confusion-matrix