

WEST UNIVERSITY OF TIMISOARA

DOCTORAL SCHOOL OF EXACT AND NATURAL SCIENCES



**DOCTORAL THESIS
ABSTRACT/REZUMAT**

SUPERVISOR:

Prof. Univ. Dr. Marc FRÎNCU

PHD. CANDIDATE:

Elena FLONDOR

2025

WEST UNIVERSITY OF TIMIȘOARA
DOCTORAL SCHOOL OF EXACT AND NATURAL SCIENCES
COMPUTER SCIENCE

Machine Learning Methods for Identifying Mislabeled Applications and Android App Classification

Supervisor:

Prof. Univ. Dr. Marc Frîncu

PhD Candidate:

Elena Flondor

Doctoral Advisory Committee:

Prof. Univ. Dr. Dana Petcu, West University of Timișoara

Prof. Univ. Dr. Daniela Zaharie, West University of Timișoara

Conf. Univ. Dr. Ciprian Pungilă, West University of Timișoara

A thesis submitted in partial fulfilment of the requirements for the degree of
Doctor of Computer Science

2025

Abstract

Over the past decade, Android’s rise as the leading mobile operating system has fueled massive application growth, with over 1.6 million on the Google Play Store. While its openness drives innovation, the sheer scale creates significant challenges in accurately classifying applications across categories. Accurate classification influences user experience and application discoverability, but mislabeled applications, which occur due to developer error or vague guidelines, can mislead users and skew analytics. Current systems rely heavily on developer input, making them unreliable and unscalable at today’s application volumes. This thesis addresses two interrelated challenges: (1) detecting misclassified applications in existing datasets and (2) automating application classification using enriched feature sets and machine learning (ML) models. Using Latent Dirichlet Allocation (LDA) on a dataset of top applications, initial experiments confirmed that even highly ranked applications are sometimes inconsistently categorized. To address this, two novel methodologies for misclassification detection were introduced. The first approach employed semantic similarity measures combined with hierarchical clustering to identify applications that deviate from category norms. This method suggested the presence of mislabeled applications in the Medical and Weather categories. However, it was computationally expensive at scale. The second method improved efficiency and coverage by computing average K-Nearest Neighbor (KNN) distances using BERT-based representation of application descriptions. Potential mislabeled applications were validated through HDBSCAN clustering and summaries of cluster semantics generated by Large Language Models (Llama). Experiments across multiple datasets confirmed this technique’s robustness, scalability, and interpretability. A significant contribution of the thesis is reversing the mislabeled detection process to identify highly representative applications for each category. This enabled the construction of well-aligned category benchmarks for developing and evaluating classification models. Building on this, the second significant contribution of the work focused on automating application and game classification using hybrid datasets enriched with static analysis (e.g., permissions, manifest components), dynamic features (e.g., GUI structure), and metadata from marketplaces (e.g., descriptions). Several ML models were evaluated, including traditional classifiers (Random Forest, Extreme Gradient Boosting) and a custom-designed multi-input Deep Neural Network (DNN). Experiments demonstrated substantial improvements over prior work. Random Forest achieved an average accuracy of 79% across multiple datasets, while Extreme Gradient Boosting improved classification quality when dynamic features were included. Using Llama-based embeddings of application descriptions significantly outperformed traditional text representations like TF-IDF, and the custom multi-input DNN model outperformed existing research: 94% accuracy, 88% precision, 90% recall, and an F1-score of 89%. These results highlight the strength of combining diverse feature types

with expressive data representations and ensemble learning strategies. Furthermore, the thesis validates the framework's generalizability by applying it to financial fraud detection, where adapted KNN-based outlier detection and ensemble models effectively identified fraudulent transactions, highlighting its flexibility and robustness beyond the Android domain. This thesis fills key research gaps and lays the groundwork for future advances, such as multi-label classification and behavioral analysis, ultimately supporting more reliable application marketplaces.

Rezumat

În ultimul deceniu, ascensiunea Android ca principal sistem de operare mobil a dus la o creștere masivă a numărului aplicațiilor, cu peste 1,6 milioane disponibile pe Google Play Store. Deși deschiderea platformei stimulează inovația, dimensiunea sa ridicată generează provocări semnificative în clasificarea corectă a aplicațiilor pe categorii. Clasificarea precisă influențează experiența utilizatorilor și descoperirea aplicațiilor, însă aplicațiile etichetate greșit, care apar din cauza erorilor dezvoltatorilor sau a unor ghiduri neclare, pot induce utilizatorii în eroare și pot denatura analizele. Sistemele actuale se bazează puternic pe inputul dezvoltatorilor, ceea ce le face nesigure și greu de scalat la volumele actuale de aplicații.

Această teză abordează două provocări interconectate: (1) detectarea aplicațiilor clasificate greșit în seturile de date existente și (2) automatizarea clasificării aplicațiilor folosind seturi de date îmbogățite și modele de învățare automată. Utilizând Latent Dirichlet Allocation (LDA) pe un set de date cu aplicațiile de top, experimentele inițiale au confirmat că și aplicațiile foarte bine cotate sunt uneori încadrate inconsistent. Pentru a remedia acest lucru, au fost introduse două metodologii noi de detectare a erorilor de clasificare. Prima abordare a utilizat măsuri de similaritate semantică combinate cu clustering ierarhic pentru a identifica aplicațiile care se abat de la normele categoriei. Această metodă a indicat prezența unor aplicații etichetate greșit în categoriile Medical și Weather. Totuși, s-a dovedit costisitoare din punct de vedere computațional. A doua metodă a îmbunătățit eficiența și acoperirea prin calcularea distanțelor medii K-Nearest Neighbor (KNN) folosind reprezentări bazate pe BERT ale descrierilor aplicațiilor. Aplicațiile potențial etichetate greșit au fost validate prin HDBSCAN clustering și generarea, folosind modele lingvistice de mari dimensiuni (Llama), a rezumatelor semantice ale clusterelor obținute. Experimentele pe mai multe seturi de date au confirmat robustețea, scalabilitatea și interpretabilitatea acestei tehnici. O contribuție semnificativă a tezei constă în inversarea procesului de detectare a erorilor de etichetare pentru a identifica aplicațiile foarte reprezentative pentru fiecare categorie. Acest lucru a permis determinarea unor seturi de aplicații bine aliniate pe categorii, utile pentru dezvoltarea și evaluarea modelelor de clasificare.

Pornind de aici, a doua contribuție majoră a lucrării s-a concentrat pe automatizarea clasificării aplicațiilor și jocurilor folosind seturi de date hibride, îmbogățite cu analiză statică (de exemplu, permisiuni, componente de manifest), caracteristici dinamice (de exemplu, structura GUI) și metadate din magazinele de aplicații (de exemplu, descrieri). Au fost evaluate mai multe modele de învățare automată, incluzând clasificatori tradiționali (Random Forest, Extreme Gradient Boosting) și o rețea neuronală cu multiple intrări, proiectată special. Experimentele au demonstrat îmbunătățiri semnificative față de lucrările anterioare. Random Forest a obținut o

acuratețe medie de 79% pe mai multe seturi de date, în timp ce Extreme Gradient Boosting a îmbunătățit calitatea clasificării atunci când au fost incluse caracteristicile dinamice. Utilizarea Llama pentru reprezentarea descrierilor aplicațiilor a depășit semnificativ reprezentările textuale tradiționale precum TF-IDF, iar rețeaua neuronală profundă cu multiple intrări a depășit rezultatele cercetărilor existente: 94% acuratețe, 88% precizie, 90% rată de regăsire și un scor F1 de 89%. Aceste rezultate evidențiază puterea combinării tipurilor de caracteristici diverse cu reprezentări expresive ale datelor și strategii de învățare de tip ansamblu.

În plus, teza validează generalizabilitatea cadrului propus prin aplicarea lui la detectarea fraudei financiare, unde adaptarea metodei KNN pentru detectarea anomaliilor și modelele de tip ansamblu au identificat eficient tranzacțiile frauduloase, demonstrând flexibilitatea și robustețea sa dincolo de domeniul Android. Această teză umple goluri cheie în cercetare și pune bazele unor cercetări viitoare, precum clasificarea cu etichete multiple și analiza comportamentală, ducând în final la magazine de aplicații mai fiabile.

Acknowledgments

I would like to sincerely thank my scientific advisor, Prof. Univ. Dr. Marc FRÎNCU, for his continuous guidance, support, and encouragement throughout my PhD journey and during the development of this dissertation. I'm genuinely grateful for his trust in me throughout this research, which allowed me to explore a research area I aim to pursue further in my career.

I also thank the members of the doctoral thesis guidance committee: Prof. Univ. Dr. Dana PETCU, Prof. Univ. Dr. Daniela ZAHARIE, and Conf. Univ. Dr. Ciprian PUNGILĂ, for their valuable insights and support.

A special acknowledgment goes to my family and boyfriend, whose unwavering support carried me through challenging moments and helped me remain committed to my goal. Their belief in my abilities and determination gave me the strength to complete this dissertation.

Finally, I offer my sincere gratitude to everyone who, directly or indirectly, contributed to the progress and completion of this scientific endeavor through their unconditional support.

Contents

Abstract	i
Acknowledgments	v
Contents	vi
List of Acronyms	xi
List of Figures	xiv
List of Tables	xvii
1 Introduction	1
1.1 Android Applications Mislabeling on Stores	4
1.2 Automated Classification of Android Apps	6
1.3 Thesis Motivation & Objectives	10
1.3.1 Thesis Motivation	10
1.3.2 Thesis Objectives	12
1.4 Results & Publications	15
1.5 Thesis Structure	16
2 Background	18
2.1 Android Operating System Overview	19
2.2 The Application Framework Unpacked	21
2.2.1 Application Organization	21
2.2.2 Intents	22
2.2.3 Application Packaging	22
2.2.4 Application Life Cycle	23
2.3 From Code to Users: The Publishing Process	24
2.4 Google Play Store Information Overview	27

2.5	Types of Android Applications Datasets	30
2.6	Extracting Insights from Android Applications	32
2.6.1	Strategies for Data Collection	32
2.6.2	Preprocessing Android Applications	34
2.6.3	Techniques for Feature Extraction & Selection	35
2.6.4	Techniques for Data Representation	37
2.7	Techniques Applied for Android Apps Classification	39
2.7.1	Types of Classification Techniques	39
2.7.2	Key Algorithms for Automated Classification	40
2.7.2.1	Clustering	40
2.7.2.2	Traditional Classification Algorithms	42
2.7.2.3	Boosting and Ensemble Learning Techniques	44
2.7.2.4	Bayesian Methods	45
2.7.2.5	Neural Network-Based Algorithms	45
2.7.3	K-Fold Cross-Validation	47
2.7.4	Imbalanced Dataset	47
2.7.5	Hyper-Parameter Tuning	48
2.7.6	Overfitting and Underfitting	48
2.7.7	Performance Evaluation Metrics	48
2.8	Chapter Conclusion	50
3	State-of-the-Art	51
3.1	Survey of Existing Datasets	52
3.1.1	A Summary of the Findings	59
3.2	Methodologies for Dataset Utilization	59
3.2.1	Data Collection Techniques in Previous Studies	59
3.2.2	Data Preprocessing, Feature Extraction and Data Transfor- mation in Previous Studies	61
3.2.3	A Summary of the Findings	67
3.3	Current Research on Misclassified Applications	70
3.4	Overview of Prior Categorization Approaches	70
3.4.1	Supervised Classification Approaches	71
3.4.2	Unsupervised Classification Approaches	77
3.4.3	Semi-supervised Classification Approaches	79
3.5	Limitations of Existing Work	79

3.6	Conclusion	80
4	Detection of Mislabeled Applications	82
4.1	Motivation	83
4.2	Data	84
4.3	Data Preprocessing Techniques	87
4.4	Categories Content Validation	87
4.4.1	Dataset Preprocessing	88
4.4.2	Topic Modeling with LDA	89
4.4.3	Experimental Setup	90
4.4.4	Evaluation Approach	91
4.4.5	Experiments & Analysis	92
4.4.5.1	Comparative Analysis of the Scenarios' Results	92
4.4.5.2	Analysis of the Best Model	94
4.5	Detecting Mislabels via Semantic Similarity	97
4.5.1	Dataset Preprocessing	98
4.5.2	Possible Outliers Identification	98
4.5.2.1	Semantic Similarity Measure	98
4.5.2.2	Hierarchical Clustering and Potential Outliers Identification	101
4.5.3	Experimental Setup	101
4.5.4	Experiments & Analysis	101
4.6	Detecting Mislabels via OOD Detection & LLMs	104
4.6.1	Data Preprocessing	104
4.6.2	Out-of-Distribution Detection	104
4.6.2.1	Data Representation	105
4.6.2.2	Out-of-Distribution Detection Algorithm	105
4.6.3	Outliers Validation	105
4.6.3.1	HDBSCAN Clustering	105
4.6.3.2	LLMs for Selected Outliers Validation	106
4.6.4	Experimental Setup	106
4.6.5	Experiments and Analysis	107
4.6.5.1	Dataset A	110
4.6.5.2	Dataset B	112
4.6.5.3	Dataset C	113

4.6.5.4	Dataset D	114
4.6.5.5	Dataset E	115
4.6.5.6	Selecting Highly Representative Applications Accord- ing to Categories Guidelines	121
4.7	Conclusion	125
5	Automated Classification of Applications	128
5.1	Motivation	129
5.2	Data	130
5.3	Methodology	131
5.3.1	Data Preprocessing & Representation	132
5.3.1.1	Prior Work	132
5.3.1.2	Proposed Data Preprocessing & Representation	132
5.3.2	Multi-View Features Selection	133
5.3.3	Classification Methods	134
5.3.3.1	Prior Work Classifiers	134
5.3.3.2	Proposed Classifiers	135
5.3.4	Experimental Setup	136
5.3.4.1	Experimental Setup for Games Classification	137
5.3.4.2	Experimental Setup for Applications Classification . .	138
5.3.4.3	Environment & Technologies	138
5.4	Games Classification Results & Analysis	139
5.4.1	Incipient Research	139
5.4.2	Prior Work Classifiers	142
5.4.3	Proposed Classifiers	143
5.4.4	Comparative Analysis	148
5.5	Applications Classification Results & Analysis	149
5.5.1	Prior Work Classifiers	149
5.5.2	Proposed Classifiers	152
5.5.3	Comparative Analysis	158
5.6	Conclusion	159
6	Generalizability of the Proposed Methdologies	161
6.1	Description of the Financial Transactions Dataset	162
6.2	Adaptation of Methodologies	164

6.2.1	Data Preprocessing and Representation	164
6.2.2	Detection of OOD Transactions	165
6.2.3	Classification Methods	165
6.3	Experimental Setup	165
6.4	Results & Analysis	166
6.4.1	OOD Detection Results & Analysis	166
6.4.2	Classifiers Results & Analysis	167
6.5	Discussion on Generalizability	168
6.6	Conclusion	169
7	Conclusion	170
7.1	Motivation & Objectives	171
7.2	Results	172
7.2.1	Detection of Mislabeled Applications	172
7.2.2	Automated Techniques for Mobile Applications Classification . .	173
7.2.3	Generalizability of the Proposed Methodologies	174
7.3	Open Issues and Future Work	175

List of Acronyms

AB AdaBoost.

AHC Agglomerative Hierarchical Clustering.

API Application Programming Interface.

ARFF Attribute-Relation File Format.

ASCII American Standard Code for Information Interchange.

Bayesian TAN Bayesian Tree-Augmented Naive Bayes.

BBN Bernoulli Naive Bayes.

BERT Bidirectional Encoder Representations from Transformers.

BIRCH Balanced Iterative Reducing and Clustering using Hierarchies.

BM25 Best Match 25.

BN Bayesian Network.

BoW Bag-of-Words.

CNN Convolutional Neural Network.

CPU Central Processing Unit.

CSV Comma Separated Values.

DBSCAN Density-Based Spatial Clustering of Applications with Noise.

DEX Dalvik Executable.

DL Deep Learning.

DNN Deep Neural Network.

DT Decision Tree.

ENN Ensemble Neural Network.

GNN Graph Neural Networks.

GPU Graphics Processing Unit.

GRU Gated Recurrent Unit.

GUI Graphic User Interface.

HDBSCAN Hierarchical Density-Based Spatial Clustering of Applications with Noise.

HTML Hypertext Markup Language.

JAR Java Archive.

KNN K-Nearest Neighbors.

LDA Latent Dirichlet Allocation.

Llama Large Language Model Meta AI.

LLM Large Language Model.

LR Logistic Regression.

LSTM Long-Short Term Memory.

MaxEnt Maximum Entropy.

MIME Multipurpose Internet Mail Extensions.

ML Machine Learning.

MNB Multinomial Naive Bayes.

MST Minimum Spanning Tree.

NB Naive Bayes.

NLP Natural Language Processing.

NMF Non-Negative Matrix Factorization.

NN Neural Network.

no number.

OOD Out-of-Distribution.

OS Operating System.

PLSI Probabilistic Latent Semantic Indexing.

RF Random Forest.

RNN Recurrent Neural Networks.

SDK Software Development Kit.

SGD Stochastic Gradient Descent.

SMOTE Synthetic Minority Oversampling Technique.

SMS Short Message Service.

SVM Support Vector Machines.

TF-IDF Term Frequency-Inverse Document Frequency.

UI User Interface.

UMAP Manifold Approximation and Projection.

URI Uniform Resource Identifier.

URL Uniform Resource Locator.

VSM Vector Space Model.

XGB Extreme Gradient Boosting.

List of Figures

1.1	iPhone vs. Android market share worldwide. The illustration is based on the data provided by Backlinko [59].	2
1.2	Number of available applications in the Google Play Store from 2009 to 2025. The illustration is based on the data provided by Statista [27].	3
1.3	Google Play Store categories interface. The images were collected from the Google Play Android application.	4
1.4	Mislabeled applications published on Google Play Store and detected as malicious samples by domain researchers. The sources of the images are Bitdefender Labs [32, 55] and Securelist [119].	7
1.5	The research process throughout the thesis development.	12
2.1	Android software platform.	21
4.1	The methodology proposed for dataset content validation against entities' guidelines.	88
4.2	Application description containing noise words for Flow Legends: Pipe Games [83].	89
4.3	Application description containing noise words for Netflix application [158].	89
4.4	Graphical representation of LDA.	90
4.5	Count of topics assigned a distinct category in every scenario.	93
4.6	Count of topics without labels in each scenario.	93
4.7	Outcomes of the labeling process for each scenario: the count of topics assigned a unique category (4.5), and the count of unlabeled topics per scenario (4.6).	93
4.8	The count of topics assigned to more than one category.	93
4.9	Visualization of the LDA clusters overlapping.	97
4.10	The methodology of misclassified samples detection through semantic similarity.	98
4.11	Histogram of the number of unique words in 100 samples of the Communication category.	103

4.12	Histogram of the number of unique words in the Events category. . . .	103
4.13	The methodology proposed for outliers detection using the OOD technique and LLMs.	104
4.14	Dataset A - Applications distribution in the benchmark of samples with the most representative applications.	122
4.15	Dataset B - Applications distribution in the benchmark of samples with the most representative applications.	122
4.16	Dataset C - Applications distribution in the benchmark of samples with the most representative applications.	123
4.17	Dataset D - Applications distribution in the benchmark of samples with the most representative applications.	123
4.18	Dataset E - Applications distribution in the benchmark of samples with the most representative applications.	124
4.19	Dataset A - Games distribution in the benchmark of samples with the most representative games.	124
4.20	Dataset E - Games distribution in the benchmark of samples with the most representative games.	125
5.1	The methodology proposed for detection of mislabeled applications through OOD and LLMs.	132
5.2	Architecture of the multi-input DNN proposed for applications classification by functionality.	136
5.3	XGB confusion matrix for early-stage game classification.	140
5.4	XGB classifier confusion matrix for games balanced subset in the incipient stage of the research.	141
5.5	XGB classifier confusion matrix for games categories of Dataset A. . . .	146
5.6	XGB classifier confusion matrix for games categories of subset E*. . . .	147
5.7	XGB classifier confusion matrix for games categories of subset E**. . .	148
5.8	Performance change from S1 to S2 for Bajaj et al.'s [25] classifier for applications classification.	150
5.9	Performance change from S1 to S2 for Guendouz et al.'s classifier [99] for applications classification.	151
5.10	Performance change from S1 to S2 for Kalaivani et al.'s classifier [118] for applications classification.	152
5.11	Confusion matrix of XGB classifier for subset E**.	157
5.12	Confusion matrix of multi-input DNN for subset E**.	158

5.13	Visual comparison between the results obtained for the classification methods of prior researchers (Section 5.5.1) and proposed classifiers (Section 5.5.2). The graphic presents the results obtained on subset E^* and the improvement obtained when features extracted through static and dynamic analysis are included.	159
6.1	Number of fraud victims by age.	163
6.2	Number of credit card frauds by month.	163
6.3	Number of credit card frauds by category.	164

List of Tables

2.1	The Google Play Store recommendations for applications categories' content. The source of these is the Google Play Console Help [48]. . . .	26
2.2	Market metadata: application details.	28
2.3	Market metadata: basic information.	28
2.4	Market metadata: localization information.	29
2.5	Market metadata: policy and legal information.	29
2.6	Market metadata: monetization and distribution.	29
2.7	Market metadata: application store performance metrics.	30
2.8	Market metadata: delivery details.	30
2.9	Market metadata: developer information.	30
2.10	Market metadata: additional metadata.	30
3.1	Summary of public datasets of Android applications used to automate classification tasks based on applications functionality.	53
3.2	Summary of the datasets based on market metadata used for applications classification by functionality.	55
3.3	Examples of proprietary datasets and their short description.	55
3.4	Summary of the datasets based on static features analysis used for applications classification by functionality.	57
3.5	Summary of the datasets based on hybrid analysis used for applications classification by functionality.	58
3.6	Table resuming the findings of the Section 3.2.1 and Section 3.2.2. . . .	69
3.7	Table showing the best results obtained by prior researchers for supervised classification approaches.	76
3.8	Table showing the best results obtained by prior researchers for unsupervised classification approaches.	79
4.1	Criteria of forming vocabularies in proposed scenarios.	91
4.2	Table presenting the evaluation metric values for each scenario, with the best overall performance (S1) highlighted in bold.	93

4.3	Scenario S5: cosine similarity and human interpretation category names assigned to unlabeled topics, and words in topics.	94
4.4	Categories assigned to topics after LDA prediction performed on categories' descriptions and examples in the case of S1 scenario.	95
4.5	Words with highest weights in labeled topics.	95
4.6	Words with highest weights in unlabeled topics.	96
4.7	Topic IDs assigned to unlabeled topics through cosine similarity and the category name assigned to topics based on human interpretation. . .	96
4.8	Time-related computational costs for the benchmark experiments. . . .	102
4.9	Results of the performance test for cluster summaries generation. . . .	109
4.10	Summary generated in case of Scenario 1.	109
4.11	Summary obtained in Scenario 9.	109
4.12	Results of validation approach for applications categories of Dataset A. .	111
4.13	Results of the validation approach for games in Dataset A.	111
4.14	Results of the validation approach for Dataset B.	113
4.15	Validation results for the benchmark of 9,256 applications of Dataset E. .	116
4.16	The results of the validation approach for Dataset E.	118
4.17	Summary of the results for possible outliers clustering with HDBSCAN for game categories of proprietary dataset.	120
5.1	Evaluation metrics values of classifiers using different embeddings. . . .	140
5.2	Time cost of classifiers using Word2Vec-based description representation during training process.	141
5.3	Game classification results for the classifier proposed by Bajaj et al. [25].	143
5.4	Game classification results for the proposal of Guendouz et al. [99]. . .	143
5.5	Game classification results for the proposal by Kalaivani et al. [118]. . .	143
5.6	Results of the proposed classifiers on game categories from Dataset A and Dataset E (with corresponding E* and E** subsets), combining Word2Vec-based description representation with the rest of the features. .	144
5.7	Results of the proposed classifiers on game categories from Dataset A and Dataset E (with corresponding E* and E** subsets) when combining Llama-based description representation with the rest of the features. .	145
5.8	Bajaj et al.'s classifier [25] results for applications classification.	150
5.9	Guendouz et al.'s classifier [99] results for applications classification. . .	151
5.10	Kalaivani et al.'s classifier [118] results for applications classification. . .	151
5.11	Results of the proposed classifiers for applications classification.	154
5.12	Results of the proposed classifiers for subsets E* and E**.	154

5.13	Classification report of the performance of the multi-input DNN for applications classification. The report shows the values of Precision, Recall, and F1-score for each category. Macro Avg. and Weighted Avg. are also specified.	155
5.14	Classification report of the performance of the XGB for applications classification. The report shows the Precision, Recall, and F1-score values for each category. Macro Avg. and Weighted Avg. are also specified.	156
6.1	Feature importance of the features selected for the classification process for the financial transactions dataset.	167
6.2	Comparison of classification metrics for DT, RF, and XGB for the classification of financial transactions.	168