ROYAUME DU MAROC
Université Mohammed V
- RABAT -
جامعة محمد الخامس
– الرباط –
Faculté des sciences
كلية العلوم
CENTRE D'ETUDES DOCTORALES - SCIENCES ET TECHNOLOGIES

N° d'ordre 3866

THESE

En vue de l'obtention du : ***DOCTORAT***

Structure de Recherche : Laboratoire Mathématiques, Informatique et Applications – Sécurité de l'Information
Discipline : Mathématiques Appliquées et Informatiques
Spécialité : Machine Learning.

Présentée et soutenue le : 16/11/2023 par :

Nadir Sahllal

Machine Learning Approaches for Imbalanced Data: Click-Through Rate Prediction and Click Fraud Detection

JURY

Khalid ZINE-DINE	PES, Université Mohammed V, Faculté des Sciences -Rabat	Président
Abdelkader BETARI	PH, Université Mohammed V,Institut Universitaire des Etudes Africaines - Rabat	Rapporteur/Examinateur
Ahmed Drissi EL MALIANI	PH, Université Mohammed V, Faculté des Sciences -Rabat	Rapporteur/Examinateur
Mounia MIKRAM	PH, Ecole des Sciences de l'information- Rabat	Rapporteur/Examinateur
Yacine GHAMRI-DOUDANE	PES, Université de la Rochelle	Examinateur
Dounia LOTFI	PH, Université Mohammed V, Faculté des Sciences -Rabat	Examinateur
El Mamoun SOUIDI	PES, Université Mohammed V, Faculté des Sciences -Rabat	Directeur de thèse

Année Universitaire : 2023-2024

Dédicaces

A ma très chère mère

Affable, honorable, aimable : Tu représentes
pour moi le symbole de la bonté par excellence,
la source de tendresse et l'exemple du dévouement.
Aucune dédicace ne saurait être assez éloquente pour
exprimer ce que tu mérites pour tous les sacrifices
que tu n'as cessé de me donner depuis ma naissance.
Tu as fait plus qu'une mère puisse faire pour que ses enfants
suivent le bon chemin dans leur vie et leurs études.
Je te dédie ce travail en témoignage
de mon profond amour.

A la mémoire de mon Père

Aucune dédicace ne saurait exprimer l'amour,l'estime
et le respect que j'ai toujours eu pour vous.
Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation
et mon bien être.
Ce travail est le fruit de tes sacrifices que tu as
consentis pour mon éducation et ma formation.

A mes très chère frères et soeurs

Aucun mot ne pourrait exprimer ma reconnaissance et ma gratitude
pour le soutien moral que vous n'avez cessé de me prodiguer

A mes amis et tous ceux qui m'aiment

Aussi nombreux que vous soyez je ne saurai vous citer
Je ne peux trouver les mots justes et sincères pour vous
exprimer mon affection et mes pensées

MERCI

Nadir.

Acknowledgment

The work in this Thesis was done in Laboratory of Mathematics, Computing and Applications - Information Security (LabMIA-SI), Faculty of Sciences, Mohammed V University in Rabat, under the supervision of Prof. El Mamoun Souidi.

First and foremost, I would like to thank Allah Almighty for giving me the strength, knowledge, ability and opportunity to undertake this research project and to persevere and complete it satisfactorily. Without his blessings, this achievement would not have been possible.

I would like to express my most sincere gratitude to my supervisor Prof. **El Mamoun SOUIDI** for the guidance, encouragement and advice she has provided during this long and rewarding journey. I have been extremely fortunate to work with a person who shared my passion in this work and has always answered my questions and paved my way for knowledge.

I would like to thank him for his precious efforts, professional and pedagogical orientations. I've learned so much from him when I was a Masters student and I still continue to learn as a research scientist.

Besides my supervisors, I would like to thank **Khalid ZINE-DINE**, Professor of Higher Education from the Faculty of Sciences in Rabat for to chair the jury for my thesis defense, and for agreeing to judge this work. I would like to express my gratitude to you and assure you of my deep respect.

I offer my thanks to **Abdelkader BETARI**, Habilitated Professor from The University institute of African studies having accepted to participate in this jury, and for agreeing to judge this work as an inspector. I want to express my gratitude to you and assure you of my deep respect.

I also offer my thanks to **Ahmed DRISSI EL MALIANI**, Habilitated Professor from the Faculty of Sciences in Rabat for agreeing to be a jury member of this defense and for evaluating this work as a reporter. I would like to express my deep gratitude and thanks to you.

I would like to thank **Yacine GHAMRI-DOUDANE**, Professor of Higher Education from La Rochelle University for agreeing to be a jury member of this defense and for examining this work. I would like to express my sincere appreciation and gratitude.

I offer my sincere thanks to **Dounia LOTFI**, Habilitated Professor from the Faculty of Sciences in Rabat for agreeing to be a jury member of this defense and for evaluating this work. I would like to express my sincere gratitude and thanks.

I also offer my thanks to **Mounia MIKRAM**, Habilitated Professor from the School of Information Science for agreeing to be a jury member of this defense and for evaluating this work as a reporter. I would like to express my deep gratitude and thanks

to you.

Of course, I would like to thank all the professors at the Laboratory of Mathematics, Computing and Applications - Information Security. And I would like to express my thanks to all the members and research colleagues of the laboratory for their coffee discussions and for the creation of a friendly working environment which brings out my best performance.

Many thanks to my family and friends who supported me during these years and helped me get through the struggle, stress and anxiety a typical PhD student deals with in Academia. Thank you for sharing this adventure with me.

Nadir.

Résumé

Cette thèse de recherche aborde les défis importants posés par les ensembles de données déséquilibrés dans le domaine de la publicité, en se concentrant sur la prédiction du taux de clics (CTR) et la détection de la fraude au clic. Elle explore l'impact du déséquilibre des données sur la prédiction du CTR, révélant comment cela affecte la précision prédictive et identifiant les caractéristiques clés essentielles pour la précision. L'étude se penche également sur le développement d'algorithmes avancés pour une détection optimisée de la fraude au clic, en employant des techniques d'apprentissage automatique comme XGBoost et des méthodes d'ensemble. Mettant l'accent sur la praticité et la reproductibilité, la recherche intègre un traitement approfondi des données et des méthodologies robustes adaptées aux applications du monde réel. Cette approche complète améliore non seulement la compréhension du déséquilibre des données dans la publicité, mais fournit également des solutions pratiques pour une prédiction efficace du CTR et la détection de la fraude au clic. La thèse se conclut avec des perspectives d'avenir, plaidant pour des recherches continues sur les considérations éthiques, le suivi des performances à long terme, la sélection avancée des caractéristiques et la détection de la fraude en temps réel, dans le but d'élargir l'applicabilité de ces résultats à d'autres domaines.

Mots-clés : Données déséquilibrées, Taux de clics (CTR), Fraude au clic, Publicité, Apprentissage automatique.

Résumé detailler

Cette thèse de recherche étudie des approches d'apprentissage automatique et leur capacité à résoudre les problèmes liés aux données déséquilibrées dans le secteur de la publicité, en mettant particulièrement l'accent sur deux domaines cruciaux: la prédiction du taux de clics (CTR) et la détection de la fraude au clic. Les objectifs fondamentaux de cette étude comprennent une exploration précise des implications du déséquilibre des données dans le contexte de la publicité, ainsi que l'observation et l'étude de deux ensembles de données majeurs dans le domaine, en l'occurrence la base de données produite par ReadPeak et le dataset TalkingData. De plus, basé sur l'étude et les caractéristiques des datasets mentionnés, on vise à développer des algorithmes de pointe visant à optimiser la précision de la prédiction du CTR et l'efficacité de la détection de la fraude au clic, ainsi que la validation de l'applicabilité pratique de ces méthodologies dans des scénarios réels.

L'industrie de la publicité, en tant que force motrice majeure de l'économie mondiale, voit des milliards de dollars circuler chaque année dans le domaine numérique. Au cœur de cette mécanique se trouve la prédiction précise du CTR, qui forge la capacité des annonceurs à prendre des décisions avisées, à canaliser leurs ressources de manière optimale, et à amplifier leur retour sur investissement (ROI). Toutefois, cette dynamique est constamment mise à l'épreuve par la fraude au clic, un fléau qui érode l'intégrité de la publicité en ligne. Cette fraude, incarnée par des interactions artificielles ou malintentionnées avec les annonces, menace non seulement de dilapider les budgets mais aussi d'obscurcir les mesures de performance. Cette industrie offre ainsi un aperçu flagrant des défis engendrés par un déséquilibre criant des données. Face à cette réalité, la nécessité d'instruments de prédiction du CTR rigoureux et d'outils de détection de fraude affûtés devient impérative. Les résultats et les algorithmes issus de cette recherche ont le potentiel d'aider les professionnels de cette industrie en fournissant des stratégies exploitables pour faire face à ces défis critiques.

Le déséquilibre des données, bien qu'omniprésent dans divers domaines, est régulièrement minimisé ou incompris, malgré son impact majeur sur les performances des modèles d'apprentissage. On donne un exemple dans le domaine étudié de la publicité. Disons que, sur 10 000 interactions avec une annonce, seuls 100 aboutissent à un clic, les 9 900 restants restant inactifs. Ici, ces clics vitaux pour évaluer le succès d'une campagne ne constituent qu'un mince 1% des interactions. Un modèle formé sur de telles données serait tenté de favoriser la classe dominante (non-clic), et si on utilise la précision comme mesure, cela sera 99% efficace même s'il ne prédit que des non-clics tout le temps.

Ce biais est exacerbé par un autre facteur: la fraude au clic. Ce phénomène, orchestré par des clics artificiels souvent générés par des bots ou des fermes de clics, vise à gonfler indûment les coûts pour les annonceurs ou à percevoir des gains illicites. Dans

des données où les clics légitimes sont extrêmement rare, discerner les clics frauduleux s'avère aussi difficile.

Pour résoudre les problèmes mentionnés, nous faisons appel à l'apprentissage automatique. Dans notre étude de l'état de l'art, nous identifions des lacunes critiques dans la littérature existante, notamment en ce qui concerne la clarté du traitement des données, les contraintes liées aux ressources informatiques et la nécessité d'évaluations rigoureuses de la faisabilité pratique. Ces lacunes identifiées nous poussent à accorder une importance majeure à la clarté du traitement des données et à évaluer nos solutions dans des environnements qui simulent la réalité

Le parcours des contributions scientifiques se déroule de manière méticuleuse, avec une approche méthodique qui offre une clarté et facilite la reproductibilité des algorithmes créés et des approches conduites sur les données publicitaires. On couvre l'ensemble du processus d'apprentissage automatique, de la collecte rigoureuse des données à l'élaboration et la mise en œuvre de modèles d'apprentissage automatique spécifiquement adaptés à la prédiction du CTR et à la détection de la fraude au clic. Des métriques d'évaluation telles que la précision, la précision, le rappel, le score F1 et l'aire sous la courbe ROC (AUC-ROC) sont utilisées pour évaluer les performances de ces modèles.

Le travail présenté met en avant plusieurs contributions majeures. Il introduit un ensemble de données conséquent fourni par ReadPeak, une entreprise de publicité finlandaise. Cet ensemble de données, dès le départ, présentait un déséquilibre notable entre les classes, avec un ratio significativement faible de 0,004.

Face à ce déséquilibre, notre approche a consisté en une étude comparative de 19 techniques d'échantillonnage citées dans les recherches précédentes comme solutions possibles à ce type de déséquilibre. nous avons également intégré des méthodes d'apprentissage automatique spécifiquement conçues pour traiter les problèmes de déséquilibre. En combinant ces techniques à des algorithmes de classification traditionnels, nous avons pu, grâce à des tests pratiques, déterminer la méthode la plus performante pour améliorer la prédiction du CTR.

Nos résultats sont révélateurs: parmi toutes les stratégies testées, c'est l'association du sous-échantillonnage aléatoire et de l'algorithme Random Forest qui s'est avérée la plus efficace pour prédire avec précision les clics. Cette découverte est cruciale pour les professionnels du domaine cherchant à optimiser leurs campagnes publicitaires.

En adoptant la même méthodologie, Une autre des contribution de ce travail est l'application de l'algorithme XGBoost à la prédiction de la fraude au clic. Cette contribution est appliquée au dataset TalkingData. Deux approches distinctes de l'utilisation de XGBoost sont présentées:

1. La première est axée sur l'optimisation de la détection de fraude. Grâce à l'optimisation des hyperparamètres de XGBoost, cette méthode a obtenu un AUC très compétitif par rapport aux autres études dans la littérature.

2. La deuxième approche, nommée "Growing and Pruning", est centrée sur la scalabilité et l'efficacité computationnelle de l'algorithme. Cette méthode améliore la complexité des calculs, bien qu'elle conduise à une légère baisse de la précision des prédictions. Néanmoins, les résultats demeurent compétitifs par rapport aux standards actuels.

Finalement, la dernière contribution abordée ici se rapporte à une stratégie avancée pour prédire la fraude au clic en utilisant l'apprentissage automatique. On propose une

méthode de classification innovante basée sur la technique d'empilement, ou "Stacking". Le concept repose sur une architecture à deux niveaux de prédiction: au premier niveau, divers classificateurs, tels que Random Forest, Decision Tree, AdaBoost et Naive Bayes, sont appliqués pour estimer la probabilité qu'un clic soit frauduleux. Ces estimations sont ensuite transmises au second niveau, où la régression logistique est utilisée pour affiner et consolider ces prédictions initiales en une prédiction finale. Ainsi, au lieu de se fier à la prédiction d'un seul modèle, cette approche capitalise sur la force combinée de plusieurs modèles, augmentant potentiellement la précision et la robustesse de la prédiction finale. Cette approche s'avère optimale en termes de précision pour la prédiction des clic frauduleux.

En synthèse, cette thèse sert de lien essentiel, reliant efficacement les avancées théoriques dans le domaine de l'apprentissage automatique à leur application pratique sur les données déséquilibrées, notamment dans le domaine complexe de la publicité. Les découvertes tirées de cette recherche représentent d'importantes contributions aux domaines de la prédiction du CTR et de la détection de la fraude au clic, en fournissant des solutions tangibles à des défis réels qui souffrent d'un déséquilibre extrême des données. De plus, cette étude ouvre plus de questions pour des futures recherches, encourageant l'exploration des considérations éthiques, de l'observation de performance à long terme, de l'adaptabilité des méthodes proposées à la constante évolution des défis émergents. En outre, elle ouvre la porte à des questions d'applicabilité des résultats présentés à d'autres domaines que la publicité.

Mots-clés : Données déséquilibrées, Taux de clics (CTR), Fraude au clic, Publicité, Apprentissage automatique.

Abstract

This research dissertation tackles the significant challenges of unbalanced datasets in advertising, focusing on Click-Through Rate (CTR) prediction and click fraud detection. It explores the impact of data imbalance on CTR prediction, revealing how it affects predictive accuracy and identifying key features crucial for precision. The study also delves into advanced algorithm development for optimized click fraud detection, employing machine learning techniques like XGBoost and ensemble methods. Emphasizing practicality and reproducibility, the research incorporates thorough data treatment and robust methodologies suitable for real-world applications. This comprehensive approach not only enhances the understanding of data imbalance in advertising but also provides practical solutions for effective CTR prediction and click fraud detection. The dissertation concludes with future perspectives, advocating for ongoing research in ethical considerations, long-term performance, advanced feature selection, and real-time fraud detection, aiming to broaden the applicability of these findings to other domains.

Keywords : Unbalanced data, Click-through rate (CTR), Click fraud, Advertising,Machine learning.

Detailed Abstract

This research dissertation tackles the significant challenges of unbalanced datasets in advertising, focusing on Click-Through Rate (CTR) prediction and click fraud detection. It explores the impact of data imbalance on CTR prediction, revealing how it affects predictive accuracy and identifying key features crucial for precision. in the following we will give a detailed resume of each part of this thesis.

Introduction

This research dissertation addresses challenges posed by unbalanced datasets in the advertising sector, particularly focusing on click-through rate (CTR) prediction and click fraud detection. The primary objectives are:

1. **Understanding data imbalance in advertising**:
   - ◇ Investigating the effect of data imbalance on advertising data, specifically on CTR prediction.
   - ◇ Analyzing challenges brought by unbalanced datasets and exploring optimal sampling strategies to mitigate these issues.
2. **Developing advanced algorithms**:
   - ◇ Designing cutting-edge algorithms, leveraging machine learning techniques to optimize information and prediction precision in CTR prediction and click fraud detection.
   - ◇ These algorithms aim to handle unbalanced data challenges, enhancing prediction accuracy.
3. **Ensuring practicality**:
   - ◇ Emphasizing the real-world applicability of the developed methodologies.
   - ◇ Conducting experiments in realistic settings, using industry-standard datasets, and prioritizing scalable algorithms suitable for real-time environments.

The research offers significant contributions in two main areas:

a. **Data imbalance and CTR prediction**:
   - ◇ Delving deep into the impacts of different data sampling techniques on native advertising data.

- ◇ Examining 19 class imbalance methods using four classical classifiers.
- ◇ Discovering the role of data balancing in CTR prediction and how performance varies with imbalance ratios.

b. **Click fraud detection with advanced ML techniques**:

- ◇ Utilizing advanced machine learning algorithms, like XGBoost, to detect fraudulent clicks effectively.
- ◇ Introducing ensemble methods, harnessing the combined power of various algorithms for enhanced fraud detection accuracy.

In conclusion, the study seeks to bridge the gap between theoretical advancements and their real-world applications in advertising, ensuring accurate CTR predictions and effective click fraud detection. It also points towards potential future research avenues, hoping to inspire further investigations in the domain.

Chapter 1: Preliminaries and Literature Review

In this chapter, we review prior research on unbalanced data, with a focus on its implications in CTR prediction and click fraud detection, particularly in advertising. We highlight the inherent challenges of class imbalance, where one class has significantly fewer samples, leading to biased classifiers. Researchers have devised techniques and evaluation criteria to address these issues, forming a critical foundation for our work.

Imbalanced data techniques We discuss techniques for dealing with class imbalance in machine learning in this section. Class imbalance can lead to biased model outputs favoring the majority class, particularly impacting tasks like click-fraud detection and CTR predictions in advertising.

Three key techniques are highlighted:

Sampling techniques: These alter the training data to improve class distribution. Undersampling reduces the majority class, while oversampling generates new minority class samples. Hybrid methods combine both.

Algorithm level methods: These adjust classifiers, often through weighting or cost assignment, to mitigate class imbalance effects.

Ensemble methods: These internally combine algorithms and sampling to address class imbalance. Techniques like Balanced Bagging and Balanced Random Forest aim to prevent favoring the majority class.

These techniques are widely used in machine learning to enhance classification performance across various domains, including advertising and fraud detection.

Understanding online advertising and the Real-Time Bidding (RTB) model: Online advertising, driven by the Real-Time Bidding (RTB) model, has revolutionized marketing through real-time auctions for ad impressions. In milliseconds, advertisers bid for ad space upon a user's visit to a webpage or app, allowing precise targeting, cost optimization, and timely ad delivery. However, RTB also introduces challenges like click fraud and accurate CTR prediction, which are crucial for campaign success and integrity.

Predicting CTR for effective online advertising: Predicting CTR is of paramount importance in online advertising, where CTR represents the ratio of ad clicks

to ad impressions. Accurate CTR prediction is essential for advertisers to assess ad performance, allocate resources wisely, and optimize campaigns effectively. It enables advertisers to estimate expected ad performance, make data-driven decisions before launching ads, and compare predicted and actual CTRs during and after campaigns for adjustments. Additionally, precise CTR prediction aids in optimizing resource allocation, ensuring budget, ad placement, and targeting strategies are directed towards the most promising opportunities. Advanced models, including factorization machines, neural networks, and interest-based segmentation, have significantly enhanced CTR prediction, enabling advertisers to maximize ROI through data-driven decision-making and personalized ad delivery.

Detecting and mitigating click fraud in online advertising: Detecting and mitigating click fraud is crucial in safeguarding the integrity of online advertising campaigns. Click fraud involves fraudulent or malicious clicks on ads, often driven by financial incentives, to inflate ad costs, deplete budgets, and distort performance metrics. It can be perpetrated by humans or automated click bots, with the latter being more prevalent. Different types of click fraud, such as badvertising, hit shaving, botnet click campaigns, and hit inflation, employ various deceptive strategies to manipulate click data and deceive advertising systems.

Click fraud has detrimental effects on advertisers, leading to wasted ad spend and distorted performance metrics. Advertisers pay for non-genuine engagement, making resource allocation inefficient, and fraudulent clicks inflate key metrics, making it challenging to assess campaign effectiveness accurately. Click fraud detection relies on anomaly detection techniques, statistical modeling, machine learning algorithms, and user behavior profiling. These approaches help identify abnormal click patterns that deviate from expected behavior, distinguishing fraudulent clicks from legitimate interactions.

Machine learning solutions have been explored extensively for click fraud detection, including ensemble learning, deep learning, and various supervised learning models. These models aim to improve the accuracy and effectiveness of click fraud detection, providing advertisers with valuable insights to protect their budgets and optimize campaign performance. Ongoing research continues to address the evolving challenges posed by click fraud in online advertising.

Gaps in the literature of CTR prediction and click fraud: Extensive research in CTR prediction and click fraud detection reveals critical gaps. Many studies lack transparency in data processing, hindering reproducibility. Limited computational resources often lead to small data subsets, affecting representativeness. The complexity and overlap in CTR and click fraud data challenge accurate modeling and distinction between genuine and fraudulent behavior.

Moreover, practicality and real-world applicability need rigorous evaluation. Novel algorithms are proposed, but their feasibility and effectiveness in practical scenarios remain untested. Comprehensive assessments in realistic settings are crucial for scalability and actual performance measurement. Addressing these gaps will enhance the reliability and applicability of CTR prediction and click fraud detection methods.

Chapter 2: Methodology

In this chapter, we present our methodology for examining the effects of data sampling techniques on native advertising data. To establish a strong foundation for our study, we commence with an extensive literature review, which informs our research design. This review provides valuable insights into existing approaches and techniques within

the domains of CTR prediction and click fraud detection. It helps us identify the most relevant methods and algorithms to apply in our investigation, ensuring that our research is grounded in established knowledge.

The subsequent phase of our methodology focuses on data collection, preprocessing, and feature engineering. We collect relevant data pertaining to ad impressions, clicks, user behavior, and contextual information from appropriate sources. This raw data is meticulously processed to eliminate noise, address missing values, and convert it into suitable formats for analysis. Feature engineering techniques are then applied to extract meaningful features that capture underlying patterns and relationships within the data. This meticulous preparation stage is essential to ensure the reliability and suitability of the input data, ultimately leading to robust and credible results in combating click fraud and enhancing advertising campaign performance.

Following data preparation, our research design incorporates the development and implementation of various machine learning models. These models include logistic regression, factorization machines, neural networks for CTR prediction, and anomaly detection techniques, supervised learning algorithms, or user behavior profiling methods for click fraud detection, depending on the specific context. These models are trained on historical data to learn the intricate relationships between ad features, user information, and the likelihood of clicks or fraudulent activities. The evaluation of these models is a critical aspect of our methodology, involving metrics such as accuracy, precision, recall, F1-score, and the area under the receiver operating characteristic curve (AUC-ROC). These metrics serve as the yardstick for assessing the models' performance on both training and testing datasets, ensuring their effectiveness in capturing patterns and detecting fraudulent activities. Our research design is versatile and can be adapted to various CTR prediction and click fraud detection scenarios, allowing us to systematically address the challenges in these domains and derive practical insights to optimize feature importance and prediction accuracy.

Chapter 3: Effects of Data Sampling Techniques on Native Advertising Data

In this chapter, we explore class imbalance techniques for predicting advertisement CTR using a dataset from ReadPeak in Finland. The dataset initially suffered from a severe class imbalance, with a ratio of 0.004 between majority and minority classes. To address this, we delve into a comprehensive discussion of 19 imbalanced learning methods, including undersampling, oversampling, hybrid resampling, and ensemble techniques. These methods aim to balance dataset categories by adjusting majority or minority samples through various strategies. We combine these techniques with classic classifiers like logistic regression, random forest, decision tree, and naive Bayes for training and preprocessing, setting a baseline for comparison using AUC as a key performance metric.

We also explore different imbalance levels and their impact on feature importance and AUC improvement. Undersampling, particularly the Random Under-Sampling (RUS) technique within the Random Forest model, stands out as a successful approach.

This research demonstrates the practical use of sampling techniques in handling highly skewed data, such as native advertising data. It has broader applications in domains with similar data consistency challenges, like cybersecurity and intrusion detection.

Chapter 4: Forecasting Click Fraud via Machine Learning Algorithms

In this chapter, we introduce a click fraud prediction classification approach utilizing the XGBoost algorithm, contributing significantly to the click fraud detection field. The research effectively employed feature engineering and preprocessing techniques to extract valuable insights from the data. The approach's optimization involved incremental training and pruning methods, enhancing its performance, space complexity, and computational efficiency.

Furthermore, a thorough comparison was conducted with classical machine learning models, including Random Forest, Logistic Regression, Decision Tree, and traditional XGBoost. While traditional XGBoost exhibited superior performance, the proposed approach still demonstrated competitive results. Notably, the adaptability and scalability of the proposed approach offer valuable advantages for industry practitioners.

In addition, we have introduced an innovative growing and pruning approach that significantly broadens the scope and practicality of our work. This approach not only enhances the adaptability of our click fraud prediction classification method but also enables its seamless integration into various real-world applications. By implementing this growing and pruning strategy, we provide practitioners with a powerful tool to dynamically adjust and refine the model as new data becomes available, ensuring its relevance and effectiveness over time.

Chapter 5: Mastering Click Fraud Detection: Uniting Algorithms in a Stacked Prediction Framework

In this chapter, we present our scientific contributions. We have introduced a click fraud prediction classification approach centered around the Stacking algorithm. To effectively address the challenges posed by unbalanced datasets, we've incorporated under-sampling and the SMOTE technique while utilizing the Boruta algorithm for feature selection. Our research leverages the TalkData dataset for comprehensive analysis, yielding the following key contributions:

Feature Engineering and Preprocessing: We emphasize the critical role of feature engineering and preprocessing in maximizing the utility of available data. Through these processes, we've created nine additional features from the existing dataset. These engineered features have been instrumental in enhancing the performance of our prediction algorithm, a fact substantiated by the feature selection outcomes.

Effective Feature Selection: The Boruta algorithm plays a pivotal role in our study by aiding in feature selection. This step is crucial in the realm of machine learning. Our experimental results in Section 4.1 align with the Boruta algorithm's feature selection, highlighting its effectiveness. Using the Random Forest classifier, we've assessed the importance of 14 features and conducted experiments by progressively inputting these features into the model based on their importance values. This rigorous evaluation confirms that many of the engineered features hold significant importance.

Stacking-Based Classification: Our work introduces a novel click fraud prediction classification approach grounded in the Stacking technique. We've employed Random Forest, Logistic Regression, Decision Tree, and AdaBoost as base learners, with Logistic

Regression serving as the second-level learner. To assess the individual contributions of the four first-level learners, we've trained the dataset separately with each learner. The results underscore the superiority of the Stacking approach, with Random Forest closely following in terms of performance.

Conclusions & Perspectives:

This dissertation tackled the challenge of highly imbalanced data in advertising, aiming to shed light on its implications and propose the following solutions.

CTR Prediction Analysis: We analyzed data imbalance in CTR prediction, revealing its impact on predictive accuracy and identifying key features. We also found that combining Random Undersampling (RUS) with Random Forest improved accuracy.

Enhanced Click Fraud Detection: We optimized an XGBoost model for click fraud detection, developed a realistic application approach, and introduced an effective stacking algorithm.

Practicality and Reproducibility: We emphasized practicality and reproducibility by rigorous data treatment and documentation.

Future Perspectives: Future research should focus on ethical considerations, long-term performance monitoring, addressing emerging challenges, advanced feature selection, ensemble techniques, real-time fraud detection, and generalizing findings to other domains.

Keywords : Unbalanced data, Click-through rate (CTR), Click fraud, Advertising,Machine learning.

List of Figures

List of Tables

Contents

Introduction

The field of machine learning and applied mathematics has witnessed remarkable advancements in recent years, revolutionizing various domains by uncovering hidden patterns, making accurate predictions, and enabling data-driven decision-making. However, a prevalent challenge persists in many real-world datasets: the issue of unbalanced data. Unbalanced data refers to datasets where the distribution of instances across different classes is heavily skewed, with one class dominating the others. This poses significant difficulties for traditional machine learning algorithms, which are typically biased toward the majority class, resulting in suboptimal predictions for minority classes.

Within the vast landscape of applications, the domain of advertising stands out as an area where unbalanced data is particularly prevalent and demands specialized attention. Advertisers invest substantial resources in online advertising campaigns, aiming to maximize their return on investment by effectively reaching their target audience and encouraging user engagement. Click-through rate (CTR) prediction plays a pivotal role in achieving this goal, as it involves estimating the likelihood of a user clicking on a specific advertisement. Accurate CTR prediction allows advertisers to optimize their campaigns, allocate their budget wisely, and enhance overall campaign performance.

However, the practicality of accurate CTR prediction is hindered by the imbalanced nature of real-world advertising datasets. The occurrence of positive instances, such as clicks, is significantly outnumbered by negative instances, i.e., non-clicks. Consequently, the class imbalance issue poses a formidable obstacle to achieving accurate CTR predictions. Conventional models often fail to capture the intricacies of the minority class, resulting in biased and suboptimal predictions. Consequently, there is a pressing need to develop innovative methodologies that effectively handle the class imbalance issue in CTR prediction, enabling advertisers to make informed decisions and optimize their campaigns more effectively.

Furthermore, the presence of click fraud poses an additional challenge in the advertising domain. Click fraud refers to malicious activities that artificially inflate click counts or generate fraudulent ad impressions, thereby compromising the integrity of online advertising campaigns. This pervasive issue leads to financial losses for advertisers and undermines user trust in the advertising ecosystem. Detecting and mitigating click fraud activities is crucial for maintaining fair and transparent advertising practices.

Against this backdrop, this research project aims to address the practical challenges associated with unbalanced data in the domains of CTR prediction and click fraud detection within the advertising industry. By leveraging advanced machine learning techniques and mathematical approaches, this research seeks to develop innovative methodologies that effectively handle the class imbalance issue, thereby improving the accuracy of CTR

predictions and enhancing click fraud detection capabilities. The outcomes of this research have the potential to significantly benefit advertisers by providing them with reliable tools and methodologies to optimize their campaigns, maximize their return on investment, and combat the detrimental effects of click fraud activities.

Motivation

The motivation behind this research project stems from the practical significance of solving prediction problems on unbalanced data and the critical need to improve the general accuracy of click-through rate (CTR) prediction and click fraud detection within the advertising domain.

Prediction problems on unbalanced data present a significant challenge in machine learning. Many real-world scenarios, such as medical diagnosis, fraud detection, and anomaly detection, involve imbalanced datasets where the occurrences of positive instances are substantially lower than negative instances. Achieving accurate predictions for the minority class is crucial for making informed decisions and mitigating risks in these domains. This research project is motivated by the opportunity to contribute to the development of methodologies that enhance prediction accuracy, particularly in critical areas where unbalanced data poses unique challenges.

Within the advertising industry, accurate CTR prediction plays a pivotal role in optimizing advertising campaigns. Advertisers invest substantial resources to create compelling advertisements and target specific audiences to maximize user engagement. Accurate prediction of CTR enables advertisers to make informed decisions regarding ad placements, budget allocation, and campaign strategies, leading to higher click-through rates and conversion rates. By improving the accuracy of CTR prediction, this research aims to empower advertisers with reliable predictive models that enhance campaign performance and increase return on investment.

Another key motivation for this research project is the detection of click fraud. Click fraud presents a significant threat to the integrity of online advertising campaigns. It involves malicious activities such as artificially inflating click counts or generating fraudulent ad impressions, resulting in financial losses for advertisers and eroding user trust in the advertising ecosystem. Developing robust click fraud detection models is essential to identify and mitigate fraudulent activities effectively. This research project is driven by the goal of enhancing the accuracy of click fraud detection, thereby safeguarding advertisers' investments and promoting fair and transparent advertising practices.

By addressing prediction problems on unbalanced data and improving the general accuracy of CTR prediction and click fraud detection, this research project aims to contribute to the advancement of predictive analytics in the field of machine learning and applied mathematics. The outcomes of this research will provide advertisers with reliable tools and methodologies to optimize their campaigns, maximize performance, and combat the detrimental effects of click fraud activities. Ultimately, this research seeks to enhance decision-making capabilities, foster fair advertising practices, and contribute to the broader field of predictive analytics.

Objective

The primary objective of this dissertation is to investigate, analyze, and improve statistical and machine learning methods, as well as sampling techniques, for effectively handling extremely unbalanced data, particularly within the advertising field. The presence of highly imbalanced datasets in advertising poses unique challenges that hinder the accuracy and reliability of predictive models. In this research, we aim to delve into the study of these challenges and develop innovative methodologies to enhance the prediction accuracy, optimize sampling strategies, and address the intricacies of unbalanced data in advertising scenarios. By focusing on this critical area of research, we aspire to contribute to the advancement of statistical and machine learning techniques and foster more effective and precise decision-making in the realm of advertising.

Study the Effect of Data Imbalance on Advertising Data and Optimal Sampling Strategies:

The first objective of this research is to investigate the impact of data imbalance on advertising data, with a specific focus on click-through rate (CTR) prediction. We aim to thoroughly analyze the challenges posed by imbalanced datasets within the advertising domain and understand how class imbalance affects the accuracy and performance of CTR prediction models. Additionally, we will explore various sampling strategies to address the data imbalance issue and identify the optimal approaches that yield the best results for CTR prediction. By studying the effect of data imbalance and identifying effective sampling strategies, we aim to enhance the accuracy and reliability of CTR prediction in advertising campaigns.

Optimize Information and Prediction Precision with State-of-the-Art Algorithms:

The second objective of this research is to develop state-of-the-art algorithms that optimize the amount of information and prediction precision in CTR prediction and click fraud detection. We will leverage advanced machine learning techniques and mathematical methodologies to create innovative algorithms that go beyond conventional approaches. These algorithms will be specifically tailored to handle the challenges posed by unbalanced data and improve the accuracy of CTR prediction and click fraud detection. By developing cutting-edge algorithms, we aim to enhance the overall prediction performance and provide advertisers with more reliable and precise insights for campaign optimization and click fraud mitigation.

Emphasize Practicality and Realistic Testing for Industrial Use:

The third objective of this research is to prioritize the practicality and real-world applicability of our solutions. We will focus on conducting experiments and evaluations in realistic settings, ensuring that our methodologies and algorithms are tested on industry-standard datasets and scenarios. Moreover, we will emphasize the development of prac-

tical solutions that can be readily implemented in industrial settings. In particular, we will explore the concept of grow and prune algorithms, which are designed to adapt and scale efficiently in real-time production environments. By emphasizing practicality and industrial relevance, we aim to bridge the gap between theoretical advancements and practical adoption, enabling advertisers to benefit from our solutions in real-world advertising campaigns.

By achieving these three objectives, this research project aims to significantly contribute to the field of CTR prediction and click fraud detection on unbalanced data within the advertising domain. Through a comprehensive study of data imbalance effects, development of state-of-the-art algorithms, and focus on practicality and realistic testing, we aspire to advance the understanding, accuracy, and applicability of predictive models in advertising.

Contributions and Organization

After presenting an introductory Chapter 1 that delves into the current landscape of data sampling systems, CTR (Click-Through Rate) prediction, and click fraud, shedding light on their significance and influence in the advertising industry, this Thesis aims to make notable contributions in the field of advertisement, aligning with the aforementioned objectives. These contributions are categorized into two interconnected areas, corresponding to the two main sections of this document: (i) Examining the Influence of Data Imbalance on CTR Prediction: Exploring Sampling Methods, and (ii) Advancing Click Fraud Detection via Sophisticated Machine Learning Algorithms.

Unveiling the Impact of Data Imbalance on CTR Prediction: Investigating Sampling Methods

This part focuses on understanding the effects of data imbalance on predicting CTR and explores various sampling methods to mitigate these challenges.

1. **Effects of Data Sampling Techniques on Native Advertising Data:** we dive into the topic of unbalanced data treatment techniques, specifically focusing on extremely unbalanced data such as click advertising data. We thoroughly explore the impact of various data sampling techniques on native advertising data and evaluate the performance of 19 class imbalance methods using four classical classifiers. The evaluation is based on the AUC and LogLoss metrics, allowing us to identify the most effective methods. Moreover, we conduct an in-depth analysis of the unbalanced data problem using real-world data from ReadPeak. We outline the necessary steps for feature engineering, selection, and data cleaning. Through our experimentation, we uncover that the importance of features varies with different Imbalance Ratios, and we utilize the Boruta algorithm to compute feature importance at three different levels of sampling. This emphasizes the crucial role of data balancing for achieving accurate results. By employing Random Forest and random undersampling, we demonstrate that improving the balance between positive and negative samples significantly enhances performance. However, we also observe that beyond a certain threshold of imbalance ratio, the improvement in performance becomes

marginal. Additionally, we contribute to the research community by providing access to real-world datasets, facilitating further study of CTR Prediction in the native advertisement domain and the exploration of potential solutions [72].

Enhancing Click Fraud Detection Through Advanced Machine Learning Algorithms

1. **Forecasting Click Fraud via Machine Learning Algorithms:** This chapter focuses on click fraud detection, utilizing the knowledge acquired from the previous chapter on data sampling techniques and CTR prediction. Our goal is to develop an effective solution for identifying and combating click fraud in online advertising campaigns. We accomplish this by leveraging machine learning algorithms, specifically the XGBoost model. To optimize performance, we employ grid search to fine-tune hyperparameters, maximizing the model's ability to accurately detect fraudulent clicks [73]. Additionally, we introduce a novel approach of growing and pruning the XGBoost model to strike a balance between complexity and predictive power. By showcasing experimental results, we highlight the effectiveness of our proposed approaches, demonstrating the potential of advanced algorithms in combatting click fraud and enhancing online advertising security.

2. **Click Fraud Detection: Uniting Algorithms in a Stacked Prediction Framework:** In this particular chapter, we embark on a comprehensive exploration of click fraud detection and its association with ensemble methods. Our main objective is to harness the power of a stacked prediction framework, which combines the predictive capabilities of multiple algorithms to establish a robust defense against click fraud. By capitalizing on the diverse strengths of various algorithms and leveraging their collective insights, we aim to significantly enhance the accuracy and efficacy of click fraud detection. Throughout this chapter, we meticulously examine the intricacies of ensemble methods, shedding light on their inherent advantages and illustrating how they can substantially elevate the accuracy levels of click fraud prediction models [71].

In the concluding chapter of this Thesis, we provide a comprehensive summary of the results obtained from our research. We highlight the key findings and insights gained throughout the study, emphasizing the contributions made to the field of advertisement, data sampling systems, CTR prediction, click fraud detection, and ensemble methods. Building upon these achievements, we also propose several open research problems and future directions that hold potential for further investigation and advancement. These research areas address gaps in current knowledge and present opportunities for innovation, thereby guiding future researchers to explore new avenues and make significant contributions to the field. By presenting these open research problems and future directions, we aim to inspire and facilitate continued progress in the study of advertisement and related domains.

CHAPTER 1

Preliminaries and Literature Review

In this chapter, we aim to provide a comprehensive overview of the existing body of work on unbalanced data, with a specific focus on its applications in click-through rate (CTR) prediction and click fraud detection. We will explore the aspiring research efforts that have been made to address the challenges posed by data imbalance in various domains. Additionally, we will delve into the specific research conducted in the context of CTR prediction and click fraud detection within the advertising field. By examining the advancements and insights gained from prior research, we aim to build a solid foundation for our own contributions and highlight the gaps that our work intends to address. This chapter serves as a precursor to the subsequent sections, where we will present our methodologies, experiments, and findings in further detail.

In a class imbalanced dataset, there are significantly less samples in one of its classes than the other [22]. The difficulties of learning from such imbalanced data are inevitable. Standard learning classifiers are biased toward the majority class due to the skewed distribution of the training samples, making them unable to recognize unusual occurrences. It is possible to mistake noise for rare minority samples and vise-versa [29]. This kind of problem with uneven data is particularly prevalent in the advertising industry. The dataset contains a lot more non-clicked advertising than clicked advertisements, and the difference between the two is typically significant. To solve these issues, researchers have created numerous class imbalance techniques and performance evaluation criteria that play an important role in this paper.

1.1 Imbalanced Data Techniques

Class imbalance is a well-known problem in machine learning [26]. When the classes in the data have an unbalanced distribution, machine learning model will favor samples from the majority class and will not pay enough attention to samples from the minority class. It will result in the model's output being biased in favor of the dominant class [12]. Due to the classifier's disregard for minority classes, its accuracy cannot be trusted. Many academics in the field of machine learning are now focusing on class unbalanced learning

due to the impacts and potential that class imbalances can have on data and learning.

In advertising, class imbalance methods have already been used in many applications, such as click-fraud detection and CTR predictions. Class imbalanced data methods can be classified into three categories: (i) sampling techniques, (ii) algorithm level methods and (iii) ensemble methods [37].

1.1.1 Sampling techniques

Data-level approaches entail steps taken in the training data to improve the class distribution by having less samples in the majority classes or more samples in the minority classes [28]. The data-level technique is primarily at the data pre-processing stage, and redistributing the training data of various classes in the data space through resampling [53]. To balance the unbalanced class, this kind of approach can alter the dataset structure as much as possible. Resampling the data to change the samples analog distribution has been demonstrated in some research to improve the model's performance to some extent [48].

Procedures for resampling can be divided further into (i) undersampling, (ii) oversampling and (iii) hybrid sampling. In the following, we provide a brief explanation of these techniques.

Undersampling techniques retain important data for learning while discarding samples from the majority class until the number of samples in each class is approximately equal [94]. However, it is unavoidable that certain samples that are significant to the training model may be overlooked while undersampling the dataset [36]. After all, many undersampling techniques employ various filtering principles. Undersampling methods include:

1. *Random UnderSampling (RUS)*: *RUS* is the earliest undersampling method to have been invented, and it discards random samples from the majority class [64].

2. *Edited Nearest Neighbors (ENN)*: With the other samples in this procedure, each instance is checked using k-NN. The improperly identified samples will be discarded, and the updated dataset will be created from the remaining samples [90].

3. *Instance Hardness Threshold (IHT)*: This undersampling technique excludes examples that are difficult to classify or have a high likelihood of being misclassified after training a classifier to identify them [79].

4. *Near Miss*: By choosing majority samples whose average distances from the three nearest minority samples are the least, this method chooses majority samples that are close to some minority samples [56].

5. *Neighbourhood Cleaning Rule (NCR)*: The three closest neighbors of each instance in the dataset are taken into account by this strategy. A sample is eliminated from the dataset if it is misclassified by its three closest neighbors and is in the majority class. Additionally, the majority class samples in the vicinity of a sample that is a member of the minority class sample and is incorrectly classified by its three closest neighbors are eliminated [47].

6. *One-Sided Selection (OSS)*: First, 1-NN is used to choose minority class samples and majority samples that were incorrectly classified. Then the Tomek Links' vast majority of class examples are deleted [45].

7. Tomek Links (TL): Refer to a pair of instances, which belong to different classes and are each other's nearest neighbor. These links are considered to be noisy or boundary instances and are often removed from the majority class sample [26].

To provide a more equal class distribution of samples while preserving class borders, oversampling techniques create new samples based on data from the minority class. On the other hand, because it duplicates or synthesizes a small number of samples, oversampling can result in overfitting [97]. The length of training time rises with the quantity of samples. Oversampling methods include:

1. *Random OverSampling (ROS)*: *ROS* is the earliest oversampling approach to be created, and it replicates random minority class samples to achieve a more evenly distributed class [10].

2. *Adaptive Synthetic (ADASYN)*: Minority class samples are distributed in this manner based on their difficulty to learn. Minority samples that are harder to learn have more synthetic samples created for them. [31].

3. *Synthetic Minority OverSampling Technique (SMOTE)*:The k-Nearest Neighbors (kNN) interpolation method is used to generate synthetic samples from each minority sample. [16].

4. *Borderline SMOTE*: This technique utilizes borderline samples, which are frequently misclassified by their closest neighbors, to perform SMOTE [30].

The hybrid sampling approach combines under-sampling and over-sampling. Both approaches have their drawbacks: under-sampling can result in loss of important information, while over-sampling can lead to overfitting. To address these issues, methods that combine under-sampling and over-sampling have been proposed, such as SMOTEENN and SMOTETomek [10]. These methods use a combination of techniques such as SMOTE for over-sampling and ENN or Tomek links for under-sampling. These techniques aim to balance the training dataset, eliminate noisy points that are on the incorrect side of the decision boundary, discover better clusters, and build models with strong generalization capabilities.

1.1.2 Algorithm Level Methods

Algorithm-level methods are approaches in which the classifier is not affected by the skewed distribution, or where conventional machine learning classifiers are modified and linked to a weight or cost variable [21]. Many researchers have published relevant studies addressing the class imbalance issue at the algorithm level [55, 89].

1.1.3 Ensemble Methods

In ensemble systems, algorithmic and sampling techniques are used to handle data internally and modify the distribution of categories in the sample. They employ data-level approaches, and internally modify the learning process using specific algorithms [3]. This helps to prevent the model from excessively favoring the majority class during classification [26]. The standard ensemble techniques are listed below:

1. *Balanced Bagging*: This technique employs bagging and RUS to balance the dataset. It uses integrated estimators after resampling each subgroup of the data. The technique has an advantage over Sci-kit-Learn because it uses two additional parameters to control the behavior of the random sampler, namely sampling strategy and replacement [97].

2. *Balanced Random Forest*: This technique creates a balanced sample from which each tree is constructed. It starts by drawing bootstrap samples from the minority class, and then randomly selects an equal number of instances from the majority class with replacement. The final prediction is determined by a majority vote [4].

3. *Easy Ensemble*: This approach involves using AdaBoost to train classifiers on balanced subsets, and then combining the output of each classifier to produce an ensemble classifier [52].

4. *Random UnderSampling Boost (RUSBoost)*: This method combines sampling and boosting, and each round of boosting includes RUS [76].

5. *Balance Cascade* : This approach uses a double integration technique that combines bagging and boosting. It removes majority class samples that can be accurately identified during training using an iterative method and combines them with the minority class to create a base learner. This approach emphasizes samples that are more likely to be misclassified [52].

1.1.4 Uses of Sampling Techniques to Improve Machine Learning Models

Sampling techniques are commonly employed in machine learning to address challenges associated with large datasets. These techniques aim to improve model training, reduce computation time, and handle imbalanced datasets. This literature review examines the applications of sampling methods in improving machine learning algorithms, highlighting their effectiveness in various contexts. Additionally, notable studies that have utilized different sampling techniques to enhance machine learning model performance are discussed.

Various studies have utilized sampling techniques to improve machine learning predictions. Ramos et al. [67] investigated the effects of sampling techniques in combination with feature selection, concluding that using Random Under-Sampling (RUS) before feature selection yields better results, while Random Over-Sampling (ROS) and Synthetic Minority Over-Sampling Technique (SMOTE) offer improved outcomes when applied afterwards.

In a study by Ussatova et al. [88], different sampling techniques were employed to enhance the precision of comprehensive DDoS Attack Classification using Machine Learning Algorithms. They found that oversampling using SMOTE resulted in improved accuracy for their approaches.

Ahssan et al. [5] examined the impact of resampling techniques on the performance of their proposed voting classifier. They achieved the best F1 score when combining the voting classifier with SMOTE. These results indicate that resampling techniques can provide sufficient variation to improve classification performance across different classifiers.

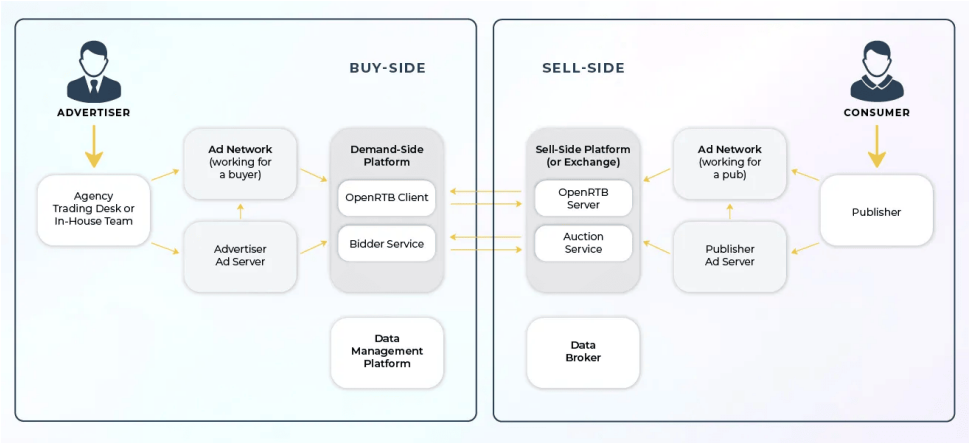


Figure 1.1: Advertisement ecosystem

Lopez-Martin et al. [54] conducted a comprehensive analysis of oversampling techniques, including SMOTE and adaptive synthetic sampling, on the NSL-KDD dataset. Their study introduced a novel generative variational autoencoder oversampling technique, which outperformed other methods when combined with various classifiers. The highest accuracy of 79.26% and an F1 score of 76.45% were achieved using MLP with the proposed oversampling technique.

Furthermore, Kim et al. [42] employed undersampling as the main sampling method in their work. They conducted experiments using real commercial data from an advertising platform to validate the improvement of click-through rate (CTR) prediction by incorporating segment interest.

In the context of credit card fraud prediction, Sasank et al. [74] explored the optimization of machine learning algorithms and sampling techniques to enhance metrics and results. They aimed to identify the best combination of algorithms and sampling methods to improve the prediction of credit card fraud.

1.2 Understanding Online Advertising and the RTB Model

Online advertising has revolutionized the way businesses promote their products and services, leveraging the vast reach and targeting capabilities of the internet. It involves the display of ads on websites, mobile apps, social media platforms, and search engines to capture the attention of potential customers. One of the key models used in online advertising is the Real-Time Bidding (RTB) model, which enables advertisers to bid for ad impressions in real-time auctions.

In the RTB model, an auction takes place whenever a user visits a webpage or an app where ad space is available, figure 1.1. The auction occurs within milliseconds, allowing advertisers to bid on the opportunity to display their ads to that particular user. Advertisers submit their bids and relevant ad creatives to an ad exchange, which conducts the auction and determines the winning bidder. The winning ad is then displayed to the user in the available ad space.

This dynamic bidding process offers several advantages for advertisers. First, it allows for precise targeting, as advertisers can bid based on specific user attributes, such as demographics, location, browsing behavior, and interests. Second, it enables advertisers

to optimize their ad spend by setting bid prices based on the perceived value of each impression. Lastly, the real-time nature of the RTB model ensures that ads are delivered to users at the right time and in the right context, increasing the chances of user engagement.

While the RTB model provides significant opportunities for advertisers, it also presents challenges in terms of click fraud and predicting Click-Through Rate (CTR) accurately. Click fraud can undermine the integrity of the bidding process, as fraudulent clicks can drive up costs and skew performance metrics. Moreover, accurately predicting CTR is crucial for advertisers to gauge the effectiveness of their ads and make informed decisions about campaign optimization. Therefore, addressing these challenges is paramount to ensure the success and integrity of online advertising campaigns.

In the subsequent sections, we will delve deeper into the intricacies of CTR prediction and click fraud detection, exploring methodologies, techniques, and advancements that can enhance the efficiency and effectiveness of online advertising. By understanding the complexities of CTR prediction and combating click fraud, advertisers can maximize their ROI, minimize wasteful ad spend, and deliver impactful ad experiences to their target audience.

1.3 Predicting CTR for Effective Online Advertising

CTR is a critical metric in online advertising that measures the ratio of ad clicks to the number of ad impressions. Accurate CTR prediction is crucial for advertisers to evaluate the performance of their ads, optimize their campaigns, and allocate their resources effectively. In this section, we will delve into the intricacies of CTR prediction, exploring methodologies, techniques, and advancements that enhance the accuracy and reliability of CTR estimation.

1.3.1 Importance of CTR Prediction

1.3.1.1 Performance Evaluation

Performance evaluation in digital advertising often involves various metrics, and one of the widely used ones is the CTR. It is calculated as the ratio of the number of clicks (Clicks) to the number of impressions (Impressions) an ad receives:

$$CTR = \frac{\text{Clicks}}{\text{Impressions}}$$

It serves as a powerful metric for assessing the relevance and appeal of ads to the target audience. A high CTR suggests that the ad is effectively capturing the attention of users and motivating them to take action, indicating a successful campaign.

Accurate CTR prediction plays a significant role in enabling advertisers to evaluate the success of their campaigns and make data-driven decisions. By accurately predicting CTR, advertisers can gain insights into the expected performance of their ads before launching them. This prediction allows them to estimate the potential reach and impact of their campaigns, helping them allocate resources effectively. Additionally, accurate

CTR prediction enables advertisers to compare the expected performance of different ad creatives, placements, or targeting strategies, allowing them to optimize their campaigns for maximum effectiveness.

Furthermore, accurate CTR prediction facilitates the measurement of ad performance during and after a campaign. By comparing the predicted CTR with the actual CTR, advertisers can assess the success of their ads in achieving their intended goals. If the actual CTR falls significantly below the predicted CTR, it may indicate a need for adjustments in the ad creative, targeting parameters, or overall campaign strategy. On the other hand, if the actual CTR exceeds the predicted CTR, it signifies a positive response from the audience and can inform future decision-making to replicate and build upon that success.

1.3.1.2 Resource Allocation

Resource allocation is a critical aspect of advertising campaigns, and accurate prediction of CTR plays a vital role in optimizing resource allocation. CTR prediction empowers advertisers to make informed decisions regarding budget allocation and ad placement, enabling them to maximize their return on investment (ROI) effectively.

Precise CTR estimation allows advertisers to allocate their resources, such as their advertising budget, strategically. By predicting the CTR for different ad creatives, placements, or targeting strategies, advertisers can identify the most effective options. They can then allocate a larger portion of their budget to the ads with higher predicted CTRs, ensuring that resources are allocated where they are likely to generate the best results.

Moreover, CTR prediction aids advertisers in optimizing ad placement. By accurately estimating the CTR for various ad placements or platforms, advertisers can determine which options are most likely to yield higher engagement and click-through rates. They can then prioritize those placements, ensuring that their ads reach the right audience at the right time, thereby increasing the chances of generating desired actions and conversions.

In addition to budget allocation and ad placement, precise CTR estimation also helps advertisers optimize their overall campaign strategy. By predicting the CTR for different targeting parameters or audience segments, advertisers can identify the most effective approaches. This allows them to focus their efforts on segments that are likely to have a higher CTR, ensuring that their message reaches the most receptive audience and increasing the likelihood of achieving their campaign objectives.

By leveraging CTR prediction to optimize resource allocation, advertisers can maximize their ROI. They can allocate their resources effectively, ensuring that their budget is directed towards the most promising opportunities. This data-driven approach enables advertisers to make informed decisions based on predicted performance, increasing the efficiency and effectiveness of their advertising efforts.

1.3.1.3 Established Works

Logistic Regression (LR) is a traditional model used for Click-Through Rate (CTR) prediction [57]. It offers advantages such as speed, simplicity, and interpretability. However, to increase the nonlinear expressiveness and reduce computational complexity,

several enhanced models have been proposed. Factorization machines [69], field-aware factorization machines [38], and gradient boosting Decision Trees are among the models that have demonstrated effectiveness in practice.

In recent years, neural network techniques have also been employed to further enhance CTR prediction models. Instead of relying on pooling interest vectors, newer approaches focus on extracting user interests directly from time-sequence data. For instance, the Recurrent Neural Network (RNN) model has been used in GRU4Rec [32] to forecast user preferences based on their previous click-through patterns. Another method, known as the Attentive Capsule Network (ACN) [49], has been developed to capture users' diverse interests. ACN employs a transformer to separate feature interactions from the variations in users' interests, thereby improving prediction accuracy.

To further boost performance, targeted segmentation and customized ad delivery to specific segments have proven to be beneficial [25, 51, 96]. Interest-based segmentation is one of the most widely used methods, grouping users with similar interests together [75]. By understanding the interests and preferences of different segments, advertisers can tailor their advertisements to better resonate with each group, increasing the likelihood of click-throughs and conversions.

These advancements in CTR prediction models, incorporating techniques such as factorization machines, neural networks, and interest-based segmentation, have significantly improved the accuracy and effectiveness of resource allocation in digital advertising campaigns. Advertisers can leverage these models to make data-driven decisions, optimize budget allocation, and deliver personalized ads to specific target segments, ultimately maximizing their ROI.

1.4 Detecting and Mitigating Click Fraud in Online Advertising

Click fraud poses a significant threat to the integrity and effectiveness of online advertising campaigns. It involves fraudulent or malicious clicks on ads with the intention of inflating ad costs, depleting advertising budgets, and distorting performance metrics. In this section, we will delve into the complexities of click fraud, examining methodologies, techniques, and advancements that aid in the detection and mitigation of this fraudulent activity.

1.4.1 Understanding Click Fraud

Click fraud, also known as malicious clicks or click spam, is a common form of fraud in the digital advertising ecosystem [63]. It is closely associated with the cost-per-click pricing model used to determine the costs of displaying ads to users on various platforms. Click fraud is often carried out to target competitors and inflate their advertising expenses. This fraudulent activity involves fraudsters making HTTP requests to the destination URLs of displayed ads. It can be performed either by human clickers or click bots, each exhibiting distinct characteristics.

Human clickers, driven by financial incentives, tend to rapidly click on multiple ads within a short timeframe. In contrast, genuine users typically engage with content, read, think, and explore a website before making a purchase [93]. On the other hand,

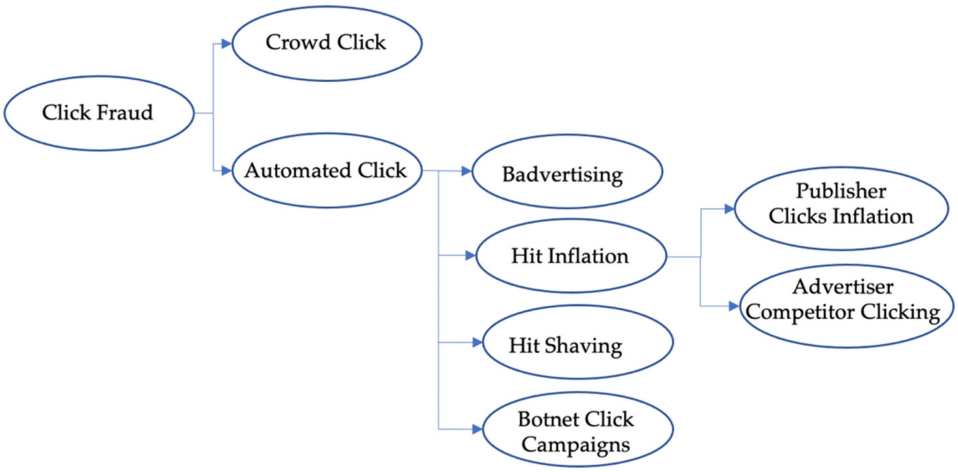


Figure 1.2: Types of Click Fraud

automated click bots generate fraudulent traffic from a relatively small number of computers. Detecting and differentiating between normal traffic generated by real users and fraudulent traffic generated by machines can be challenging. Methods used to identify automated click fraud may not be as effective in detecting human-generated fraud. While crowd click fraud, which involves exploiting a group of real people to increase fraudulent traffic, exists, it is less prevalent compared to click bots and is not the main focus of this study [63].

It is worth noting that automated scripts simulating human clicks are more cost-effective for hackers than hiring human laborers to engage in click fraud. Therefore, the prevalence of click bots surpasses that of crowd click fraud, making them a primary concern for the industry and research community.

1.4.1.1 Types of Click Fraud

Automated click fraud attacks can be broadly categorized into four distinct groups, namely badvertising, hit shaving, hit inflation, and botnet click campaigns [63, 85]. These types of attacks represent different malicious strategies employed by fraudsters to manipulate click data and deceive online advertising systems. In figure 1.2 and in the following we describe the different kinds of click fraud.

- ◇ **Badvertising**: This type of attack involves fraudulent publishers deliberately displaying ads on low-quality or irrelevant websites to generate illegitimate clicks [27]. The aim is to trick advertisers into paying for ad impressions that are unlikely to generate genuine user engagement or conversions.
- ◇ **Hit Shaving**: In hit shaving attacks [58], fraudsters attempt to manipulate click counts by selectively discarding or suppressing a portion of legitimate clicks. By artificially reducing the number of recorded clicks, fraudsters can create an illusion of lower click-through rates, potentially leading to reduced payouts for publishers or increased costs for advertisers.
- ◇ **Botnet Click Campaigns**: Botnet click campaigns involve the use of a network of compromised devices or bots to generate a large volume of fraudulent clicks [85].

These botnets can mimic real user behavior, making it challenging to differentiate between legitimate and fraudulent clicks. Fraudsters may utilize botnets to inflate click counts, exhaust advertisers' budgets, or disrupt competition.

- ◇ **Hit Inflation**: Hit inflation attacks work in the opposite manner of hit shaving [20]. Here, fraudsters generate fake or non-human clicks to artificially inflate the click counts. This deceptive tactic aims to make ad campaigns appear more successful by boosting the click-through rates, potentially leading to higher payouts for publishers or increased costs for advertisers.

By understanding the different types of automated click fraud attacks, advertisers and advertising platforms can develop robust detection and prevention mechanisms to safeguard against these fraudulent activities. In the following sections, we will explore various techniques and approaches employed in click fraud detection and mitigation, aiming to ensure the integrity and effectiveness of online advertising campaigns.

1.4.1.2 Impact on Advertisers

Click fraud has a significant impact on advertisers, leading to several adverse effects that can undermine their advertising efforts and diminish their return on investment (ROI). One of the primary consequences is the wastage of ad spend. Advertisers allocate a significant portion of their budget to display ads and pay for each click they receive. However, fraudulent clicks generated by bots or human clickers result in a substantial portion of the ad spend being wasted on non-genuine engagement. Advertisers end up paying for clicks that do not represent actual user interest or potential conversions, leading to inefficient resource allocation.

Another detrimental effect of click fraud is the distortion of performance metrics. Key metrics like Click-Through Rate (CTR), Conversion Rate (CR), and Cost per Acquisition (CPA) are essential for advertisers to evaluate the success and effectiveness of their campaigns. However, click fraud artificially inflates these metrics, making it difficult for advertisers to obtain accurate insights into their campaign performance. The presence of fraudulent clicks skews the CTR and CR, making it challenging to gauge the true engagement and conversion rates. Additionally, the CPA calculations become unreliable, as the costs are spread across both genuine and fraudulent clicks.

Accurate estimation of the impact of click fraud is essential for advertisers to make informed decisions and optimize their campaigns. While it is difficult to precisely measure the extent of click fraud, some researchers and industry experts have developed metrics to estimate the level of fraudulent activity. One such metric is the Click Fraud Rate (CFR), which calculates the percentage of fraudulent clicks out of the total clicks received. Although the exact equation for CFR may vary depending on the methodology used, a simplified formula can be expressed as:

$$CFR = \frac{\text{Number of Fraudulent Clicks}}{\text{Total Clicks}} \times 100\%$$

Having a clear understanding of the impact of click fraud allows advertisers to take necessary measures to protect their advertising budgets, optimize their campaigns, and seek more accurate insights into their performance. By addressing click fraud effectively,

advertisers can ensure that their resources are allocated wisely and their campaigns achieve the desired results, thereby maximizing their return on investment.

1.4.1.3 Click Fraud Detection

The identification of abnormal click patterns is crucial in detecting click fraud, and anomaly detection techniques play a vital role in this process. Anomaly detection involves identifying instances or patterns that deviate significantly from the expected or normal behavior [14]. In the context of click fraud, it helps distinguish fraudulent clicks from legitimate user interactions.

Statistical modeling approaches can be employed to detect anomalies in click patterns. These methods involve analyzing various statistical features such as click rates, click timestamps, IP addresses, and device information. Deviations from expected distributions or patterns can indicate potential click fraud [43]. For example, if a specific IP address generates an unusually high number of clicks within a short period or exhibits inconsistent behavior compared to other users, it could be flagged as an anomaly.

Machine learning algorithms are also extensively utilized in click fraud detection. Supervised learning algorithms can be trained on labeled datasets containing examples of both fraudulent and legitimate clicks. The models learn the patterns and characteristics associated with each category and can subsequently classify new instances as either fraudulent or legitimate. Unsupervised learning techniques, such as clustering or density-based methods, can also be used to identify anomalies without the need for labeled data. These algorithms can group clicks into clusters and identify outliers that differ significantly from the majority of the data, indicating potential fraudulent activity [80].

User behavior profiling is another effective approach for detecting click fraud. By establishing profiles of legitimate user behavior, advertisers can identify deviations that may indicate fraudulent activity. This can involve analyzing various features such as browsing history, session duration, click frequency, and interaction patterns. Anomalous behavior that deviates significantly from the established profiles can be indicative of click fraud [24].

1.4.2 Advancements and Emerging Approaches

In this section, we will explore the related literature on solving the issue of click fraud, with a specific focus on machine learning solutions. While it is important to note that this dissertation primarily focuses on sampling techniques and their combination with machine learning models to solve two advertising prolems namely ctr and click fraud prediction, we will concentrate our literature review on the broader field of machine learning solutions in this context.

Attempts to address click fraud using machine learning techniques include the work by [62] presented an ensemble learning approach for click fraud detection, combining several machine learning techniques, such as logistic regression, random forests, and support vector machines, to create a more robust model capable of detecting click fraud.

Deep learning has also been explored for click fraud detection, with [84] proposing a deep neural network architecture to learn complex patterns and relationships in the input data. Their deep learning model outperformed traditional machine learning techniques in detecting fraudulent clicks, suggesting that deep learning could be a promising direction for future research in this area.

[60], Clicktok proposes a protection approach that utilizes the temporal aspects of click traffic and unifies the technical response to separate organic and non-organic click fraud attacks. The author of online click requests was identified using AdSherlock, and click fraud was detected afterward, as stated by [9].

Various supervised learning models have been developed for detecting click fraud in online advertising. [39] proposes an ad fraud detection approach that utilizes robust features to prevent attackers from evading detection.

[40] developed new features based on statistics seen in an ad network, estimated from a large number of legitimate user ad requests, including the popularity of publisher websites and client environment tendencies ([7]). Another study by [86] conducted a study in which they developed a time-series model incorporating multiple scales to predict and detect click fraud behavior in real-time. These are only a few instances of the numerous approaches suggested to tackle click fraud in online advertising. The range of proposed models and methods highlights the significance of ongoing research in this field.

1.5 Gaps in the Literature of CTR Prediction and Click Fraud

While extensive research has been conducted on CTR prediction and click fraud detection, several significant gaps and limitations exist within the current literature. This section aims to highlight these gaps and shed light on areas where further research and improvement are needed.

1. **Lack of Reproducibility:** Many studies in the field of CTR prediction and click fraud detection lack sufficient details regarding their data treatment processes. This lack of transparency makes it challenging for other researchers to reproduce and validate the results. Furthermore, some studies only utilize a small subset of the available data due to limited computational resources, raising concerns about the representativeness of their findings. The unbalanced nature of the CTR and click fraud data adds to the complexity, making it crucial to address these limitations and ensure the reproducibility of research outcomes.

2. **Complexity and Overlapping Data:** The data related to CTR and click fraud is inherently complex, often exhibiting significant overlap between legitimate and fraudulent activities. This complexity poses a challenge in accurately modeling and analyzing the data, as well as distinguishing between genuine user behavior and fraudulent clicks. The existing literature needs to address these complexities by developing robust methodologies that can effectively handle overlapping data and provide accurate predictions and fraud detection.

3. **Practicality and Real-World Testing:** Another gap in the literature is the practicality and applicability of the proposed models and techniques. While many studies propose novel algorithms and methods, their feasibility and effectiveness in real-world scenarios are often not thoroughly evaluated. Rigorous testing in practical environments is essential to assess the scalability, efficiency, and performance of the proposed models. Additionally, the metrics used to evaluate the models in the literature may not reflect the real-world impact of CTR prediction and click fraud detection accurately. Future research should focus on conducting comprehensive

evaluations in realistic testing environments to provide more practical insights and assess the true effectiveness of the proposed techniques.

Addressing these gaps in the literature will contribute to the advancement of CTR prediction and click fraud detection methodologies. By promoting reproducibility, accounting for data complexity and overlap, and conducting rigorous real-world testing, researchers can enhance the reliability, applicability, and practicality of the models and techniques proposed in the field.

In the subsequent chapters, we will present our work aimed at addressing some of the gaps identified in the previous section. Our research is designed to contribute to the field of CTR prediction and click fraud detection by focusing on two major aspects. The first part of our study delves into sampling techniques and their impact on CTR prediction. By exploring various sampling methods and their effectiveness in handling imbalanced and overlapping data, we aim to improve the accuracy and reliability of CTR prediction models. The second part of our research centers around click fraud detection, where we develop novel algorithms and approaches to effectively identify and mitigate fraudulent activities. Through these two major parts, our work aims to fill existing gaps in the literature, providing practical insights and contributing to the advancement of CTR prediction and click fraud detection methodologies.

CHAPTER 2

Methodology

The methodology employed to investigate the effects of data sampling techniques on native advertising data. The methodology encompasses several key aspects, including research design, data collection and preprocessing, machine learning algorithms, handling imbalanced data, evaluation metrics, and the experimental setup. Each of these components plays a crucial role in our study, allowing us to gain valuable insights into optimizing feature importance and prediction accuracy in native advertising. By following a rigorous and systematic approach, we ensure the reliability and validity of our findings, contributing to the advancement of knowledge in the field.

2.1 Research Design

The research design for addressing the problems of both click-through rate (CTR) prediction and click fraud detection involves a systematic approach to develop effective models and algorithms. While the specific techniques and methodologies may vary depending on the case at hand, the general framework remains consistent.

The research design begins with a comprehensive literature review to understand the existing approaches and techniques in CTR prediction and click fraud detection across various domains. This review forms the foundation for identifying the most relevant methods and algorithms to be applied in the research.

The next step involves data collection, preprocessing, and feature engineering. Relevant data related to ad impressions, clicks, user behavior, and contextual information are gathered from appropriate sources. The data is then processed to remove noise, handle missing values, and transform it into suitable formats for analysis. Feature engineering techniques are applied to extract meaningful features from the data that can capture the underlying patterns and relationships.

Following data preparation, the research design incorporates the development and implementation of machine learning models. For CTR prediction, models such as logistic regression, factorization machines, or neural networks can be employed. These models are trained using historical data to learn the relationships between ad features, user information, and the likelihood of clicks. In the case of click fraud detection, anomaly detection techniques, supervised learning algorithms, or user behavior profiling methods

can be utilized, as discussed in the previous sections.

The evaluation of the models is a critical aspect of the research design. Metrics such as accuracy, precision, recall, and F1-score are commonly used to assess the performance of CTR prediction models. For click fraud detection, metrics like detection rate, false positive rate, and area under the receiver operating characteristic curve (AUC-ROC) are commonly employed. The models are evaluated on both training and testing datasets to ensure their effectiveness in capturing patterns and detecting fraudulent activities.

It is important to note that while the research design presented here is generic, it can be modified and tailored based on the specific requirements of the CTR prediction or click fraud detection problem at hand. The specific models, algorithms, and evaluation metrics may change accordingly, but the underlying principles and steps remain consistent.

Overall, this research design provides a systematic framework for addressing the challenges of CTR prediction and click fraud detection, allowing for effective analysis and modeling to enhance advertising campaigns' performance and mitigate fraudulent activities.

2.2 Data Collection and Preprocessing

Data collection plays a crucial role in addressing the challenges of click fraud detection and click-through rate (CTR) prediction. The quality and relevance of the data used directly impact the accuracy and effectiveness of the models and algorithms developed. In the case of click fraud detection, datasets such as the TalkingData AdTracking dataset provide a rich source of information about user interactions with ads, enabling researchers to analyze and detect fraudulent click activities accurately. On the other hand, for CTR prediction, real-life data from native advertising platforms like the dataset provided by ReadPeak offers insights into user behaviors, ad impressions, and click patterns, allowing advertisers to make data-driven decisions and optimize their campaigns. Collecting diverse and representative data, and applying appropriate preprocessing techniques, ensures the reliability and validity of the input, leading to robust and reliable results in combating click fraud and improving advertising performance.

2.2.1 Native Advertising Dataset: ReadPeak

The dataset is provided by a real Demand Side Platform (DSP) called ReadPeak [1]. It is collected from ReadPeak, which is a platform that enables advertisers to promote their content online by buying inventory from a publisher of their choice. The training data used in this study is one week of native advertisements that were shown to consumers all over Finland in the month of June 2022. The dataset comprises of a total of 17.925.833 lines of data, out of which 72.540 lines represent confirmed clicks. The testing set used in this study is one day's worth of data collected from the following week, comprising of 4.803.760 lines of data, out of which 17.179 lines represent confirmed clicks.

The dataset includes 13 features and a target feature named 'label.' Table 2.1 describes and details the features available. It is evident from the purpose of this study and the number of clicks observed in comparison to the total data, that the dataset used is highly unbalanced.

Features	Format	Description
art	int	Article id identifies each unique advertisement
loc	int	The location id in the ReadPeak platform, one site represent a location.
tag	string	The placement identification. It identifies the advertisement placement within a location
dt	int	The device type
type	int	represents the type of advertisement it is; one of 2 types: native or banner
os	string	Operating system of the device e.g. Android
make	string	The maker brand of the device e.g. Samsung or Apple
client	int	Represents the client id within the ReadPeak platform
city	string	The city of where the advertisement will be shown
lang	string	The language of the navigator used to access the advertisement
cl	int	The number of clicks a certain advertisement accumulated in a certain location
ts	datetime	The time stamp of when the request to show the advertisement is received
label	boolean	Show either if the advertisement was clicked or not

Table 2.1: Features description

2.2.2 Click Fraud Dataset: TalkingData

The TalkingData AdTracking Fraud Detection dataset from Kaggle [12] was utilized in this study to train and evaluate the proposed model. The dataset comprises records of 200 million clicks over four days, with eight features. TalkingData, a leading big data service provider in China, caters to a significant proportion of active mobile devices in the country and processes a vast volume of data, including 3 billion clicks per day, many of which could be fraudulent. Table 1 outlines the features included in the dataset, such as IP, App, Device, OS, Channel, click time, attributed time, and is Attributed.

2.2.3 Data Preprocessing

Preprocessing plays a vital role in preparing the data for analysis and modeling in click fraud detection and CTR prediction. This section discusses various techniques and approaches commonly employed in the preprocessing phase to enhance the quality and usability of the data.

2.2.3.1 Missing Values

Missing data is a common issue encountered in datasets and can have a significant impact on the analysis and modeling processes. It is crucial to handle missing data appropriately to avoid biased or inaccurate results. The following are some common approaches to address missing data:

Features	Format	Description
IP	int32	IP address of click
App	int16	App ID for marketing
Device	int16	Device type ID of user mobile phone
OS	int16	OS Version ID of user mobile phone
Channel	int16	Channel ID of mobile ad publisher
Click_Time	int8	Time Stamp of click (UTC)
Attributed_Time	int8	If user download the app for after clicking an ad, this is the time of the app download
Is_Attributed	bool	the target that is to be predicted, indicating the presence of fraud

Table 2.2: TalkingData Features description

1. **Complete Case Analysis:** In this approach, any data points with missing values are excluded from the analysis. While this method is straightforward, it may lead to a loss of valuable information, especially if the missing data is not random. It is suitable when the amount of missing data is minimal and does not affect the overall analysis.

2. **Mean/Mode/Median Imputation:** This method involves replacing missing values with the mean (for continuous variables) or mode (for categorical variables) of the available data. Imputation assumes that the missing values are missing completely at random (MCAR) or missing at random (MAR). While it is a simple approach, it may distort the distribution of the variable and underestimate the variability in the data.

3. **Regression Imputation:** Regression imputation estimates missing values by predicting them from other variables using regression models. This method captures the relationship between the missing variable and other variables. However, it assumes that the relationship between the missing variable and the predictors is linear and may introduce additional uncertainty if the model is misspecified.

4. **Multiple Imputation:** Multiple imputation generates multiple plausible imputed datasets by incorporating the uncertainty associated with missing values. Each dataset is analyzed separately, and the results are combined to provide more robust estimates. Multiple imputation is suitable when the missing data mechanism is ignorable and provides more accurate estimates compared to single imputation methods.

It is important to consider the characteristics of the data and the underlying reasons for missingness when choosing an appropriate approach. Understanding the missing data mechanism, such as missing completely at random (MCAR), missing at random (MAR), or missing not at random (MNAR), can guide the selection of the most suitable imputation technique.

2.2.3.2 Data Cleaning

Data cleaning, also known as data cleansing or data scrubbing, is an essential step in the data preprocessing phase. It involves identifying and rectifying or removing errors,

inconsistencies, and inaccuracies in the dataset. Proper data cleaning ensures that the data is reliable, consistent, and suitable for analysis. Here are some common techniques used in data cleaning:

1. **Handling Duplicate Entries:** Duplicate entries can skew the analysis and lead to biased results. Identifying and removing or merging duplicate records is crucial to ensure data accuracy. Techniques such as deduplication based on key attributes or similarity measures can be employed to detect and resolve duplicates.

2. **Removing Irrelevant or Outliers:** Irrelevant or erroneous data points that do not contribute to the analysis or model can be identified and removed. Outliers, which are extreme values that deviate significantly from the rest of the dataset, can also be handled by either removing them, replacing them with appropriate values, or transforming them to minimize their impact.

3. **Resolving Inconsistencies:** Inconsistencies in the dataset, such as misspellings, inconsistent capitalization, or variations in formatting, can be addressed through standardization or normalization techniques. This ensures consistency in the data and facilitates accurate analysis.

4. **Addressing Incomplete or Inaccurate Data:** Missing data, as discussed in the previous section, should be handled appropriately using imputation methods or by excluding incomplete records. Inaccurate data, such as data that violates logical constraints or falls outside expected ranges, can be corrected or flagged for further investigation.

5. **Handling Data Format Issues:** Data may be stored in different formats or units, making it necessary to convert or standardize them. This includes converting dates into a consistent format, converting units of measurement, or harmonizing data types across variables.

6. **Dealing with Data Integration:** In situations where data is sourced from multiple datasets or databases, data integration techniques are employed to combine and reconcile the data. This ensures data consistency and minimizes errors resulting from merging different sources.

Data cleaning is an iterative process that requires careful examination and understanding of the data. It is important to document the decisions and transformations made during the cleaning process to maintain transparency and reproducibility. By performing thorough data cleaning, researchers can ensure the accuracy, reliability, and integrity of the data, providing a solid foundation for subsequent analysis and modeling tasks in both click fraud detection and CTR prediction.

2.2.3.3 Normalization

The purpose of normalization is to convert the values of numeric columns in the dataset to a common scale while preserving disparities in value ranges. This is only important when the ranges of the features differ. The benefit of data normalization is that many machine learning algorithms tend to perform better and it speeds up the training in some cases.

The data in this research is normalized to eliminate the influence of dimensionless disparities across features in the dataset. The goal is to set the data mean to 1 and the variance to 0. The following shows the calculation formula to normalize a certain feature X:

$$x' = \frac{X - X_{mean}}{X_{max} - X_{min}} \tag{2.1}$$

where X_{mean} is the mean value, X_{max} is the maximum value, and X_{min} is the minimum value.

2.2.3.4 Categorical Encoding

Categorical encoding refers to the process of assigning numeric values to nominal (non-numeric) features in order to facilitate the processing task. Since we are using only neural networks in our model, it is necessary to convert the textual data of the dataset into numerical values. Generally speaking, there are two approaches to encode categorical variables:

1. One-hot (binary) encoding: A binary representation of nominal features where the categorical value is removed and a new binary variable is added for each unique nominal value.

2. Integer encoding: Refers to the process of coding a categorical variable using integers such as 1, 2, and 3.

The main difference between the two approaches is that One-Hot Encoding preserves the order relationship between the values of the nominal feature, while Label Encoding does not. Additionally, One-Hot Encoding requires a higher memory consumption than Label Encoding. In this research, We are using label encoding, also called integer encoding, because most of our categorical variables have a large number of categories. Using this approach allows for less memory consumption and makes training the data less computationally expensive.

2.2.4 Feature Selection

One of the key aspects of machine learning is feature selection, which aims to eliminate irrelevant and redundant information in the dataset, in order to boost model accuracy and improve computational efficiency. There are several popular feature selection techniques, such as filter, wrapper, and embedded methods [6]. In this study, we use the Boruta algorithm [46], an advanced feature selection method based on Random Forest, to select the most important features for our model. Boruta not only identifies the significance of each feature, but also evaluates whether it should be retained for further analysis. This approach is considered to be more efficient and accurate than other feature selection methods.

The Boruta algorithm's core process is as follows:

1. Creates a shadow feature by randomly sorting the original features to create a shadow feature matrix, then splicing the shadow feature matrix with the original feature matrix to create the new feature matrix.

2. For training, the new feature matrix is fed through a Random Forest classifier, which outputs the relevance of features v.

3. The original feature and shadow feature's z scores are computed, and the formula is as follows:

$$z_{score} = \frac{A_v}{S_v} \tag{2.2}$$

Where the average value of feature importance is represented by A_v, while the standard deviation of feature importance is S_v.

4. The maximum z_{score}, indicated as Z_{max}, is searched in the shadow feature.

5. If the z_{score} of the original feature is larger than Z_{max}, the feature is classified as "important" if the original feature's z_{score} is less than Z_{max}, on the other hand, the feature will be classed as "unimportant" and removed.

6. Repeat steps 1–5 until all of the features have been noted.

2.2.5 Data-split

We use Stratified KFold ($K = 5$) to split the dataset, retaining the sample category ratio while dividing the complete development set into five independent subgroups. For each split in this procedure, four-fifths of the dataset are used. The final fifth serves as the test set and the remaining as the training set. Each split can be regarded as the ith time ($i = 1, ..., 5$), and AUC is calculated on the ith test set [81]. It is important to note that the test set obtained each time is set aside and it was not be used in the scaling, recoding, or model-building processes. The test set needs to be separated from the training process, because the oversampling strategy will clone or synthesize some minority samples, meaning the data obtained in this way cannot accurately represent the original dataset.

But to also put these sampling methods to the test in a real environment we re-train the model on all the available data and then test it on a one day data three days after the collected data.

2.3 Machine Learning Algorithms

Machine learning algorithms play a crucial role in click fraud detection and CTR prediction tasks. These algorithms are designed to automatically learn patterns and relationships from the data, enabling accurate prediction and classification. Here are some commonly used machine learning algorithms for these tasks:

1. **Decision Trees:** Decision trees are simple yet powerful models that use a hierarchical structure of nodes and branches to make predictions. They split the data based on features and their thresholds, creating a tree-like structure. Decision trees can be used for both click fraud detection and CTR prediction, as they can handle categorical and numerical data effectively.

2. **Naive Bayes (NB):** Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem. It assumes independence among features, making it computationally efficient. NB has been successfully applied to click fraud detection, where it estimates the probability of a click being fraudulent based on the features associated with it.

3. **AdaBoost (Adaptive Boosting):** AdaBoost is an ensemble learning algorithm that combines multiple weak learners to create a strong classifier. It iteratively adjusts the weights of misclassified samples, allowing subsequent learners to focus on challenging instances. AdaBoost has been used in click fraud detection and CTR prediction, leveraging the strengths of multiple weak classifiers to improve overall performance.

4. **Logistic Regression (LR):** Logistic Regression is a widely used algorithm for binary classification problems. It models the relationship between input variables and the probability of a specific outcome. In click fraud detection, LR can be used to classify clicks as fraudulent or legitimate based on various features.

5. **Random Forest (RF):** Random Forest is an ensemble learning algorithm that combines multiple decision trees to make predictions. It is effective in handling high-dimensional data and capturing complex interactions between variables. RF can be applied to both click fraud detection and CTR prediction tasks.

6. **Gradient Boosting Machines (GBM):** GBM is another popular ensemble learning technique that builds an ensemble of weak prediction models, typically decision trees, to create a strong predictive model. GBM is known for its ability to handle imbalanced datasets and capture non-linear relationships. It has shown promising results in click fraud detection and CTR prediction tasks.

7. **Support Vector Machines (SVM):** SVM is a supervised learning algorithm that separates data points into different classes using a hyperplane. It is effective in handling high-dimensional data and can capture non-linear relationships using kernel functions. SVM has been used in click fraud detection to classify fraudulent clicks from legitimate ones.

8. **Deep Learning:** Deep learning models, such as Artificial Neural Networks (ANNs), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs), have gained significant attention in recent years. These models are capable of learning complex patterns and have shown promising results in various tasks, including click fraud detection and CTR prediction. They can automatically extract features and capture non-linear relationships in the data.

9. **XGBoost:** XGBoost is an optimized gradient boosting framework that has gained popularity for its efficiency and scalability. It combines the advantages of both gradient boosting and regularized regression techniques. XGBoost has been successfully applied to click fraud detection and CTR prediction tasks, providing accurate and robust predictions.

It is worth mentioning that the choice of machine learning algorithm depends on the specific characteristics of the data, the problem at hand, and the desired performance

metrics. It is common to experiment with multiple algorithms and compare their performance to select the most suitable one. In subsequent chapters, we will delve into these algorithms, providing detailed explanations as needed to understand their workings and how they can be effectively applied in various scenarios.

2.4 Evaluation Metrics

To assess the prediction outcomes, the output confusion matrix is used to calculate *Accuracy*, *Precision*, *Recall*, and *F1 Score*. Table 2.3 from [8] depicts the generated confusion matrix.

confusion matrix		Predicted	
		Positive	Negative
True	Positive	TP	FN
	Negative	FP	FN

Table 2.3: Confusion Matrix

True Positive (TP) denotes a situation where both the true and predicted values are positive, which means that the number of positive samples has been correctly predicted. False Positive (FP) denotes a situation where the true value is negative, but the predicted value is positive, meaning that the number of negative samples has been incorrectly predicted to be positive. True Negative (TN) denotes a situation where both the true and predicted values are negative, meaning that the number of negative samples has been correctly predicted. False Negative (FN) denotes a situation where the true value is positive, but the predicted value is negative, meaning that the number of positive samples has been incorrectly predicted to be negative.

The ratio of accurately predicted samples to the total number of samples is known as *Accuracy*, and the method for calculating it is as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{2.3}$$

The fraction of valid predictions in the sample with a positive projected value is known as *Precision*. The formula for calculating it is as follows:

$$Precision = \frac{TP}{(TP + FP)} \tag{2.4}$$

The fraction of correct predictions in a sample with a positive real value is called *Recall*. The following is the calculating formula:

$$Recall = \frac{TP}{(TP + FN)} \tag{2.5}$$

In this study, *Precision* is a measure of the proportion of the model's predicted fraudulent clicks that are actually fraudulent, whereas *Recall*a measure of the proportion of the actual fraudulent clicks that are correctly identified by the model. Further, *F1*

Score, which is the harmonic average of *Precision* and *Recall*, can take both into account. Consequently, we will use *F1 Score* as one of the four main metrics to evaluate this work. The formula for calculating the *F1 Score* is as follows:

$$F1score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{2.6}$$

However, most studies on imbalanced class data have found that accuracy may not be the best metric to evaluate the performance of a model on such datasets [15]. This is because in many real-world applications, the minority class is often more important and any errors in classifying it (i.e False Negatives or False Positives) may have a significant impact on the overall performance of the model. For example, in the case of click-through rate (CTR) prediction, the goal is to correctly identify the minority (positive) cases in order to find clicks, as they are usually the desired outcome of an online advertisement. Therefore, incorrectly assessing where to show the advertisements can lead to significant financial losses.

We describe an alternative performance evaluation metric, the area under the Receiver Operating Characteristic (ROC). ROC curve plots the True Positive Rate ($TPR = TP/(TP+FN)$) on the y-axis against the False Positive Rate ($FPR = FP/(TN+FP)$) on the x-axis at various threshold values [11]. The area under the ROC curve (AUC) identifies the classifier's ability to distinguish between classes and compares ROC curves [35]. As such, we use the following metrics to evaluate the compared methods:

AUC: A widely used metric for CTR prediction tasks. It indicates the Area Under the ROC Curve over the test set. It reflects the probability that a model ranks a randomly chosen positive instance higher than a randomly chosen negative instance. Here, the larger the better. We will note that an improvement of 1% in AUC is usually regarded as significant for the CTR prediction because it will bring a large increase in a company's revenue if the company has a very large user base.

LogLoss (cross entropy) a training goal function, and it may be computed as follows:

$$LogLoss = \frac{1}{N}\sum_{i=1}^{N}(y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i)) \tag{2.7}$$

where N is the total number of training samples, y_i is the true label of i^{th} instance and $\hat{y}_i$ is the prediction.

2.5 Experimental Setup

All methodologies and models were tested on an Amazon Web Services (AWS) SageMaker instance with 32 vCPUs and 72GB of RAM. The implementation was done using two programming languages, Python 3.9.11 and R 4.2.0. The Python implementation used several packages such as Pandas, Numpy, and Sklearn. For feature importance, we primarily used the Boruta package in R.

Part I

Unveiling the Impact of Data Imbalance on CTR Prediction: Investigating Sampling Methods

CHAPTER 3

Effects of Data Sampling Techniques on Native Advertising Data

This chapter, delve into the impact of data sampling techniques on native advertising data. We evaluate the performance of 19 class imbalance methods using four classical classifiers and identify the best performing methods based on the AUC and LogLoss metrics. Furthermore, we conduct an analysis of the unbalanced data problem using real-world data from ReadPeak, outlining the steps for feature engineering, selection, and data cleaning. Through experimentation, we discover that feature importance varies with different Imbalance Ratios, utilizing the Boruta algorithm to compute feature importance with three levels of sampling. This highlights the importance of balancing the data for accurate results. Using Random Forest and random undersampling, we show that improving the balance between positive and negative samples enhances performance. However, there is a threshold of imbalance ratio beyond which the performance improvement becomes marginal. Additionally, we provide the research community with access to real-world datasets, facilitating the study of CTR Prediction in the native advertisement field and the development of potential solutions.

3.1 Introduction

Native advertising has emerged as a highly influential form of online advertising, designed to seamlessly blend in with the surrounding content on digital platforms, such as news, sports, and social websites. With its ability to capture users' attention in a non-disruptive manner, native ads have gained immense popularity among advertisers and online platforms alike. As the demand for native advertising continues to rise, understanding its intricacies and optimizing the analysis of native advertising data become critical for advertisers and marketers [41, 95].

Major internet companies such as Yahoo, Google, and Facebook rely on online

advertising as their primary source of revenue, functioning as intermediaries between publishers and advertisers in ad networks [41]. In 2021, advertisers are estimated to have spent 118.72 billion dollars on display advertising [95].

The dataset utilized in this study exhibits a substantial disparity between non-clicked and clicked advertisements, with the latter being notably fewer in number. Addressing this issue, researchers have devised a variety of class imbalance techniques and performance evaluation criteria that hold significant importance in the context of this thesis. To address this challenge, we implemented multiple class imbalance methods using a real-world dataset sourced from the native advertising platform ReadPeak [1]. This dataset contains information regarding in-screen advertisements and their click status. Notably, the dataset reveals an extreme imbalance between clicked and non-clicked instances, with a staggering ratio of 250 non-clicked instances for every 1 clicked instance.

3.2 Data Treatment

The Dataset we will use in this chapter have already been introduced in the chapter 2 in this section we will talk about the data treatment

Data pre-processing is a crucial step that involves handling missing values and adjusting the features of the datasets. In our specific case, we employed label encoding to appropriately handle categorical data and normalized the data to ensure consistency. The reasoning behind our choice of label encoding and normalization techniques was thoroughly explained in the previous chapter, providing a solid foundation for the processes we implemented.

3.2.1 Missing Values

The initial dataset consists of almost 18 millions in-screen advertisements with 13 columns. Out of these columns, six were found to have missing values. The missing values in the 'lang' feature were replaced with the most common category, while the missing values in the 'tag', 'city', 'client', 'os', and 'make' columns were treated as a new category. This approach was chosen due to the nature of these features.

3.2.2 Features Cleaning

The majority of the features present in our dataset underwent a number of operations to make them more manageable and to extract as much relevant information as possible. In this section, we will describe the process that each feature underwent.

- ◇ **tag** :For the majority of the variable, it was mainly an integer but there were instances that contained characters. We used Python's regular expression to clean the feature by removing the characters from the instances in question.
- ◇ **os**: We turned everything into lower case and categorized all the Operating Systems (OS) containing the word "android" into a category and did the same for "ios". From the data analysis, we curated a list of the most common OS. Any other values in the 'os' column not belonging to the list were put into their own category that we called 'unknown'.

- ◇ **make**: A similar process was done to the 'make' feature as was done to the 'os' feature. We categorized all Samsung and Apple devices in their own respective categories and limited the categories only to the most common ones. Any other values not belonging to the list were put into a category called 'unknown'.

- ◇ **city**: We matched the cities that were entered in different manners and lower-cased all entries. We only kept the top 25 cities obtained from the data analysis. Any other cities not belonging to the list were put into their own category called 'unknown'.

- ◇ **lang**: All languages that were repeated a very small number of times were removed. We matched all languages that were denoted in different ways to ensure consistency in the data.

- ◇ **ts**: From the 'timestamp' feature, we extracted the 'Day' and 'Hour' feature, which we used instead of the 'ts' feature. The 'Day' feature is represented with numbers from 0 to 6, with 0 being Sunday. The 'Hour' feature is a float, with the decimal values representing the minutes over 60.

3.2.3 Imbalance Ratio

The imbalance ratio (IR) is an essential parameter in imbalanced learning. It measures the proportional relationship between the majority and minority classes in the experiment [61]. The formula is given by Eq. 3:

$$IR = Instances_{Minority}/Instances_{Majority} \tag{3.1}$$

Most of the data-level methods used in the research involve resampling the majority or minority class in the original dataset, which increases the minority class samples or decreases the majority class samples. This changes the imbalance ratio of the dataset. As the IR decreases, the difference in sample size between the majority and minority class becomes larger, indicating that the dataset is more imbalanced. On the other hand, when the IR value is closer to 1, the dataset tends to be more balanced. Therefore, In this chapter we use IR as a metric to evaluate sampling techniques.

3.3 Imbalanced Data Strategies and Models Selection

The majority of the class-imbalanced learning strategies used in this study are data-level techniques and ensemble systems. Specifically, this research primarily investigates imbalance techniques in the Imblearn library [34]. Resampling techniques, including undersampling, over-sampling, and hybrid sampling methods, are mostly used for data-level procedures.

Undersampling techniques include :

- ◇ Random UnderSampling (RUS).

- ◇ Edited Nearest Neighbors (ENN).

- ◇ Instance Hardness Threshold (IHT).

- ◇ Near Miss (NM).

- ◇ Neighbourhood Cleaning Rule (NCR).
- ◇ One-Sided Selection (OSS).
- ◇ Tomek Links(TL).

Given the size of the dataset used in this work, algorithms based on k-NN, such as All k-Nearest Neighbors (All k-NN), Cluster Centroids (CC), and Condense Nearest Neighbors (CNN), are computationally expensive and take a long time to run. Therefore, CC, All k-NN, and CNN are not included in this study.
Examples of oversampling methods include:

- ◇ Synthetic Minority Oversampling Technique (SMOTE).
- ◇ Random Oversampling (ROS).
- ◇ Adaptive Synthetic (ADASYN).
- ◇ Borderline SMOTE.

Furthermore, methods for hybrid sampling include:

- ◇ SMOTE-ENN.
- ◇ SMOTE-Tomek.
- ◇ RUS&SMOTE
- ◇ RUS&ROS

To forecast advertisement clicks, these data level techniques are paired with classifiers. We also trained ensemble systems using the built-in classifiers, which are:

- ◇ Easy Ensemble.
- ◇ Random Undersampling Boost.
- ◇ Balanced Random Forest.
- ◇ Balanced Bagging (RUSBoost).

To note, this work will not cover the Balance Cascade algorithm because it has been regularly modified by the Imblearn library in recent years and was finally dropped in version 0.6.0.

In the next section , we will discuss the effectiveness and safety of using oversampling techniques in our native advertising dataset.

Oversampling Concerns

Oversampling has been shown to improve model performance on imbalanced datasets, But, a recent review paper [82] has raised concerns about the risks associated with this approach. The paper argues that oversampling methods assume that all synthesized data belong to the minority class, without providing any guarantee that some of the generated examples may actually belong to the majority class.

Despite the concerns raised, we argue that oversampling techniques can still be a useful tool for CTR prediction on native advertising data. In fact, we found that oversampling techniques had a positive effect on the performance of CTR prediction models on our dataset. Additionally, the extreme imbalance and nature of the task at hand make it important to generate more positive samples, as they are so scarce.

It is worth noting that overlap in the native advertising data used can explain that some of the generated examples may belong to the majority class. This means that the risk of synthesizing minority class data that actually belongs to the majority class doesn't make the generated samples invalid. Additionally, we have explored hybrid methods that combine undersampling and oversampling techniques to create a balanced dataset. This approach can help alleviate some of the concerns of the mentioned paper.

Lastly, given the nature of the current study, it would be imprudent not to include oversampling in our systematic comparison. Oversampling is a commonly used techniques for addressing class imbalance in machine learning, and excluding it from our evaluation could limit the usefulness and relevance of our findings. By including oversampling in our comparison, we can provide a more comprehensive and nuanced analysis of the relative benefits and drawbacks of different sampling techniques for CTR prediction on native advertising data.

Overall, while it is important to be aware of the potential risks associated with oversampling, we believe that the benefits of generating more positive samples outweigh the risks in the current case. Therefore, we conclude that oversampling techniques can be a useful and tolerable approach for CTR prediction on native advertising data.

3.3.1 Baseline Model Selection

The selection of algorithms for this study was based on their suitability for handling imbalanced datasets and their computational efficiency. The algorithms selected, namely Logistic Regression [33], Naive Bayes [96], Decision Tree [66], and Random Forest [50], are known for their simplicity, low computational cost, and ability to handle imbalanced datasets. These algorithms typically work well "out of the bag," meaning that they often provide a good baseline model without requiring extensive tuning.

While several other machine learning algorithms, including Support Vector Machines (SVM) and k-Nearest Neighbors (KNN), were initially considered, they were ultimately excluded from the study due to their high computational cost and memory requirements. SVM is known for its high accuracy and ability to handle complex datasets, but it can be slow and require a large amount of memory. KNN, on the other hand, is a non-parametric algorithm that is easy to implement and interpret, but it can also be slow and require significant amounts of memory, especially when dealing with large datasets.

In addition, neural networks and factorization machines were not included in the selection of algorithms for this study. While neural networks have shown promising results in various domains, they can be computationally expensive and require significant

amounts of data for training. Moreover, their suitability for this study was limited by the size of the dataset and the need to compare the performance of different sampling techniques. Similarly, factorization machines are also computationally expensive and can be challenging to implement, requiring a substantial amount of expertise and resources. Therefore, the selected algorithms were deemed sufficient to provide a useful baseline for comparison purposes and achieve the primary goal of identifying the most suitable sampling technique for predicting CTR in the context of native advertising.

3.3.2 Evaluation

In order to determine the best class-imbalanced technique for the dataset based on the following standard, this study uses four traditional classifiers as the baseline model:

- ◇ Logistic Regression (LR).
- ◇ Random Forest (RF).
- ◇ Naive Bayes(NB).
- ◇ Decision Tree(DT).

We will compare the results based on the metrics mentioned in the section.

3.4 Experiment

Following, we will explore the effects of the IR on both the feature importance and the predictions result. Later also we list the imbalance ratio provided by the resampling technique and then show the prediction results of the imbalance technique model, which can help analyse the effect of the imbalance technique comprehensively. We have used the area under the curve (AUC) and *LogLoss* error for the evaluation of the proposed methods. The AUC and *LogLoss* provide the best indication of performance when the dataset is imbalanced [12, 83].

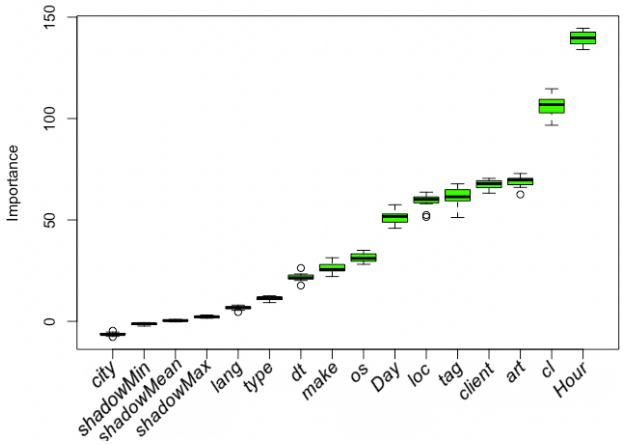
3.4.1 Feature Selection

The Boruta algorithm is used to select features for our dataset. Blue boxplots indicate the minimal, average, and maximum Z score of a shadow attribute. As previously explained, these shadow attributes are generated by adding randomness to the dataset by creating shuffled copies of all features. As seen in the figure, the importance of shadow features should be minimal in comparison to the rest of the authentic features. The Boruta algorithm assigns one of three outcomes for each feature: accepted (green), inconclusive (yellow), and rejected (red) in the boxplot. The algorithm outputs can be observed in the figure 3.1.

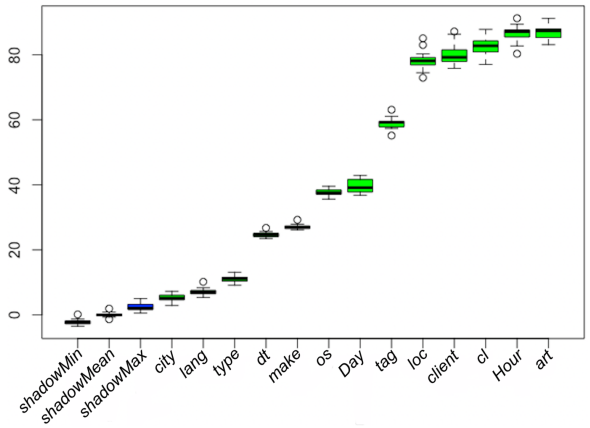
Given that we are trying to figure out which sampling method work best with native advertising data, our first experiment pertained to how the feature importance of our variables change with the level of sampling. To do this, we used three levels of sampling:

- ◇ First level was with the full data with no sampling applied, where the clicks represent approximately 0.4% of the data. The results for the arrangement are depicted in the bottom of both Table 3.1c and Figure 3.1c.
- ◇ Represented in Figure 3.1a and Table 3.1a is level 2 where we applied a moderate random undersampling to make the clicks constitute 5% .
- ◇ The last level was where we made both clicks and non clicks completely balanced. The results for this level can be seen in Figure 3.1b and Table 3.1b.

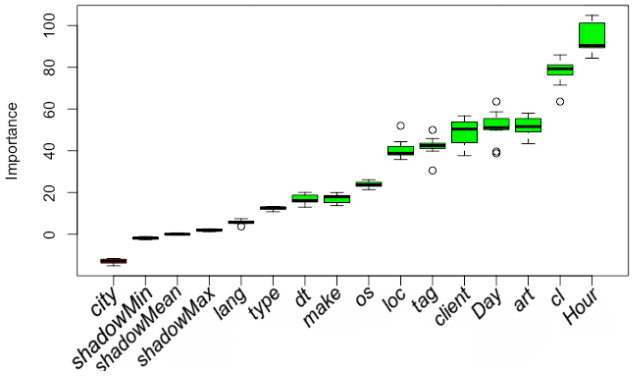
Applying the Boruta algorithm on our data at different proportions shows differences in the importance of the features. Between the first and the second level, the city feature is rejected in both, while all the other features are accepted. Additionally, there is a change in the order of importance of the features between the first and the third level. The bottom three features remain consistent, and the city feature is rejected in both the full data and moderate undersampling, maintaining the lowest importance in the balanced layout. In general, it can be observed that the top, middle, and lower ranking features remain in the same splits, regardless of the different sampling proportions.



(a) The clicks representing 5% of the data



(b) The clicks representing 50% of the data



(c) Unsampled data

Figure 3.1: Feature importance over different level of data sampling

	meanImp	medianImp	minImp	maxImp	normHits	decision
art	68.976	69.702	62.571	72.890	1	Confirmed
loc	59.131	60.156	51.456	63.700	1	Confirmed
tag	61.552	61.418	51.194	67.801	1	Confirmed
dt	21.815	21.489	17.663	26.275	1	Confirmed
type	11.294	11.665	9.237	12.583	1	Confirmed
os	31.516	31.067	28.121	35.012	1	Confirmed
make	26.479	25.562	22.102	31.356	1	Confirmed
client	67.433	67.900	63.204	70.505	1	Confirmed
city	-6.294	-6.492	-7.723	-4.631	0	Rejected
lang	6.604	6.770	4.575	7.996	1	Confirmed
cl	106.349	106.881	96.690	114.683	1	Confirmed
Day	51.238	51.739	45.907	57.475	1	Confirmed
Hour	139.534	139.746	133.984	144.501	1	Confirmed

(a) The clicks representing 5% of the data

	meanImp	medianImp	minImp	maxImp	normHits	decision
art	86.913	87.357	83.130	91.198	1.00	Confirmed
loc	78.277	78.123	72.937	85.077	1.00	Confirmed
tag	59.011	59.256	55.166	63.089	1.00	Confirmed
dt	24.708	24.623	23.477	26.754	1.00	Confirmed
type	10.999	11.105	9.110	13.098	1.00	Confirmed
os	37.746	37.526	35.608	39.553	1.00	Confirmed
make	27.059	26.903	26.155	29.249	1.00	Confirmed
client	79.939	79.231	75.837	87.174	1.00	Confirmed
city	5.171	5.144	2.877	7.228	0.93	Confirmed
lang	7.166	7.006	5.357	10.162	1.00	Confirmed
cl	82.682	82.780	77.044	87.834	1.00	Confirmed
Day	39.619	39.144	36.800	42.877	1.00	Confirmed
Hour	86.407	87.109	80.334	91.250	1.00	Confirmed

(b) The clicks representing 50% of the data

	meanImp	medianImp	minImp	maxImp	normHits	decision
art	51.588	51.602	43.382	58.019	1	Confirmed
loc	40.607	38.776	35.797	51.969	1	Confirmed
tag	42.129	42.452	30.532	49.967	1	Confirmed
dt	16.664	16.125	12.952	20.040	1	Confirmed
type	12.390	12.627	10.760	13.262	1	Confirmed
os	23.885	23.651	21.371	26.104	1	Confirmed
make	16.942	17.920	13.704	19.956	1	Confirmed
client	48.559	50.380	37.632	56.698	1	Confirmed
city	-12.965	-12.903	-15.127	-11.643	0	Rejected
lang	5.742	5.637	3.740	7.426	1	Confirmed
cl	77.813	79.257	63.579	85.926	1	Confirmed
Day	51.454	51.083	38.766	63.577	1	Confirmed
Hour	94.360	90.421	84.359	104.858	1	Confirmed

(c) Unsampled data

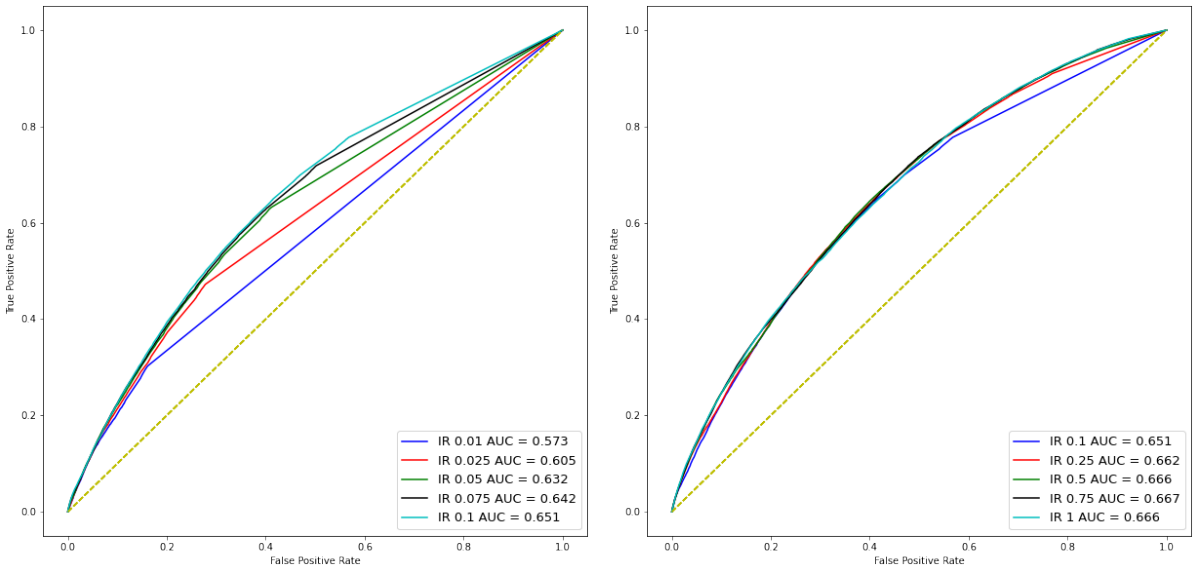
Table 3.1: Feature importance over different level of data sampling

3.4.2 Imbalance Ratio Effect

In this subsection, we discuss the imbalance ratio effect. Given that the data studied in this work is very unbalanced, we ask the following questions: Can the unbalance itself be helpful in getting a better perdition? or, will balancing the data completely give the best results? To answer these question we run a Random Forest model on different imbalance ratios ranging from IR = [0.01 .... 1]. To measure the effect of the ratio imbalances we use the AUC metric. AUC has been shown through various research to be a great indicator of performance for imbalanced datasets.

In Figure 3.2 illustrates a gradual improvement of the AUC score from 0.547 to 0.663. This increase is a clear indication that balancing the data plays a significant role in the performance of our predictor. It is obvious that the substantial improvements in the AUC are mainly at the level where the IR is between [0.01...0.1]. While it is true that there is improvement as the IR increases to 1, the increase is quite marginal after IR = 0.25.

To answer the questions posed at the beginning of this subsection, it can be concluded that fully balancing the data yields the best results. It is also clear that the imbalance itself is not as helpful. With this in mind, another question arises: After the benefit of increasing the imbalance ratio using undersampling becomes marginal, would it make sense to combine other sampling techniques to further improve our predictions? This will be addressed by generating two hybrid sampling techniques.



(a) IR from 0.01 to 0.1 (b) IR from 0.1 to 1

Figure 3.2: Effects of Ratio imbalance on training

3.4.3 Sampling Techniques Effect on Native Advertisements Data

The class imbalanced dataset used has an imbalanced ratio of 0.004. Through resampling technology, the class proportion of the dataset has changed. Table 3.2 lists the class distribution in the training set after each sampling.

The outcome shows that undersampling affects the majority of samples, oversampling only affects the minority of samples, and the hybrid method affects both categories.

All sampling techniques enhance the IR value, while oversampling and hybrid sampling have IR values that are almost equal to 1, ensuring that the dataset is as class-balanced as possible.

Applying various undersampling methods for the ReadPeak provided dataset, we show the resulting *AUC*s and *LogLoss*es for 4 different classifiers in Table 3.3. RUS showed the best performance with Random Forest and Decision Tree with an *AUC* of 0.666 and 0.59 respectively. While Naive Bayes's *AUC* of 0.607 IHT using Random Forest as a base learner.

For oversampling methods, ROS had the best performance for Logistic Regression, Naive Bayes. While for Random Forest and Decision Tree, SMOTE is the most suitable for performance. These are shown in Table 3.4. Naive Bayes had the highest mean *AUC* of 0.607.

On top of the hybrid methods provided by imlearn library [34] we manually manufacture two more hybrid methods based on RUS and SMOTE and ROS. These new sampling methods were inspired following the experiment in section 3.4.2.

For ensemble methods shown in Table 3.6, EasyEnsemble achieved the highest mean *AUC*, followed by RUSBoost.

We evaluate all the resampling methods using Random Forest. We chose Random Forest not only because it was the best-performing model with RUS, but also because it is the most sensitive to the applied sampling techniques. In Figure 4, blue represents the

Method	Imbalance Ratio	Majority Samples	Minority Samples
BaseLine	0.004	17853293	72540
RUS	1	72540	72540
ENN	0.004	17659810	72540
IHT_LR	0.004	17853293	72540
IHT_RF	0.005	15992013	72540
IHT_Ada	0.982	73887	72540
NM	1	72540	72540
NRC	0.004	17669435	72540
OSS	0.004	17843717	72540
TL	0.004	17843859	72540
ROS	1	17853293	17853293
ADASYN	0.99	17858792	17853293
SMOTE	1	17853293	17853293
SMOTEENN	0.964	17468821	16525698
SMOTETomek	1	17827350	17827350
BorderLineSMOTE	1	17853293	17853293
RUS&SMOTE	1	2176200	2176200
RUS&ROS	1	2176200	2176200

Table 3.2: Class distribution for data-level methods

Method	Logistic Regression		Decision Tree		Naive Bayes		Random Forest	
	AUC	*LogLoss*	*AUC*	*LogLoss*	*AUC*	*LogLoss*	*AUC*	*LogLoss*
BaseLine	0.500000	0.693147	0.526707	0.180339	0.603738	0.026056	0.547485	0.119183
RUS	0.5009305	0.69323475	**0.590349**	13.56434325	0.60582575	0.686635	**0.66606825**	1.03333775
ENN	0.5	0.693147	0.5294935	0.473606	0.6058525	0.026212	0.55096425	0.13126
IHT_LR_DT	0.5	0.693147	0.52787375	0.181053	0.6058525	0.02619775	0.548526	0.11978025
IHT_RF	0.5	0.693147	0.5329115	0.77200925	**0.606784**	0.026351	0.5629535	0.2122795
IHT_Ada	**0.51395**	0.694194	0.50601475	33.9179185	0.6045575	6.14427175	0.52633675	31.923336
NM	0.490552	0.73223225	0.49239475	34.15809675	0.39801575	1.4396265	0.47378175	32.5987575
NRC	0.5	0.693147	0.5288185	0.2365955	0.6058525	0.02621125	0.551633	0.12060425
OSS	0.5	0.693147	0.52808575	0.188301	0.6058425	0.02619875	0.5490755	0.11967825
TL	0.5	0.693147	0.5281335	0.1883825	0.6058425	0.02619875	0.54908475	0.11967675

Table 3.3: *AUC* and *LogLoss* results for undersampling methods

Method	Logistic Regression		Decision Tree		Naive Bayes		Random Forest	
	AUC	*LogLoss*	*AUC*	*LogLoss*	*AUC*	*LogLoss*	*AUC*	*LogLoss*
BaseLine	0.500000	0.693147	0.526707	0.180339	0.603738	0.026056	0.547485	0.119183
ROS	**0.569335**	0.693148	0.528662	0.297515	**0.607195**	0.686737	0.548611	0.257295
ADASYN	0.45561475	0.69314625	0.532529	0.443388	0.60546575	0.68677225	0.57850275	0.16850475
SMOTE	0.455315	0.69314625	**0.53267875**	0.440561	0.60572075	0.6867105	**0.57911525**	0.16841625
BorderLineSMOTE	0.5	0.693147	0.5305105	0.32716475	0.60438	0.68293625	0.5701645	0.155574

Table 3.4: *AUC* and *LogLoss* results for oversampling methods

Method	Logistic Regression		Decision Tree		Naive Bayes		Random Forest	
	AUC	*LogLoss*	*AUC*	*LogLoss*	*AUC*	*LogLoss*	*AUC*	*LogLoss*
BaseLine	0.500000	0.693147	0.526707	0.180339	0.603738	0.026056	0.547485	0.119183
SMOTEENN	0.500000	0.693147	**0.594370**	6.620541	0.605620	0.675026	**0.657399**	1.407990
SMOTETomek	0.45564425	0.69314625	0.5328025	0.44819025	0.60579925	0.6869205	0.57915525	0.17127225
RUS&SMOTE	**0.503822**	0.693147	0.569831	4.133750	0.605619	0.686698	0.652872	0.522852
RUS&ROS	0.482566	0.693147	0.556217	2.319643	**0.605901**	0.686387	0.647432	0.768557

Table 3.5: *AUC* and *LogLoss* results for hybrid-sampling methods

Method	*AUC*	*LogLoss*
Balanced Bagging	0.671409	0.741580
Balanced Random Forest	0.695569	0.655088
EasyEnsemble	**0.710696**	0.690277
RUSBoost	0.709652	0.690229

Table 3.6: *AUC* and *LogLoss* results for ensemble methods

baseline, green represents the undersampling methods, red is for the oversampling methods, and yellow represents the hybrid methods. The baseline AUC value is 0.547; it can be seen that the lowest value that appears in Near Miss is 0.474, the highest value appears in ROS, and its AUC value is 0.666. Observing the bar chart shows that the AUC displayed by the undersampling method has more significant fluctuations than other methods. Also, through the LogLoss result from tables 7, 8, and 9, they show that oversampling is more stable than other imbalanced learning techniques, and undersampling is the most unstable.

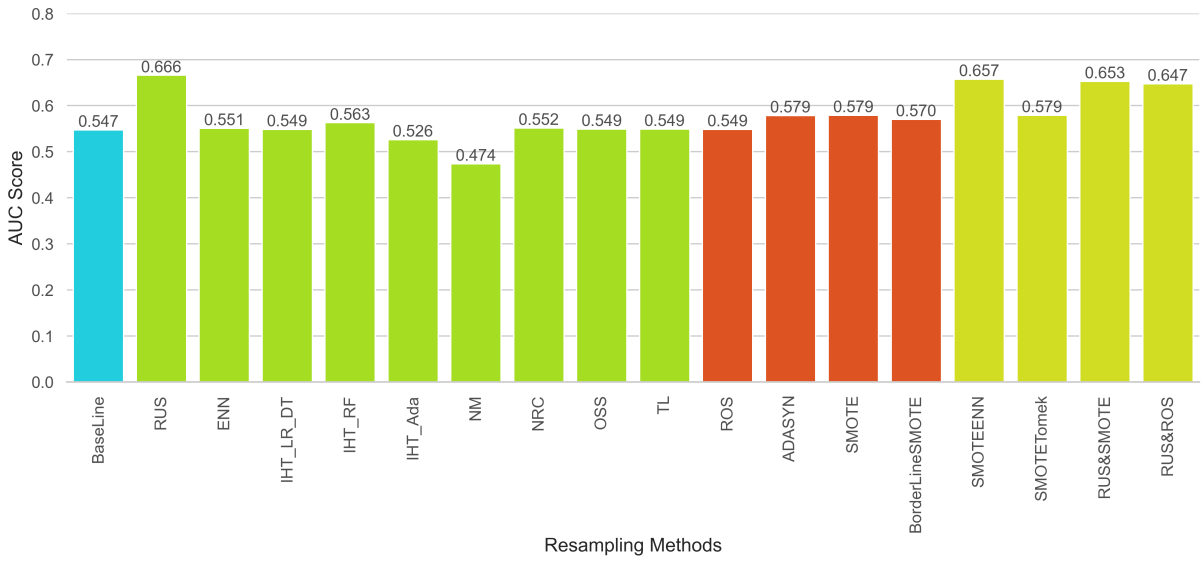
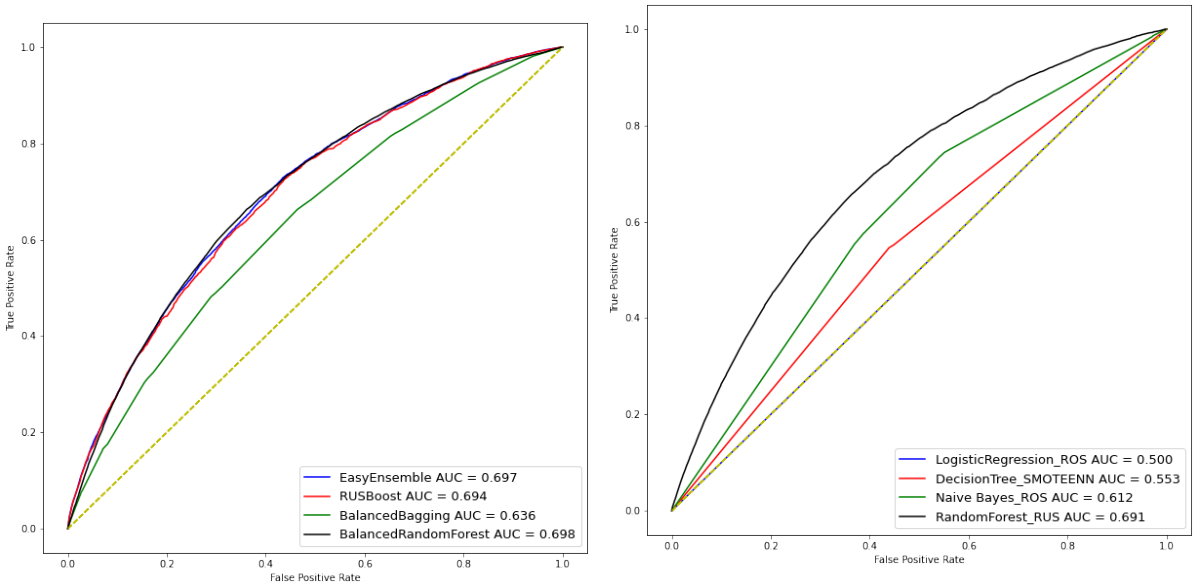


Figure 3.3: Comparison of sampling method on Random Forest

3.4.4 Generalization to Further Data

In this part we expose our last experiment. We take the best combinations between the used machine learning models and sampling technique, see if the results obtained trough cross validation can be generalized to future data. The best combinations found trough the previous experiments are: Random Forest with RUS, Decision Tree with SMOTEENN, Naive Bayes and Logistic Regression with ROS.

From Figure 3.4 and Table 3.7, we can see that Random Forest is the best performing model when it is put in a real situation. To reiterate Random Forest generalize well for practical use. While we can also see that for ensemble methods apart from Balanced Random Forest which *AUC* slightly improved, all the others performed slightly worse. Making Random Forest based models the best performing for this type of data.



(a) Ensembles Methods

(b) Base models with most suitable sampling method

Figure 3.4: ROC Curves for test data three days in the future from the training data

Method	*AUC*
EasyEnsemble	0.697
RUSBoost	0.694
Balanced Bagging	0.636
Balanced Random Forest	**0.698**

(a) Ensembles Methods

Method	*AUC*
Logistic Recression & ROS	0.500
Decision Tree & SMOTEENN	0.553
Naive Bayes & ROS	0.612
Random Forest & RUS	**0.693**

(b) Base models with suitable sampling method

Table 3.7: AUC for data three days in the future from the training data

3.5 Discussion

In this section, we will explore the implementation of class imbalance techniques in our study. First, we will examine the impact of data imbalance on machine learning algorithms

and the significance of feature importance. Next, we will provide a more detailed analysis of the results obtained from the combinations of different data sampling techniques and baseline classifiers used. We will also address the challenges presented by the dataset and how these were reflected in the results. Finally, we will discuss the relevance and significance of our contribution.

3.5.1 Data Unbalance Effect on Learning

In this study, different data-level sampling techniques were applied to four baseline classifiers, including Random Forest, Logistic Regression, Decision Tree, and Naive Bayes. The techniques used were data-level oversampling, undersampling, and hybrid methods. The results showed that Random Forest achieved the highest AUC with the RUS technique compared to the other classifiers, and also showed the greatest improvement from the baseline, going from 0.547 to 0.666. Naive Bayes, on the other hand, was the least affected by sampling methods. These results suggest that the Random Forest classifier is well-suited for the imbalanced advertisement data used in this study.

Despite the initial low AUC values of Logistic Regression and Decision Trees, it is worth noting that the AUC values of most models were improved through the application of resampling approaches. This highlights the effectiveness of these techniques in improving a model's classification abilities. Additionally, our study found that most undersampling and hybrid sampling approaches had higher average AUC values than oversampling techniques. These findings suggests that undersampling and hybrid sampling methods are likely to provide the best performance for CTR prediction, as they effectively handle the class imbalance in advertisement data while preserving the valuable information.

To further investigate the relationship between the sampling methods and the model's AUC, we used the imbalance ratio (IR) as a metric to measure the ability of resampling techniques to adjust the class distribution. The study's findings revealed that the oversampling and hybrid methods effectively achieved a near-equal distribution of classes in the dataset. However, the undersampling algorithms were not as successful in significantly reducing the majority class, suggesting that these methods may not be appropriate for a dataset with a high level of noise, like ours. It's worth noting that this conclusion is only based on the undersampling methods that were barely changed the data proportions, techniques such as RUS ans IHT_ada have proven to be more effective and suitable to be used on the same dataset.

In summary, the results of this study suggest that oversampling is generally more stable. However, for data that is as noisy and unbalanced as advertisement data, significant undersampling can outperform other approaches.

3.5.2 Sampling Techniques Effect on the Advertisement Data

The results of the study revealed that the feature importance of the native Advertisement dataset changed depending on the level of data sampling used. For example, when using the whole dataset, or when the IR was at 10%, it was observed that some features, such as city, were rejected by the Boruta algorithm. This suggests that the presence or absence of certain features can have a significant impact on the performance of the

model when working with imbalanced datasets. It's also worth noting that this is just one example, and the feature importance and Boruta algorithm may have different results in different dataset and sampling ratios. It's important to keep in mind that while the results of this study were specific to the native ads dataset and specific sampling ratios, the takeaways and insights gained from this experimentation can be applied to other datasets and scenarios. By going through this process, we not only demonstrate good practice in handling imbalanced data, but we also see how the imbalance ratio can affect the feature selection process. Additionally, by testing the performance of different classifiers and sampling methods, we gain a deeper understanding of the strengths and weaknesses of each approach, which can inform future decisions when working with imbalanced data. Therefore, even though results may vary when applying these methods to other datasets, the experimentation and analysis conducted in this study can provide valuable guidance for practitioners.

After comparing the performance of different sampling methods, it can be seen that RUS has good performance when combined with Decision Tree and Random Forest. In fact, RUS had the best results of all combinations with Random Forest. In the case of oversampling techniques, ROC performs well with Logistic Regression and Naive Bayes, while SMOTE seems to be more suited for Random Forest. Additionally, for ensemble methods, EasyEnsemble and RUSBoost classifiers performed well. However, when evaluating the performance of these models on data from a few days in the future, it was found that the Balanced Random Forest classifier outperformed the others. This underlines the importance of considering real-world scenarios and testing models on future data when evaluating their performance.

From the baseline, it was observed that Random Forest had the most improvement with an increase of 0.119 in the AUC metric. After each classifier was processed by the sampling method in the table, the AUC of the model was increased, with the exception of Naive Bayes which showed little to no improvement. Additionally, among all the classifiers, Logistic Regression performed the worst, while all other classifiers combined with RUS performed well. As such, the Random Forest model using RUS was determined to be more suitable for processing imbalanced advertisement datasets and achieving the highest AUC. Another undersampling technique, Near Miss, was tested, but it showed lower results than the AUC of the corresponding baseline classifier. Therefore, it can be concluded that Near Miss is not suitable for datasets with an imbalance rate of about 0.004, as it was used in this study.

In conclusion, the results of this research suggest that the Random Forest model using RUS is a suitable approach for handling highly imbalanced advertisement datasets and can achieve the highest AUC even when generalized to future data. This highlights the effectiveness of the RUS technique in balancing the data and improving the performance of the Random Forest model.

3.5.3 ReadPeak's Data

The dataset used in this study presents several challenges when it comes to predicting click-through rates. It is a native advertising dataset that is unique in its nature, with limited access to user demographic information and a highly imbalanced distribution of positive and negative samples and by nature a significant percentage of overlap. These unique characteristics are suspected to have impacted the performance of the algorithms used in our analysis.

We uncovered challenges in using logistic regression with our dataset. While this algorithm has been shown to perform well on CTR prediction tasks in different datasets, our results showed that it's AUC Improved in combination with ROS, but in a more general scale it did not perform well. One possible explanation for this discrepancy is that Unlike most datasets used in the literature, our dataset has limited access to user demographics, which may have affected the accuracy of the logistic regression model. Additionally, the dataset exhibits overlap between positive and negative samples, violating logistic regression's assumption of independent and identically distributed samples.

The challenges presented by the used data suggest that additional care must be taken when selecting data sampling and machine learning techniques for an effective prediction. The performance of logistic regression in this context serves as an example. We aimed to provide a balanced study focusing on data sampling techniques that can serve as a guideline and a starting point for future works.

3.5.4 Contribution

We have taken a systematic approach to feature engineering, selection, and data cleaning, outlining the complete steps taken in detail. Through experimentation using The Boruta algorithm to compute feature importance using three levels of sampling,it was found that the importance of features varies with different Imbalance Ratios.

Using Random Forest and RUS, it was shown that as the balance between positive and negative samples is improved, the performance of the algorithms also improves. However, there is a threshold of imbalance ratio beyond which the performance improvement becomes marginal. This highlights the importance of selecting the appropriate sampling technique for the dataset and algorithm in question.

The ultimate output of the sampling technique comparison resulted in RUS performing the best with Random Forest in terms of overall prediction AUC, it is important to note that this is not necessarily an obvious or expected result. While RUS has been successful in other studies, we noticed that there is a lack of rational on the selection of the algorithm and not enough evidence to support that it is the best technique for the problem at hand [42, 59].

In this work we provide valuable guidance for the development of effective click-through rate prediction models in native advertising by systematically comparing several practically used sampling techniques in combination with four different machine learning algorithms.

We would like to mention that the impact of these findings on machine learning in the field of native advertising. Advertisers and marketing professionals often deal with imbalanced datasets, where a small percentage of customers are responsible for the majority of conversions. In such scenarios, the ability to accurately identify and target potential customers becomes crucial for the success of an advertising campaign. The results of this study demonstrate that using the most appropriate sampling technique can help improve the performance of machine learning models on imbalanced native advertising datasets, leading to more effective targeting and better return on investment for advertisers. Additionally, by considering real-world scenarios and testing models on future data, advertisers can ensure that their models will continue to perform well over time. These insights can be used to inform the development of more effective and efficient advertising strategies.

3.6 Conclusion

In this chapter, we have investigated class imbalance techniques, including data-level and hybrid systems, to predict the click trough rate of advertisements. A dataset provided by ReadPeak's traffic in Finland for a week in June of 2022. The data had an imbalance ratio of 0.004. The imbalanced learning method was used to solve the problem of a skewed majority in prediction. This research discusses 19 imbalanced learning methods, including seven undersampling techniques, four oversampling techniques, four-hybrid resampling methods, and four ensemble systems. The class imbalance technology adjusts the majority or minority samples by discarding the majority samples, copying or synthesizing the minority samples to balance the categories in the dataset. In addition, foura classic classifiers (Logistic Regression, Random Forest, Decision Tree, Naive Bayes) combined with resampling techniques were used to train the dataset. The prediction results obtained using the classifier training pre-processing data (except for null values, etc.) were used as a baseline for comparison with models built using imbalance techniques. Additionally, the method used to evaluate the sampling technique was the imbalance ratio, and the index used to assess the classification ability of the model was AUC.

Further, using different levels of unbalance between the majority and minority samples we evaluated the features importance and the improvement of the AUC. For the current advertisement data when the undersampling was significant enough it was the more suitable approach. Interestingly, using RUS technology to process our data in the Random Forest model can achieve the highest AUC value.

This work's aim is to show how we can make use of sampling techniques in extremely skewed data such as native advertisement data. We would like to note that this methodology and results could be expanded to data related to cyber security; such as intrusion detection and click fraud, because of similarity in the data consistency of such use cases. In other words the results from the current work may apply to the mentioned cases.

We would like to point out that all the machine learning used in the current work have been used as is with no fine tuning. For future work we can explore pushing these methods further by finding more suitable hyper parameters that can improve the result further. Another point that we would like to explore in the future is that advertisement data is extremely noisy and adding some more cleaning to it could improve the performance of the sampling method studied in this work even further. What is more explore the effect of other sampling methods that we could not include in the scope of the this work given the extremely high computational power needed.

Part II

Enhancing Click Fraud Detection Through Advanced Machine Learning Algorithms

CHAPTER 4

Forecasting Click Fraud via Machine Learning Algorithms

In this chapter, we delve into the realm of click fraud detection, leveraging the knowledge gained from the previous chapter on data sampling techniques and CTR prediction. Building upon the foundations laid, we explore the application of machine learning algorithms to effectively identify and combat click fraud in online advertising campaigns.

The primary objective of this chapter is twofold. Firstly, we present an optimized approach for click fraud prediction using the powerful XGBoost algorithm. To achieve optimal performance, we employ grid search to fine-tune the hyperparameters of the XGBoost model, maximizing its potential in accurately identifying fraudulent clicks. Secondly, we introduce a novel approach of growing and pruning the XGBoost model, aiming to strike a balance between model complexity and predictive power.

By leveraging the insights gained from the preceding chapter, we aim to provide a robust and efficient solution for click fraud detection. Our methodology encompasses both the technical aspects of optimizing XGBoost through grid search and the strategic approach of growing and pruning the model to enhance its efficiency in identifying fraudulent click patterns.

Throughout this chapter, we will present experimental results showcasing the effectiveness of our proposed approaches. The findings will highlight the improved performance achieved by leveraging machine learning techniques, shedding light on the potential of advanced algorithms for combating click fraud in online advertising.

4.1 Introduction

Online advertising has witnessed significant advancements, providing businesses with unprecedented visibility and access to a broader customer base. As discussed earlier, the digital advertising landscape offers tremendous opportunities for businesses to showcase their products and services. However, these opportunities also attract malicious actors who engage in fraudulent activities. Click fraud, in particular, has become increasingly prevalent and poses a significant threat to the digital advertising industry. It is crucial to address this issue to safeguard businesses and advertisers from substantial financial losses. According to industry reports, click fraud resulted in billions of dollars of losses

globally, with estimates indicating that it accounted for around 9% of the total digital advertising spend in 2018 [68]. Alarming projections indicate that ad fraud could surpass $50 billion annually by 2025 [13]. These staggering figures highlight the urgent need for effective solutions in click fraud detection and prevention.

Click fraud encompasses intentional and malicious clicking on pay-per-click advertisements, depleting advertisers' budgets. Offenders can include competitors, web administrators, and organized fraud rings. Various techniques, such as brute force attacks, crowd-sourcing, reward traffic, click farms, hit inflation, and botnets, have emerged as common types of click fraud. Addressing click fraud is paramount to maintaining the trust of businesses and consumers within the digital advertising ecosystem.

Recognizing the significant impact of click fraud on the digital advertising industry, our research aims to contribute to the development of scalable, adaptable, and practical solutions for detecting and preventing click fraud. By providing a robust framework, we seek to mitigate financial losses, safeguard the integrity of the online advertising industry, and ensure the continued growth and sustainability of digital advertising. Our work strives to restore trust among businesses and consumers, fostering a secure and thriving online advertising ecosystem for years to come.

4.2 Data Exploration and Feature Engineering

In order to provide a comprehensive understanding of the dataset, we have chosen to include data exploration in this chapter. Since we have already covered data treatment in previous chapters, the techniques and approaches used for exploring the current data are not substantially distinct from those employed previously.

4.2.1 Data Exploration

In this subsection, we will be performing data exploration on our dataset to gain insights into possible trends, patterns, and correlations. By analyzing the data, we aim to uncover seasonality repetitions, identify significant correlations between different variables, and gain a deeper understanding of the underlying data structure. Our data exploration will serve as a foundation for the subsequent stages of our research, providing critical insights for the development and testing of our click fraud detection model.

4.2.1.1 IP Frequency

Typically, fraudulent clicks are associated with specific IP addresses, which serve as indicators of the malicious actor. Analyzing these IP addresses enables the creation of rules that can flag suspicious activity and prevent significant losses. Therefore, understanding the role of IP addresses in click fraud detection is vital in developing effective and accurate click fraud detection models. To this end, we conducted a frequency analysis of the IP addresses associated with fraudulent clicks in our dataset, focusing on the top 10 repeated IPs. Our aim was to analyze the patterns and frequency of click fraud associated with these IPs to gain insights into potential feature importance and inform the development of our click fraud detection model.

Based on the bar plot in Figure 4.1, we can see that there are two specific IPs that have a significantly higher number of attributed click fraud compared to the rest.

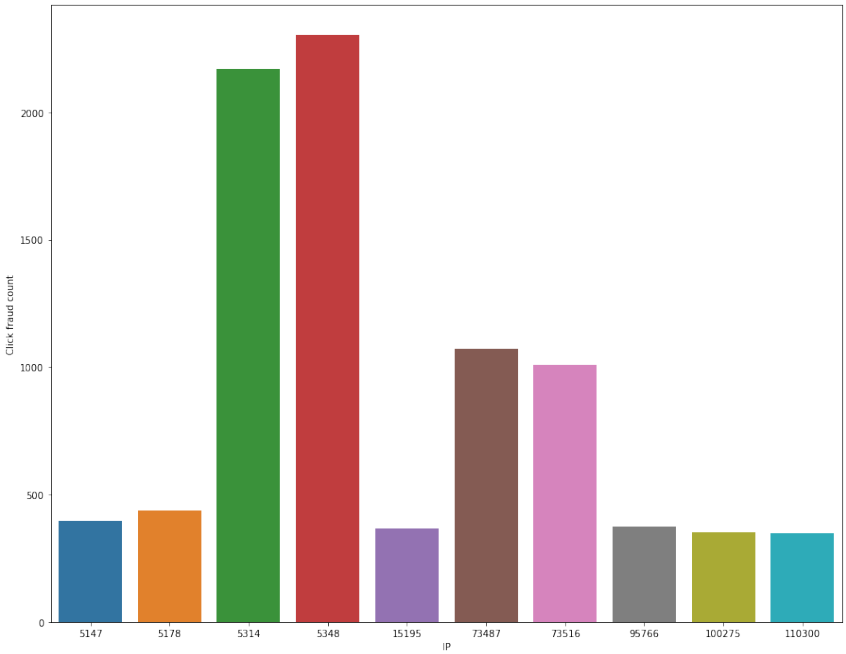


Figure 4.1: Top 10 IPs frequency

These two IPs have over 2000 click fraud attributed to them. On the other hand, the remaining eight IPs in the top 10 list have a much lower number of attributed click fraud, each fluctuating around 400 fraudulent clicks. This analysis suggests that the majority of fraudulent click activity is concentrated among a small number of IPs, highlighting the importance of monitoring these IPs to prevent fraudulent activity. The lower levels of click fraud attributed to the other IPs in the top 10 list also suggest the possibility of click fraud activity being more dispersed among a larger number of IPs.

4.2.1.2 Seasonality Analysis

Seasonality analysis is another potential exploration that could provide valuable insights into the click fraud data we have available. While the available data covers only a period of four days, there may still be seasonal patterns that can be identified and analyzed. While it is worth noting that four days may not be sufficient to observe long-term seasonality, it is still worth exploring the data to identify any potential patterns that may be present. Such insights can provide critical information for developing more accurate and effective click fraud detection models that account for seasonality and other temporal factors.

Figure 4.2 shows the seasonal decomposition of the click fraud time series data into its trend, seasonal, and residual components. The interpretation of the results depends on the patterns observed in these plots. Here's a general guideline for interpreting the results:

- ◇ Original plot: This plot shows the original time series data. We can identify that there is an overall trends and cycles in the data. In this case, the plot shows the total number of fraudulent clicks per hour over a period of 3 days.
- ◇ Trend plot: This plot shows the long-term trend in the data, after removing any seasonal or irregular components. If the trend is increasing, it indicates that the

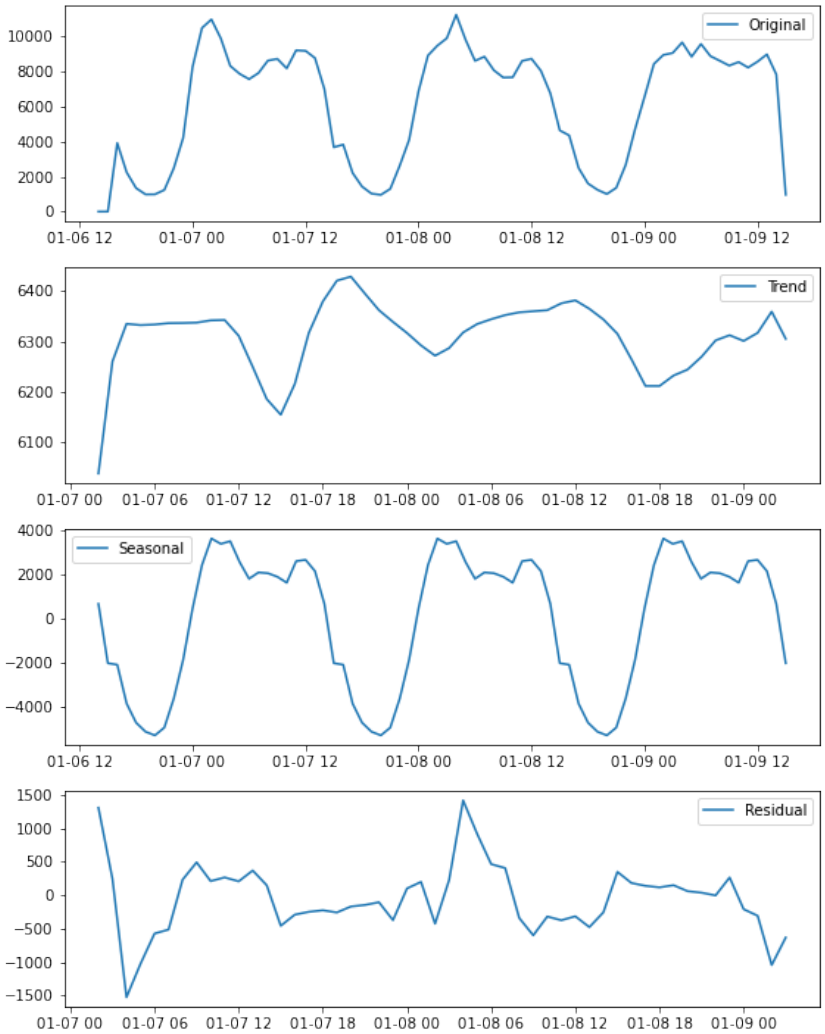


Figure 4.2: Seasonal behavior for click fraud

overall number of fraudulent clicks is increasing over time. If the trend is decreasing, it indicates that the overall number of fraudulent clicks is decreasing over time. If the trend is flat, it indicates that the overall number of fraudulent clicks is staying relatively constant over time. In this case, a fluctuating over time, with no clear overall direction, it suggests that there is no consistent trend in the number of fraudulent clicks over the 3-day period. While there may not be a clear long-term trend in the data, the fluctuations observed in the trend plot can still provide valuable insights into the click fraud patterns. The trend plot shows spikes and decrease in fraudulent clicks at a particular time of day, this could suggest that there is a specific window of vulnerability that click fraudsters are exploiting and invertely that the platform's anti-fraud measures are effective at detecting and preventing click fraud.

- ◇ Seasonal plot: This plot shows the seasonal pattern in the data, after removing any long-term trend or irregular components. In our case the seasonal pattern is regular and consistent, indicating that the number of fraudulent clicks follows a predictable pattern over a fixed period of time.

- ◇ Residual plot: This plot shows the remaining variation in the data, after removing the trend and seasonal components. As the residual plot shows random and unpredictable fluctuations around 0, this is explained that the trend and seasonal components have successfully explained most of the variation in the data.

4.2.1.3 Pairwise Correlation

A correlation matrix provides a tabular representation of correlation coefficients between different variables in a dataset, which helps in understanding the linear relationship between variables. However, it is crucial to note that correlation does not imply causation, and the strength of the correlation does not necessarily indicate the magnitude of the relationship's effect.

Figure 4.3 shows the heatmap of our dataset's features, including three engineered features that we will discuss in the next section. The correlation matrix provides valuable insights into the linear relationships between variables in the dataset, including their relationship with the target feature is_attributed, which indicates whether click fraud has occurred or not. Notable correlations include a weak positive correlation between is_attributed and both IP and App, suggesting a slight relationship between these variables and the presence of click fraud. Additionally, there is a strong positive correlation between device and OS, indicating that these two variables are closely related. However, the correlation of these variables with is_attributed is weak, suggesting they may not be significant predictors of click fraud.

Overall, while the correlation matrix can serve as a foundation for further analysis and investigation into the causes or more complex relationships between variables that could help in predicting click fraud, it is crucial to keep in mind that correlation is only one aspect of understanding the data.

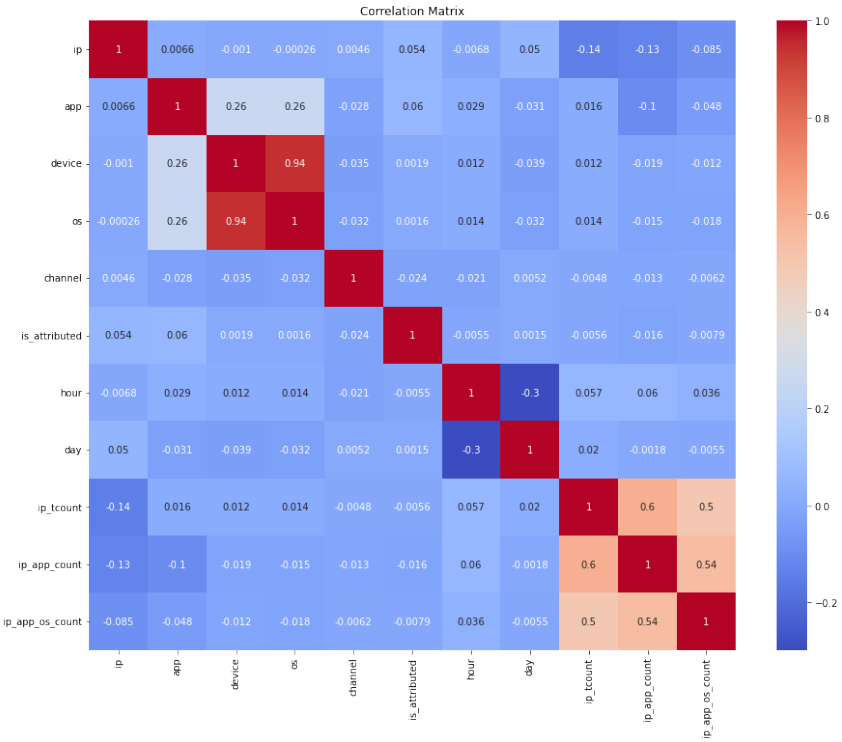


Figure 4.3: Correlation Matrix

4.3 Feature Engineering

In this research, we utilized feature engineering methods to extract novel features from the raw dataset. By utilizing our domain knowledge and expertise, we identified and extracted characteristics, attributes, and properties from the data that were not initially obvious. The IP was identified as one of the crucial features for classification in the dataset. Therefore, to improve the diversity and informativeness of features, we combined it with one or two other attributes to create a new dataset. we also removed the "attributed_time" column and separated the "click time" column into separate day and hour columns the say was not used as given our data split it will be one value. By doing so, we were able to refine the dataset and create more informative features.

Table 4.1: Engineered features description

Features	Format	Description
ipCountPerHour	int16	Count of similar IPs with the same hour
countIpApp	int16	Count of the unique combination of IP and App
countIpAppOs	int16	Count of the unique combination of IP and App and OS

4.4 Machine Learning Algorithms

In this section, we will present and discuss both the traditional machine learning algorithms utilized in our approach and the custom algorithm that we trained to compare

against our approach.

Firstly, we will introduce the traditional machine learning algorithms that form a crucial part of our methodology. These established algorithms have been widely used in various applications and have proven to be effective in solving similar problems. We will delve into the details of each algorithm, explaining their underlying principles, strengths, and limitations.

Furthermore, we will introduce the custom algorithm that we trained specifically for this project. This algorithm was developed to address the unique requirements and challenges of our problem domain. We will provide insights into the design, training process, and performance evaluation of this algorithm.

By including both the traditional machine learning algorithms and our custom-trained algorithm, we aim to provide a comprehensive comparison that showcases the strengths and potential improvements of our approach. This analysis will contribute to a better understanding of the capabilities and limitations of different methodologies in tackling the problem at hand.

4.4.1 Naive Bayes

Naive Bayes is a popular probabilistic machine learning algorithm based on Bayes' theorem. It assumes that the features are conditionally independent given the class variable. Naive Bayes is particularly effective for text classification tasks.

Given a set of training data (X, y), where X represents the feature vectors and y represents the class labels, Naive Bayes calculates the posterior probability of a class given the features using Bayes' theorem:

$$P(y|X) = \frac{P(X|y) \cdot P(y)}{P(X)}$$

where $P(y|X)$ is the posterior probability of class y given the features X, $P(X|y)$ is the likelihood of observing the features X given the class y, $P(y)$ is the prior probability of class y, and $P(X)$ is the evidence probability.

To estimate the likelihood $P(X|y)$, Naive Bayes makes the naive assumption that the features are conditionally independent given the class. This assumption allows us to factorize the likelihood as:

$$P(X|y) = \prod_{i=1}^{n} P(x_i|y)$$

where n is the number of features and x_i represents the i-th feature value.

To handle continuous features, Naive Bayes assumes a probability distribution for each feature. Commonly used distributions include Gaussian (for continuous features) and Multinomial (for discrete features).

During the training phase, Naive Bayes calculates the prior probabilities $P(y)$ based on the class distribution in the training data. It also estimates the conditional probabilities $P(x_i|y)$ for each feature given the class using maximum likelihood estimation or other smoothing techniques.

During prediction, Naive Bayes calculates the posterior probability for each class and selects the class with the highest probability as the predicted class.

Naive Bayes is known for its simplicity, efficiency, and ability to handle high-dimensional data. Although the assumption of feature independence may not hold in many real-world scenarios, Naive Bayes often performs well and serves as a strong baseline for classification tasks, particularly in text classification and spam filtering.

4.4.2 Logistic Regression

Logistic Regression is a widely used classification algorithm that models the relationship between a set of independent variables (features) and a binary dependent variable (outcome). It estimates the probability of the binary outcome based on the input features.

The logistic regression model uses the logistic function, also known as the sigmoid function, to transform the linear combination of the features into a value between 0 and 1, representing the predicted probability of the positive class. The logistic function is defined as:

$$f(x) = \frac{1}{1 + e^{-x}}$$

where x represents the linear combination of the input features and model parameters.

The linear combination of the features can be expressed as:

$$x = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$$

where $\beta_0, \beta_1, \ldots, \beta_n$ are the model coefficients, and $x_1, x_2, \ldots, x_n$ are the input features.

To estimate the model coefficients, logistic regression maximizes the likelihood function. The likelihood function measures the probability of observing the training data given the model parameters. The goal is to find the values of $\beta_0, \beta_1, \ldots, \beta_n$ that maximize the likelihood.

To avoid overfitting, regularization techniques like L1 or L2 regularization can be applied. Regularization adds a penalty term to the likelihood function, which discourages large coefficient values and promotes simpler models.

During prediction, the logistic regression model uses a decision threshold (typically 0.5) to classify instances as either the positive or negative class based on the predicted probabilities. Adjusting the decision threshold can trade off precision and recall, depending on the specific requirements of the problem.

Logistic regression is widely used due to its interpretability, simplicity, and efficiency. It is a powerful algorithm for binary classification tasks and can be extended to handle multiclass classification problems using techniques like one-vs-rest or multinomial logistic regression.

4.4.3 Decision Trees

Decision Trees are versatile and widely used machine learning algorithms for both classification and regression tasks. They learn a hierarchical structure of if-else conditions based on the input features to make predictions.

A decision tree consists of internal nodes, which represent feature tests, and leaf nodes, which represent class labels or predicted values. Each internal node applies a feature test on a specific feature, branching to different child nodes based on the test outcome.

To build a decision tree, different splitting criteria can be used. One commonly used criterion is the Gini impurity, which measures the impurity or randomness of the class labels at a node. The Gini impurity for a node can be calculated as:

$$Gini(p) = 1 - \sum_{i=1}^{c} p_i^2$$

where p_i represents the proportion of samples belonging to class i at a node, and c denotes the total number of classes.

Another commonly used splitting criterion is entropy, which measures the average amount of information needed to identify the class label at a node. The entropy for a node can be calculated as:

$$Entropy(p) = -\sum_{i=1}^{c} p_i \log_2(p_i)$$

where p_i represents the proportion of samples belonging to class i at a node, and c denotes the total number of classes.

The decision tree algorithm recursively splits the data based on the chosen criterion until certain stopping criteria are met. These stopping criteria can include reaching a maximum tree depth, achieving a minimum number of samples at a leaf node, or a lack of further improvement in impurity reduction.

During prediction, a new instance is traversed down the decision tree by following the feature tests at each internal node until a leaf node is reached. The class label or predicted value associated with the leaf node is then assigned as the final prediction.

Decision Trees have several advantages, including their interpretability, ability to handle both categorical and numerical features, and capturing non-linear relationships. However, they can be prone to overfitting and may not generalize well to unseen data. Techniques like pruning, ensemble methods (e.g., Random Forests), and regularization can be applied to mitigate overfitting and enhance the performance of decision trees.

4.4.4 Random Forest

Random Forest is an ensemble learning method that combines multiple decision trees to make predictions. Each decision tree in the Random Forest is built independently, using a random subset of the training data and a random subset of the input features. The final prediction is obtained through a voting or averaging process among the individual trees.

Random Forest operates by constructing decision trees using the same principles as mentioned earlier. However, it introduces additional randomness by using bootstrap samples from the training data and randomly selecting a subset of features at each node.

To build a Random Forest, a specified number of decision trees are trained independently using different bootstrap samples from the training data. Each tree is constructed by randomly selecting a subset of features at each node. This randomness promotes diversity among the trees, reducing overfitting and enhancing generalization.

During prediction, the Random Forest combines the outputs of all the individual trees. For classification tasks, the class with the majority of votes is selected as the final prediction. In regression tasks, the predictions from all trees are averaged to obtain the final output.

The randomness introduced in Random Forest offers several advantages. It helps to capture different aspects of the data and reduces the risk of relying too heavily on any particular feature. Moreover, it provides estimates of feature importance, allowing for insights into the relative importance of different features in the prediction process.

Random Forest is a powerful and widely used algorithm due to its ability to handle high-dimensional data, handle missing values, and provide estimates of feature importance. The randomization aspects of Random Forest contribute to its robustness and make it less prone to overfitting compared to a single decision tree model.

4.4.5 Xgboost

XGBoost [17] is a gradient boosting algorithm that is primarily used for supervised learning tasks such as classification and regression. The algorithm works by building an ensemble of decision trees, where each tree is trained to correct the errors made by the previous tree. The training process begins by fitting an initial decision tree to the data, and then, in subsequent iterations, new trees are added to the ensemble and trained to correct the residual errors made by the previous trees. The final predictions are then made by taking the average of the predictions from all the trees in the ensemble. XGBoost has several key features that make it a popular choice for machine learning tasks. Firstly, it uses a technique called gradient boosting [23] which is efficient and can handle large datasets. Secondly, it uses a technique called regularization which helps to prevent overfitting. Thirdly, it has several built-in parallel processing capabilities, which makes it highly scalable.

Here is a step-by-step explanation of the XGBoost:

1. Initialize the model: The first step is to initialize the model with a set of parameters such as the number of trees, the maximum depth of the trees, the learning rate, etc.

2. Create the first tree: The algorithm starts by creating a decision tree model on a sub-sample of the data, known as the "negative gradient" of the loss function.

3. Add more trees: The algorithm continues to add more trees to the model, with each tree trying to correct the mistakes made by the previous tree. The new trees are created by fitting to the negative gradient of the loss function, which is calculated from the residuals of the previous tree.

4. Update the model: As new trees are added to the model, the algorithm updates the model by combining the predictions of all the trees in the ensemble.

5. Repeat steps 2-4: The algorithm continues to repeat steps 2-4 until the specified number of trees is reached or a stopping criterion is met.

6. Make predictions: Once the model is trained, it can be used to make predictions on new data by taking the average of the predictions of all the trees in the ensemble.

Algorithm 1 can be summarizes the XGBoost process:

Algorithm 1 XGBoost Algorithm

```
procedure XGBoost
Input: Traning data and hyper parameter
Output: A trained model
    Initialize the model with a set of parameters
    Create the first tree
    while stopping criterion not met do
        Add more trees
        Update the model
    end while
    Make predictions
end procedure
```

4.5 Grid Search Optimized XGBoost

In this section, we showcase the outcomes of our comprehensive experimentation, aimed at assessing the efficacy of the proposed methodologies and models. Our objective is to offer an in-depth analysis of the achieved results while conducting a thorough comparison between the performance of our proposed methodologies and existing methods. Through this rigorous evaluation, we aim to provide valuable insights into the strengths and weaknesses of our approaches, enabling a robust assessment of their effectiveness in addressing the given problem.

4.5.1 Hyper Parameter Tuning

In order to optimize the performance of our XGBoost model, we implemented a technique called Hyperparameter Tuning. We used GridSearchCV, which is a method of searching for the optimal set of hyperparameters that results in the best performance of the model. The process of GridSearchCV involves specifying a set of hyperparameters and their possible values, and then training and evaluating the model for each combination of hyperparameters. In our case, we used a list of hyperparameters such as 'min_child_weight', 'gamma', 'subsample', 'colsample_bytree', 'max_depth', 'num_boost_round' and 'learning_rate' with specific range of values for each. This allowed us to systematically explore different combinations of hyperparameters and select the set that resulted in the best performance of the model, thus improving the accuracy and efficiency of the XGBoost algorithm. And the result of the grid search landed on these parameters:

- gamma : 1.0
- max_depth: 5,

- ◇ min_child_weight: 1,
- ◇ num_boost_round : 500
- ◇ learning_rate: 0.2

These parameters were used to train the final model and were found to provide the best results on the test dataset.

It is worth noting that grid search can be a time-consuming process, and this can impact the overall performance of the system. Specifically, the time required to execute a grid search depends on the size of the search space, which can increase rapidly with the number of hyperparameters and the range of values. As a result, a very large search space could lead to grid search taking an impractical amount of time to complete.

Despite the potential drawbacks and limitations of grid search, it remains a very useful method for hyperparameter tuning, especially when the practitioner has the necessary computational resources and time to execute it.

Moreover, the selection of hyperparameters based on previous experience and empirical results, as well as the practitioner's expertise in choosing the hyperparameter space to search, can help to minimize the computational cost and time required for grid search. In this way, grid search can still be an effective and practical method for hyperparameter tuning in a wide range of machine learning applications, including the prediction of click fraud using XGBoost.

4.5.2 Experiment Results

When it comes to machine learning, there is no single algorithm that is guaranteed to work in every situation. Instead, it is crucial to explore different algorithms and evaluate their performance using various metrics. In this section, we compare the XGBoost algorithm with other popular machine learning algorithms, such as logistic regression, decision tree, and random forest. Additionally, we also compare the proposed boosting algorithm with other established algorithms that have been used in similar studies in the literature, to see how it performs in comparison. This allows us to gain a better understanding of the strengths and weaknesses of the XGBoost algorithm and determine its suitability for a given task. By comparing XGBoost with other algorithms, we can also identify which algorithm performs best under different conditions and with different types of data.

To evaluate the performance of the different algorithms, we use two metrics: the *AUC* Score and *LogLoss*. The closer the *AUC* score is to 1, the better the algorithm will be. The results of the comparison are shown in Figure 4.2, where we find that XGBoost reaches an *AUC* score of 0.96. The performance of the Random Forest algorithm is the closest to that of XGBoost among the algorithms compared.

Table 4.2: AUC and *LogLoss* comparison to classical machine learning algorithms

	XGBoost	Logistic regression	Decision tree	Random forest
AUC	**0.96**	0.81	0.86	0.94
LogLoss	**0.15**	0.53	4.00	0.21

From Table 4.2, it is clear that our model compared to the rest of the learners performs better. While just from a first look to Table 4.3, it is clear that the proposed work does not perform significantly better then the other models in fact the result is basically the same as the ETCF model in terms of *AUC*.

Table 4.3: AUC Comparison To Previous Work

	SVM [65]	ETCF [65]	**Proposed Model**
AUC	0.93	0.96	**0.96**

From a first look in Table 4.3 it seems that our approach perform just as well to the ETCF model, we will discuss the meaning such result in the next secession.

4.5.3 Discussion

Based on the results presented in the previous section, it is clear that the XGBoost algorithm outperforms the other algorithms in terms of both *AUC* and *LogLoss*. With an *AUC* of 0.96 and a *LogLoss* of 0.15, the XGBoost algorithm demonstrates the highest level of accuracy and lowest level of error.

In comparison, the logistic regression algorithm has an *AUC* of 0.81 and a *LogLoss* of 0.53, indicating that it is less accurate and more prone to error than XGBoost. The decision tree algorithm has an *AUC* of 0.86 and a *LogLoss* of 4.00, which is also lower than the XGBoost algorithm.

The Random Forest algorithm is the closest competitor to XGBoost, with an *AUC* of 0.94 and a *LogLoss* of 0.21. While it is still not as accurate as XGBoost, it is still a good choice for certain types of problems.

Overall, these results suggest that the XGBoost algorithm is a suitable choice for classification problems with imbalanced data, particularly when the goal is to accurately identify fraudulent clicks. The results also highlights the importance of carefully preprocessing the data and avoiding data leakage in order to achieve the best possible results. It is worth noting that we performed hyperparameter tuning on the XGBoost algorithm using grid search, also helped to optimize the performance of the model even further.

We also compared our work to available literature, and from a first look it seems to perform similar to ETCF [65]. It is hypothesized that the difference in results between this work and previous work is primarily attributed to the preprocessing step. Specifically, the care taken in this work to ensure that the test set accurately emulates a real-world environment, as opposed to previous work where data leakage may have occurred due to the preprocessing and generation of new variables using the entire dataset prior to the train/test split. Additionally, it is hypothesized that the use of randomly selected rows for the test data in previous work, rather than the last rows of the original data, may have also contributed to the discrepancy in results.

It is true that some of the results may appear to be similar, and it may be tempting to question the value of this research. However, it is important to note that even a small improvement in *AUC* can lead to significant real-world benefits. Furthermore, our approach has been shown to be more stable, as evidenced by the better *logloss*. This suggests that our model is better able to handle the complexity of the data and make more accurate predictions. Additionally, our approach places a strong emphasis on reproducing

a realistic setting in order to avoid data leakage and accurately mimic the distribution of the original data. This is an important consideration when evaluating the performance of a model in a real-world scenario.

4.6 Growing and Pruning Approach to Click Fraud Prediction

In this section, we introduce a novel method specifically optimized for real-life continuous integration in the context of click fraud prediction. Our approach is designed to address the unique challenges and requirements of click fraud detection in real-time systems, where timely and accurate predictions are crucial. We present the detailed architecture and implementation of our method, highlighting the techniques employed to ensure efficient and seamless integration with continuous deployment pipelines. Furthermore, we provide comprehensive evaluation results that demonstrate the effectiveness and practical viability of our method in accurately detecting click fraud instances, enabling proactive mitigation and safeguarding the integrity of online advertising campaigns.

4.6.1 Working Scenario

In a realistic environment, detecting click fraud is an ongoing task that requires constant adaptation to new data and evolving fraudulent techniques. The current scenario involves collecting more data daily, if not hourly, and the fraud evolves with the passing time. If we have a machine learning algorithm only trained on previous data, it means that it's harder for it to adapt and will be prone to more errors as time passes. The naive approach is to try to train the models on all the data again and again to keep up with the evolving environment. This might provide the best prediction, but it does not guarantee the best performance, especially if older data becomes outdated. In this case, the proposed approach is to train a machine learning algorithm incrementally on the data, for example, daily. Before each incremental training, we prune the trees and nodes with the least contribution to the model and then trigger incremental training. This approach will save both on the training time and the model size, making it more efficient and effective for detecting click fraud in a realistic environment.

4.6.1.1 Pruning

The first step of our approach involves implementing pruning, a technique commonly used in decision tree-based machine learning algorithms like XGBoost. Pruning aims to improve the accuracy and efficiency of the model by removing unnecessary branches or nodes in the decision tree that do not significantly contribute to the model's performance. The process of pruning leads to a simplified model with reduced complexity and size, which results in faster training and prediction times. Pruning is essential to achieving better generalization and reducing the risk of overfitting, which occurs when the model becomes too complex and overfits the training data, leading to poor performance on new, unseen data. Typically, pruning is done after the initial training phase, and the decision on which branches or nodes to prune is based on a measure of their contribution to the model's accuracy or importance. Algorithm 2 describes the process of pruning.

Algorithm 2 Pruning in XGBoost

Require: Ensemble of decision trees $\{T_i\}$, training data (X, Y)
Ensure: Pruned ensemble of decision trees $\{T_i'\}$
procedure PRUNE($\{T_i\}, (X, Y)$)
  for $i = 1$ to n **do**
    Compute the reduction in loss for each leaf node in T_i
    Remove the leaf node that results in the smallest increase in loss
    Repeat until the desired level of pruning is achieved
  end for
  Compute the predictions of the pruned trees $\{T_i'\}$
  return $\{T_i'\}$
end procedure

4.6.1.2 Growing

The second step of our approach, called growing, involves incremental training. Incremental training, also known as online learning, is a technique used in machine learning algorithms that enables the model to learn from new data as it becomes available, without retraining the entire model from scratch. In the context of XGBoost, incremental training involves updating the existing ensemble of decision trees with new trees that are trained on the new data.

The process of incremental training in XGBoost entails adding new trees to the existing ensemble and updating the predictions by taking the weighted sum of the predictions from all the trees. The weight assigned to each tree is determined by its performance on the new data.

The key advantage of incremental training is that it allows the model to adapt to changes in the data over time, without the need to retrain the entire model from scratch. This is particularly useful in scenarios where the data is continuously evolving, such as in fraud detection, where new types of fraud may emerge over time. Additionally, incremental training can be more efficient than retraining the entire model, as it allows the model to reuse the existing trees that are still relevant to the new data, thus reducing training time and memory usage. In Algorithm 3, we show the steps done in Incremental training.

Algorithm 3 Incremental Training in XGBoost

Require: Existing XGBoost model $\{T_i\}$, New data $(X, Y)_{\text{new}}$
Ensure: incrementally trained ensemble of decision trees $\{T_i'\}$
$(X, Y) \leftarrow$ Training data used to build $\{T_i\}$
$\{T_i\}_{\text{old}} \leftarrow \{T_i\}$ ▷ Make a copy of the old model
$\{T_i\} \leftarrow$ Train $\{T_i\}$ on $(X, Y) \cup (X, Y)_{\text{new}}$ ▷ Update the model with new data
for $i = 1$ to $|\{T_i\}| - |\{T_i\}_{\text{old}}|$ **do** ▷ Add new trees to the model
  $t \leftarrow$ The i-th new tree in $\{T_i\}$
  $w \leftarrow$ Weight assigned to the new tree
  $\{T_i\}_{\text{old}} \leftarrow$ Add tree t to $\{T_i\}_{\text{old}}$ with weight w
end for
return $\{T_i\}_{\text{old}}$

4.6.2 Experiment and Analysis

This section reports the findings of our experiment that aimed to assess the effectiveness of the proposed approaches and models.

4.6.2.1 Chosen Hyper Parameters

Based on several experiments conducted, we have fixed several parameters in the proposed model.

XGBoost chosen hyper parameters for bother the base model and the rounds of incremental training will be:

- ◇ max_depth: 6,
- ◇ min_child_weight: 1,
- ◇ num_boost_round : 100
- ◇ learning_rate: 0.2

As for the pruning round the hyper parameters are:

- ◇ gamma : 0.2
- ◇ max_depth: 3,
- ◇ min_child_weight: 1,
- ◇ num_boost_round : 50
- ◇ learning_rate: 0.2

It should be noted that the hyperparameters selected in this study may not be the most optimal for the problem at hand. Further experimentation may be required to determine the best hyperparameters, and techniques such as grid search and random search could be implemented. However, the focus of this study is to explore the efficacy of the grow-and-prune approach in a practical situation, and optimizing the hyperparameters was deferred to future work.

4.6.2.2 Experimental Results

To achieve successful outcomes in machine learning, it is important to acknowledge that there is no one-size-fits-all algorithm that can work efficiently in every scenario. Therefore, it is essential to examine various algorithms and assess their performance using diverse metrics. In this section, we evaluate the performance of the prune-and-grow algorithm by comparing it to XGBoost and other prevalent machine learning algorithms such as logistic regression, decision tree, and random forest trained on all the data. This approach enables us to obtain a more comprehensive understanding of the effectiveness of the prune-and-grow methodology and its applicability to the specific task at hand. By comparing its performance with other algorithms, we can assess whether the benefits of using prune-and-grow outweigh its limitations.

In order to assess the effectiveness of the various algorithms, we have utilized two metrics: the AUC Score and LogLoss. An AUC score closer to 1 indicates superior performance of the algorithm. As shown in Table 4.4, the results of the comparison reveal that XGBoost trained on all the data achieved an AUC score of 0.96. Furthermore, we observed that the proposed approach is not far behind with an AUC of 0.95 slightly better than Random Forest algorithm had the closest performance to XGBoost among the algorithms that were compared.

While in Table 4.5 we present a comparison between our proposed approach and established related works in the field of click fraud detection. As expected, our approach does not perform as well as some of the existing methods. However, it is important to note that we have not yet optimized our hyperparameters, and our primary focus in this study is not to achieve the best possible performance. Instead, we are more concerned with the practicality of our approach in real-world scenarios.

Table 4.4: AUC and *LogLoss* comparison to classical machine learning algorithms

	Grown-and-Prune	XGBoost [73]	Logistic regression	Decision tree	Random forest
AUC	**0.95**	0.96	0.81	0.86	0.94
LogLoss	**0.16**	**0.15**	0.53	4.00	0.21

Table 4.5: AUC Comparison To Previous Work

	SVM [65]	ETCF [65]	CFXGB [85]	Stacked Model [71]	**Grown-and-Prune**
AUC	0.93	0.96	0.989	0.972	**0.95**

4.6.2.3 Prediction Performance

In this study, our primary goal was to propose a practical approach to tackle a real-world scenario using the grow-and-prune methodology. We aimed to establish the benefits and limitations of this approach by comparing it to other popular algorithms that were trained on the entire dataset. By doing so, we sought to determine the effectiveness of our approach in achieving high accuracy while reducing the computational cost of training the model. To evaluate the performance of the different algorithms, we used two widely accepted metrics - AUC Score and LogLoss.

Our proposed approach, Grow-and-Prune, showed promising results in this study when compared to other popular machine learning algorithms. Specifically, when compared to the traditional trained XGBoost, Grow-and-Prune achieved a relatively high AUC score of 0.95. This suggests that Grow-and-Prune is a viable approach for practical machine learning applications.

Furthermore, when compared to other algorithms such as Logistic Regression and Decision Tree, the Grow-and-Prune approach outperformed them with AUC scores of 0.81 and 0.86, respectively. However, Random Forest achieved an AUC score of 0.94, which is slightly lower than the Grow-and-Prune approach.

To ensure the validity of our proposed approach, we conducted a comparison against previous studies, including state-of-the-art solutions, to evaluate its performance. The results indicate that our approach performs competitively when compared to these works. However, it is important to note that, with the exception of [71] study, the other works focused on only a limited part of the dataset and did not simulate a real-world scenario of training the models on historical data and testing on unseen "future" data. In contrast, our study emphasized the real-world applicability of the grow and prune approach, where the model is trained on past data and tested on recent data. This approach is particularly relevant in practical implementations of click fraud detection systems in real-time advertising campaigns. By prioritizing the practicality and scalability of our approach, we believe that it can provide a more effective and feasible solution to the problem of click fraud.

Overall, these results indicate that the Grow-and-Prune approach can perform competitively with other popular machine learning algorithms and offer additional benefits such as reduced computational complexity and faster training times. That is to say that our grow and prune approach offers a balance between performance and practicality by providing a solution that is scalable and can keep up with the ever-evolving nature of click fraud.

4.6.2.4 Model Complexity

We compared our proposed method, the grow-and-prune approach, with the traditionally trained XGBoost algorithm, focusing on computational and space complexity. When training XGBoost to adapt to changing trends in the data, the model must be retrained, which has a computational complexity of $O(NMD * log(N))$, where N, M, and D represent the number of samples, features, and boosting rounds, respectively. If this retraining occurs daily, it introduces a time factor that increases both training time and model complexity. Furthermore, the space complexity of the XGBoost model is $O(KN)$, where K is the number of trees, and it increases as the number of samples (N) grows over time.

In contrast, the grow-and-prune approach maintains nearly constant space complexity since pruning eliminates the least relevant trees for prediction, and the model is rebuilt to a number close to the original model size. The training complexity also does not increase significantly, as we only train on new data each day rather than retraining on all data from scratch. As a result, the training complexity of the grow-and-prune method is estimated at

$$O(NMDlog(N) + STlog(T))$$

where S is the number of trees in the existing model, and T is the maximum depth of those trees. In summary, the grow-and-prune approach provides a more efficient and scalable solution for addressing changing trends in data when compared to traditional XGBoost training.

4.7 Conclusion

In summary, this chapter has presented a click fraud prediction classification approach based on the XGBoost algorithm, making significant contributions to the field of click

fraud detection. The research has effectively utilized feature engineering and preprocessing techniques to extract valuable information from the available data. By adopting incremental training and pruning methods, the proposed approach has been optimized in terms of performance, space complexity, and computational complexity.

Furthermore, the study has conducted a comprehensive comparison with other classical machine learning models, including Random Forest, Logistic Regression, Decision Tree, and traditional XGBoost. While traditional XGBoost exhibited higher performance, the proposed approach still demonstrated competitive results. Notably, the adaptability and scalability of the proposed approach provide valuable advantages for practitioners in the field.

To further enhance this research, future work should focus on optimizing the hyperparameters of the incremental training, pruning, and base XGBoost model. By fine-tuning these parameters, there is potential to achieve even better performance and efficiency. Additionally, expanding the application of the proposed approach to diverse datasets can establish its robustness and wider applicability. Considering datasets with varying characteristics, such as different levels of class imbalance or data distribution, would provide a more comprehensive evaluation of the approach.

CHAPTER 5

Mastering Click Fraud Detection: Uniting Algorithms in a Stacked Prediction Framework

In this chapter, we explore the realm of click fraud detection and ensemble methods. Our focus lies on harnessing the power of a stacked prediction framework, where we unite multiple algorithms to create a robust defense against click fraud. By combining the strengths of diverse algorithms and leveraging their collective insights, we aim to elevate click fraud detection to new levels of accuracy and effectiveness. Throughout this chapter, we will thoroughly investigate the intricacies of ensemble methods, highlighting their inherent advantages and demonstrating how they can significantly improve the accuracy of click fraud prediction models.

5.1 Introduction

Ensemble methods have gained significant attention in the field of machine learning due to their ability to improve predictive performance by combining multiple individual models. These methods have been successfully applied in various domains, showcasing their effectiveness in solving complex problems. In this section, we will provide an introduction to ensemble methods and highlight their proven improvement in performance across several fields.

Ensemble methods operate on the principle that a combination of diverse models can often achieve better results than a single model. By aggregating the predictions of multiple models, ensemble methods can mitigate the limitations of individual models and exploit their collective wisdom. This diversity can be achieved through different means, such as using different algorithms, varying model architectures, or training on different subsets of the data.

The effectiveness of ensemble methods has been demonstrated in numerous fields, including computer vision, natural language processing, and finance. In computer vision, ensemble methods have been utilized to improve image classification accuracy by combining the outputs of multiple convolutional neural networks (CNNs). Similarly, in natural language processing, ensemble methods have been employed to enhance sentiment analysis and text classification tasks by combining the predictions of multiple classifiers.

In finance, ensemble methods have been used to improve stock market predictions and portfolio management by aggregating the forecasts of multiple models.

Several studies have shown the performance benefits of ensemble methods compared to individual models. For instance, in the field of computer vision, studies by Krizhevsky et al. [44] and Simonyan and Zisserman [78] demonstrated the superiority of ensemble methods, such as deep ensembles, in achieving state-of-the-art results on image classification benchmarks. In natural language processing, research by Severyn and Moschitti [77] and Zhou et al. [98] highlighted the improved performance of ensemble models in sentiment analysis and text classification tasks. Furthermore, in the field of finance, studies, such as, Tsantekidis et al. [87] showcased the advantages of ensemble methods in improving stock market predictions and portfolio management strategies.

There is several different ensemble methods, such as bagging, boosting, and stacking. we in this chapter will focus on stacking and discuss the underlying principles and applications. We will explore how this methods can be effectively employed to address the challenges of click fraud detection and enhance the accuracy of prediction models. By leveraging the power of ensemble methods, we aim to develop robust and reliable solutions for combating click fraud and improving the overall performance of online advertising systems.

5.2 Stacking Algorithm

Stacking was first introduced by Wolpert [91] and may be thought of as a more advanced variant of cross-validation, utilizing an approach more complicated than crude cross-validation's winner-take-all strategy for merging individual learners. It is inspired from the concept of ensemble learning, which is based on the combinations of learners. The first-level stacking learner consists of numerous basics learners selected for training the same datasets, and the projected outputs create a new dataset to be transported into the second-level stacking learner [91]. When the first-level learner is the training model, cross-validation may be used to avoid over-fitting. In this study, we apply the k-fold cross-validation approach [70]. Figure 5.1 depicts the process of stacking techniques.

For our specific use we implement a stacking model that uses logistic regression, decision tree, random forest, and AdaBoost as base learners, each of these algorithms would be trained on the data independently and would make predictions based on their own models. These predictions would then be combined by a second-level learner, which in this case is logistic regression, to make a final prediction.

The choice of base learners in a stacking model is often guided by the specific characteristics of the data and the desired performance of the final model. Logistic Regression is a simple and fast linear model that is often used for binary classification tasks, making it a popular choice as a base learner. Decision Trees are also a popular choice for classification tasks and are relatively easy to understand and interpret. Random Forest is an ensemble method that can handle complex relationships between features and is generally more robust and accurate than a single decision tree. AdaBoost is another ensemble method that is known for achieving good performance on a wide range of tasks and is often used as a base learner in stacking models.

Overall, the use of logistic regression, decision tree, random forest, and AdaBoost as base learners in a stacking model allows for the combination of the predictions of multiple different models, potentially resulting in a more accurate and robust final model.

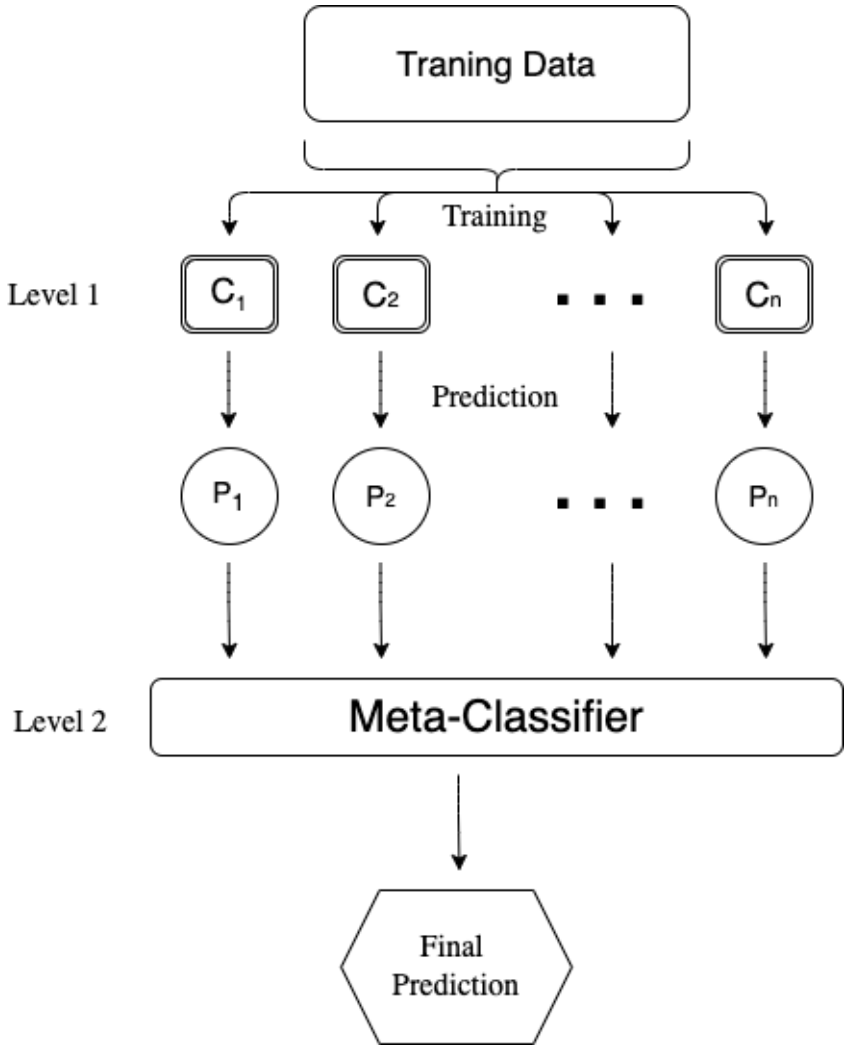


Figure 5.1: Stacked methods scheme

The use of logistic regression as the second-level learner allows for the combination of the predictions of the base learners in a simple and interpretable way.

5.3 Experiment and Analysis

5.3.1 Feature Engineering

Feature engineering or feature extraction is the process of leveraging domain expertise to extract features (characteristics, properties, attributes) from raw data. To improve the dataset's properties, we engineered new features. Because IP was the major feature for categorization, it was combined with one or two other attributes. This resulted in a dataset with more characteristics.

The features available in the dataset can be seen in Table 2.2. The "attributed_time" column was dropped. "Click time" was separated into separate columns, i.e., day and hour. Additional columns were created based on the repetition mean and variance of unique IPs in combination with other features from the given data such as device, App, OS and channel. We can see the engineered features in Table 5.1.

Features	Format	Description
ipCountPerHour	int16	Count of similar IPs with the same hour
countIpApp	int16	Count of the unique combination of ip and app
countIpAppOs	int16	Count of the unique combination of ip and app and os
tCountIpChannel	int16	Count of the of grouping by : ip, day, chanel and hour
varIpAppOs	float64	Variance of the combination of IP. App and OS
varPerDayIAC	float64	Variance of the unique value of the combination of IP, APP and Channel during a specific day
hMeanIAC	float64	mean of the combination of IP, App and Channel over an hour

Table 5.1: Engineered features description

The Boruta algorithm is utilized to select features for our dataset, and the results are shown in Figure 5.2. Blue boxplots correspond to minimal, average and maximum Z score of a shadow attribute, as previously explained, these shadow attributes are generated by adding randomness to the given data set by creating shuffled copies of all features. shadow features importance should be as we can observe quite minimal compared to the rest of the authentic features. The Boruta algorithm outputs one of three outcomes for each feature: accepted (shown in green in the boxplot), inconclusive (shown in yellow). or rejected (shown in red). As it is evident from the figure all features are marked as important. However, the IP feature is the most important of the original features. The features importance is shown in Table 5.2.

	meanImp	medianImp	minImp	maxImp	normHits	decision
ip	102.26642	103.21766	95.54441	108.70774	1	Confirmed
app	92.12307	91.93251	87.57589	98.38676	1	Confirmed
device	41.49026	40.80802	39.51531	45.89329	1	Confirmed
os	41.28527	41.04331	40.07366	42.67472	1	Confirmed
channel	77.84140	77.32284	73.61123	82.29046	1	Confirmed
hour	102.31205	102.95311	93.04829	113.15919	1	Confirmed
day	40.82301	40.93466	37.20855	44.67079	1	Confirmed
ipCountPerHour	112.41861	113.38703	107.82960	116.13228	1	Confirmed
countIpApp	101.03790	100.33621	96.48974	105.43832	1	Confirmed
countIpAppOs	77.17519	77.21642	69.28536	82.59379	1	Confirmed
tCountIpChannel	116.82685	118.12416	109.90573	124.10413	1	Confirmed
varIpAppOs	72.04046	72.68897	68.64300	77.22527	1	Confirmed
varPerDayIAC	93.10270	94.06280	86.11182	96.81464	1	Confirmed
hMeanIAC	128.17877	129.60581	119.26713	134.30723	1	Confirmed

Table 5.2: Features important

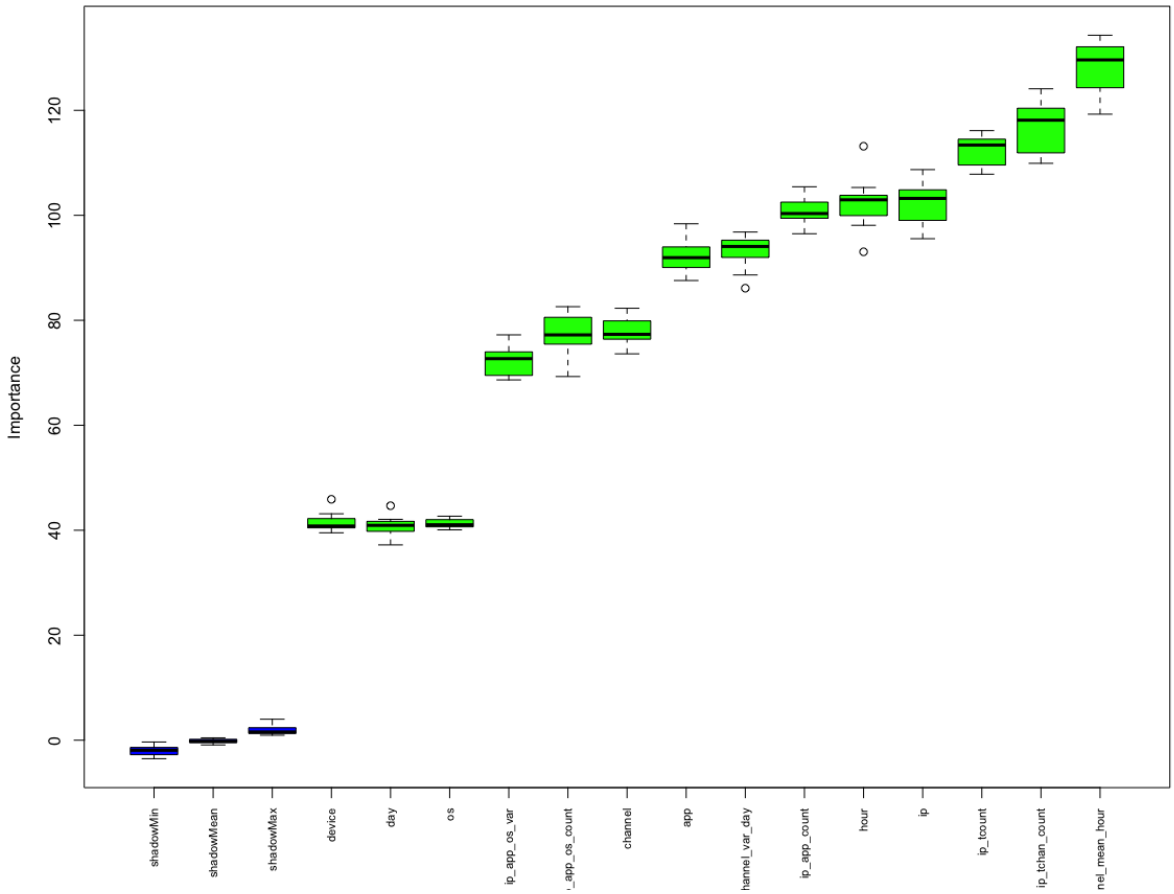


Figure 5.2: Feature selection outcomes

5.3.2 Experiment Results

5.3.2.1 Algorithm Comparison

There is no "one size fits all" algorithm in Machine Learning, so it is necessary to try multiple algorithms. In this section, we compare the Stacking algorithm using different metrics, first with the base learners and then with previously established algorithms.

In this research, we selected four algorithms: Logistic Regression, Decision Tree, Random Forest, and AdaBoost, in addition to the Stacking algorithm used, to train the same dataset, respectively. The experiment results are shown in Table 5.3. other than Stacking, Random Forest also showed great prediction results, as its evaluation indexes were the closest to the Stacking algorithm. Logistic Regression and Decision Tree models had the lowest AUC and Accuracy. On the other hand AdaBoost performed poorly, comparatively, in terms of *LogLoss* and had the second-lowest F1 Score.

The *ROC curve* can measure an algorithm's ability to generalize. The *AUC* is the area under the curve. The closer the *AUC* is to 1, the better the algorithm will be. We plotted the *ROC* ve for each algorithm and calculated the *AUC* Score, and the results are shown in Figure 5.3. We find that stacking reaches 0.972. The outcome of Random Forest is the closest to that of Stacking. This suggests that Random Forest contributes more to Stacking than the other algorithms.

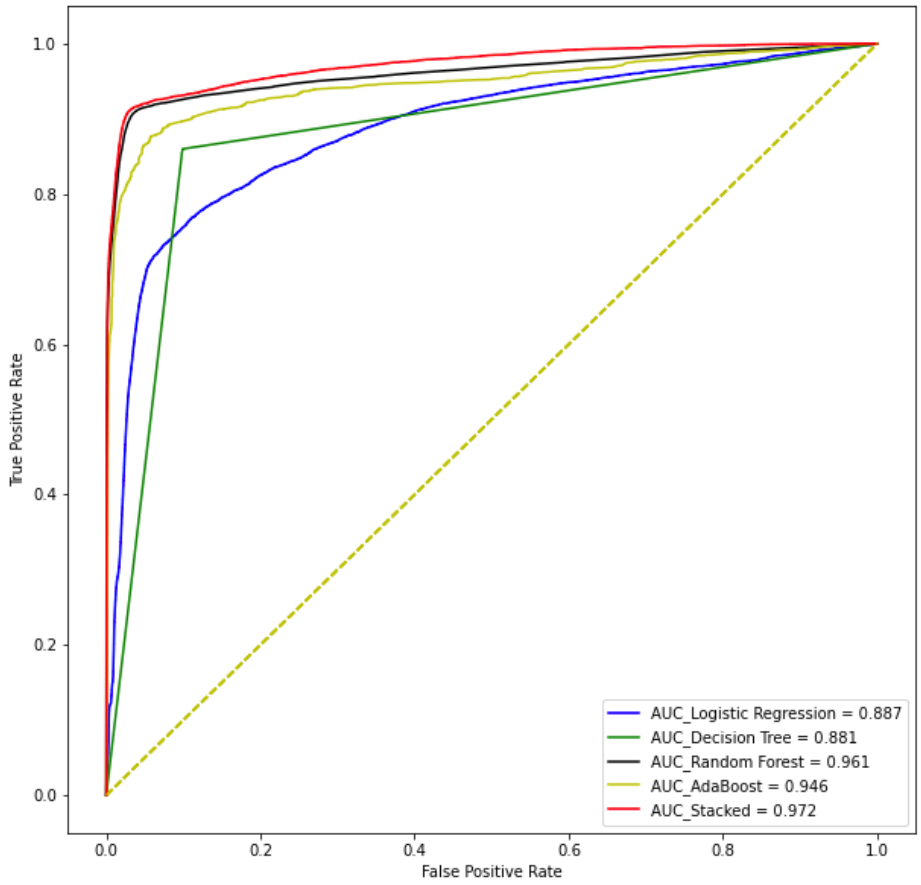


Figure 5.3: Receiver operating characteristic curve

	AUC	*Accuracy*	*F1 score*	*LogLoss*
Logistic Regression	0.887	0.922	0.031	0.273
Decision Tree	0.881	0.902	0.211	0.460
Random Forest	0.961	0.969	0.534	0.144
AdaBoost	0.946	0.930	0.128	0.670
Stacked Model	**0.972**	**0.975**	**0.552**	**0.075**

Table 5.3: The prediction results of different algorithms

Table 5.4 presents a metrics comparison to some of the previous related work. The metrics used *AUC* and the *F1score* given that they're the ones most available.

From Table 5.3, it is clear that our model compared to the rest of the base learners performs better in all aspects. While just from a 1st look to Table 5.4, it is clear that the proposed work does not perform as well as the CFXGB model in terms of *AUC*.

	SVM [65]	ETCF [65]	CFXGB [85]	**Stacked Model**
AUC	0.932	0.960	0.989	**0.972**

Table 5.4: AUC Comparison To Previous Work

This difference in results is mainly due to the preprocessing step. In this work we carefully ensure that our test set emulates a real environment. While in the previous work there was data leakage as they preprocessed the data and generated new variables based on the entire dataset, and then performed the train/test split.The distribution of real clicks and fraudulent clicks in the test set did not match the distribution in the original data. Also the test data rows are randomly selected from their upsampled/downsampled data instead of being the last rows of the original data.

5.3.2.2 Base Level Learners Analysis

Logistic Regression, Decision Tree, Random Forest, and AdaBoost are all classic machine learning algorithms that are used as base learners in our stacking ensemble. These algorithms are chosen for this particular work, because they have a proven track record of achieving good performance on a wide range of tasks.

From Table 5.3, it is clear that Random Forest outperforms Logistic Regression and Decision Tree. In the final part of our experiment, we removed one algorithm at a time to see how strong or weak algorithms alter Stacking prediction outcomes. The findings are provided in Table 5.5 where LR, DT RF and AB refers respectively to Linear Regression, Decision Tree, Random Forest and AdaBoost. Overall, there is no statistically significant difference between the different groups of first-level learners. The four assessment indices are comparable throughout the different situations. It is possible to infer that the Stacking algorithm not only ensures forecast accuracy but also maintains high stability. Although Random Forest performs well, the model still performs well when Random Forest is removed from the first-level learner. We can conclude from this part that the Stacking algorithm might be a wonderful approach for dealing with method selection, especially for large and complicated datasets like click fraud datasets.

	AUC	*Accuracy*	*F1 score*	*LogLoss*
LR, DT, RF	0.953	0.922	0.491	0.121
LR, AB, RF	0.962	0.961	0.505	0.092
AB, DT, RF	0.959	0.957	0.428	0.134
LR, DT, AB	0.940	0.930	0.322	0.397
Stacked Model	**0.972**	**0.975**	**0.552**	**0.075**

Table 5.5: The prediction results of base level learner combinations

5.4 Discussion

In this discussion section, we will delve deeper into the results of our study on predicting click fraud using a stacking algorithm. We will begin by explaining our approach to feature selection and how the Boruta algorithm was utilized to identify the most important features for the model. We will then discuss the performance of the proposed stacking model compared to the individual base learners and previous work in the field.

5.4.1 The practicality of the Boruta algorithm

One important aspect of our study was feature selection, as the success of a machine learning model can often depend on the quality and relevance of the input features. In order to identify the most useful features for our click fraud prediction model, we used the Boruta algorithm, which is a feature selection method that operates by iteratively adding and removing features from the model and comparing the performance with and without each feature.

Given that we had many engineered features in our dataset, the use of the Boruta algorithm was particularly useful in helping us to identify the features that were most important for the model. However, it's worth noting that in our case, the Boruta algorithm did not remove any features from the model. This suggests that all of the features in our dataset were relevant and contributed to the model's performance.

Overall, the use of the Boruta algorithm was beneficial in helping us to select the most useful features for our click fraud prediction model. This, in turn, helped to improve the performance of the model and allowed us to achieve more accurate predictions.

5.4.2 Evaluation of Stacking algorithm against the base learners

In this study, we evaluated the performance of five machine learning algorithms for predicting click fraud: Logistic Regression, Decision Tree, Random Forest, AdaBoost, and a Stacking model, which combines the predictions of the individual algorithms.

The results showed that all of the algorithms had high accuracy scores, ranging from 0.902 for Decision Tree to 0.975 for the Stacking model. However, the F1 scores were much lower, with the exception of the Random Forest and Stacking models, which had F1 scores of 0.534 and 0.552, respectively. This suggests that the algorithms may be better at accurately classifying the majority class (non-fraudulent clicks) than the minority class (fraudulent clicks). Of all the algorithms, the Stacking model appears to be the most

adept at detecting fraudulent clicks, as demonstrated by its higher F1 score compared to the other algorithms.

In an extremely unbalanced dataset like ours, the AUC and log loss are often more indicative of the model's performance than the accuracy score. The AUC measures the model's ability to distinguish between the positive and negative classes, with a score of 1 indicating perfect discrimination and a score of 0.5 indicating no discrimination. The log loss measures the model's confidence in its predictions, with a lower score indicating higher confidence.

The results showed that the Stacking model had the highest AUC score of 0.972, followed by AdaBoost with a score of 0.946, Random Forest with a score of 0.961, Logistic Regression with a score of 0.887, and Decision Tree with a score of 0.881. In terms of log loss, the Stacking model also had the lowest score of 0.075, followed by Random Forest with a score of 0.144, AdaBoost with a score of 0.67, Logistic Regression with a score of 0.273, and Decision Tree with a score of 0.460.

Overall, the Stacking model outperformed the individual algorithms in terms of both AUC and log loss. One possible reason for this is that the Stacking model is able to combine the strengths of the individual algorithms and make more accurate predictions as a result. Additionally, the use of the second-level learner in the Stacking model may have helped to improve the model's performance by weighting the predictions of the base learners differently.

It's worth noting that the Stacking model does come with an extra computational cost compared to the individual algorithms. However, the improvement in performance demonstrated by the Stacking model suggests that this extra cost might be worth it, especially in cases where accurate prediction is particularly important, such as in the detection of fraudulent clicks.

5.4.3 Evaluation of Stacking algorithm against previous work

In comparing our results to previous work, it's worth noting that the AUC scores for the SVM, ETCF, and CFXGB models were 0.932, 0.960, and 0.989, respectively. the score for CFXGB is slightly higher than the AUC score of 0.972 that we obtained for the Stacking model. However, it's important to consider that previous studies may have suffered from data leakage, which can artificially inflate the performance of the model. The claim of data leakage can be verified by examining the data processing and description steps and the train/test splits in the study.

In order to ensure that our results were as reliable as possible, we took extra care to eliminate data leakage in the test environment for the Stacking model. Data leakage occurs when information from the test set leaks into the training set, leading to overly optimistic results. By carefully controlling the test environment and eliminating data leakage, we were able to obtain more accurate and realistic results.

Overall, our results suggest that the Stacking model is a promising approach for predicting click fraud, particularly in an extremely unbalanced dataset. While the AUC score for the Stacking model was slightly lower than the one obtained in [85], the extra care we took to eliminate data leakage means that our results are likely to be more reliable and reflective of the model's true performance in a real-world setting.

5.4.4 Limitations

One limitation of this research is the limited availability of real-life data for testing fraud detection algorithms. While the dataset used in this study is a large and publicly available dataset, it may not fully reflect the complexities and nuances of detecting fraud in a real-world setting. Using click-through rate (CTR) datasets for fraud detection may be a viable option due to similarities between the two types of datasets. However, it is important to carefully consider the differences and similarities between CTR and fraud datasets in order to accurately apply and interpret the results.

Another limitation to consider is the additional computational resources required for the stacking algorithm compared to some of the base learners. While the improved performance of the stacking algorithm may justify the additional computational cost, it is important to weigh the trade-off between performance and efficiency in practical applications.

Overall, these limitations should be taken into account when interpreting and applying the results of this study. Further research and testing with more diverse and realistic datasets may provide valuable insights and improve the robustness of the proposed method.

5.5 Conclusion

We have presented a click fraud prediction classification approach based on the Stacking algorithm in this study. To handle unbalanced datasets, under-sampling and the SMOTE method are introduced, and the Boruta algorithm is used to pick input features. TalkingData dataset is used for a comparative study. The following are the key contributions:

1. Feature engineering and preprocessing are essential steps for maximizing the usefulness of available data. Therefore, we engineered nine additional features from the existing data that play a major role in the prediction algorithm, as indicated by the feature selection results.

2. To select features, we used the Boruta algorithm. Feature selection is critical when using machine learning technology. As shown in Section 4.1, the comparative experimental findings are consistent with the feature selection results obtained using the Boruta algorithm, demonstrating its effectiveness. Using the Random Forest classifier, we obtained the importance values of 14 features, and conducted experiments by inputting different combinations of these features into the model in order of their importance values. We verified all features in the dataset and found that many of the engineered features were among the most important.

3. This chapter proposes a click fraud prediction classification approach based on Stacking. We used Random Forest, Logistic Regression, Decision Tree, and AdaBoost as base learners, and Logistic Regression as the second-level learner. To assess the individual contributions of the four first-level learners, we trained the same dataset using each learner individually. The results suggest that Stacking outperforms all the other learners, with Random Forest being the closest in terms of performance.

In future research, it would be interesting to explore the use of other machine learning technologies for click fraud prediction. Additionally, it may be useful to consider

the impact of laws such as GDPR on the detection of fraudulent clicks. Another potential area of study is optimizing data curation to balance the positive and negative categories. An alternative approach could be to focus on improving the quality of the data rather than solely improving the performance of the predictors.

CHAPTER 6

Conclusions & Perspectives

This dissertation aimed to address the challenge of extremely unbalanced data within the field of advertising. Recognizing the substantial impact of data imbalance on the quality and reliability of insights and predictions, we undertook a comprehensive investigation in this area. By focusing on advertising, a domain characterized by an exceptionally extreme data imbalance, we sought to shed light on the implications of this issue and propose effective solutions.

6.1 Contribution

6.1.1 Analyzing Data Imbalance in CTR Prediction for Advertising: Insights, Feature Significance, and Sampling Technique Evaluation

In the first part of our research, we delved into the impact of data imbalance on CTR prediction within the field of advertising. We systematically analyzed the effects of varying degrees of data imbalance on prediction accuracy and extracted valuable insights regarding the behavior and performance of models under different imbalance ratios. By examining the relationship between data imbalance and prediction outcomes, we gained a deeper understanding of the challenges and limitations imposed by skewed data distributions.

Additionally, we investigated the significance of different features in the prediction process. Through a comprehensive analysis, we identified key features that played a crucial role in accurately predicting user clicks. This insight provided valuable guidance for advertisers in optimizing their advertising strategies and allocating resources effectively.

Building upon these findings, we conducted a comparative study to determine the most suitable combination of sampling techniques and machine learning algorithms for CTR prediction. By systematically evaluating various sampling techniques, such as Random Undersampling (RUS), Random Oversampling (ROS), and Synthetic Minority Over-sampling Technique (SMOTE), in conjunction with popular machine learning algorithms, we aimed to identify the most effective approach for handling data imbalance in the context of CTR prediction.

After rigorous experimentation and performance evaluation, our results revealed

that the combination of RUS and Random Forest yielded the best results in terms of prediction accuracy. This finding highlights the importance of considering the interplay between sampling techniques and machine learning algorithms when dealing with highly imbalanced advertising datasets. It also provides valuable guidance for researchers and practitioners seeking to develop robust CTR prediction models that can effectively handle extreme data imbalance.

6.1.2 Enhancing Click Fraud Detection in Online Advertising: Optimized Models, Realistic Application, and Stacking Algorithm Approach

In the second part of our research, we turned our attention to the critical issue of click fraud detection within the advertising domain. Click fraud represents a significant challenge, as it undermines the integrity of online advertising campaigns and can result in financial losses for advertisers. We aimed to develop effective methods for identifying fraudulent clicks and enhancing the overall transparency and trustworthiness of online advertising.

To achieve this, we initially proposed an optimized XGBoost model with hyperparameter tuning through grid search. XGBoost is a powerful machine learning algorithm known for its effectiveness in handling complex and imbalanced datasets. By fine-tuning the hyperparameters, we aimed to maximize the performance of the model in accurately identifying fraudulent clicks. Through rigorous experimentation and evaluation, we demonstrated the improved fraud detection capabilities of our optimized XGBoost model.

Furthermore, we recognized the need for realistic application scenarios to evaluate the practical effectiveness of our proposed methods. To address this, we developed a growing and pruning approach, which allowed us to simulate the real-world deployment of our fraud detection system. By gradually introducing new data and continuously pruning the model, we were able to assess its performance in a dynamic and evolving environment. This realistic application approach provided valuable insights into the adaptability and robustness of our click fraud detection system.

Finally, we presented a novel and highly effective model built on the stacking algorithm. Stacking combines the strengths of multiple machine learning models by training a meta-learner to make predictions based on the outputs of individual base models. By leveraging this approach, we were able to achieve even higher accuracy in detecting click fraud. The stacking model demonstrated superior performance by effectively integrating the predictive power of multiple algorithms and further enhancing the reliability of our fraud detection system.

6.1.3 Ensuring Practicality and Reproducibility: Robust Data Treatment and Methodology for Real-World Applications

Throughout this dissertation, we placed significant emphasis on the practicality and reproducibility of our work, ensuring that our findings can benefit both academia and the industry. We recognized the importance of meticulous data treatment and adhered to rigorous standards to maintain data integrity and prevent data leakage.

First and foremost, we implemented thorough data preprocessing techniques to ensure that the data was properly cleaned, normalized, and prepared for analysis. This step involved addressing missing values, handling outliers, and resolving any inconsistencies in the dataset. By following established best practices in data preprocessing, we aimed to maintain the integrity and reliability of our results.

Furthermore, we prioritized the prevention of data leakage, which can inadvertently introduce biases and compromise the accuracy of our findings. To mitigate this risk, we strictly adhered to the principle of separating training and testing datasets, ensuring that information from the testing set did not influence the training process or model development. By maintaining a clear delineation between these datasets, we could provide a fair and unbiased evaluation of the performance of our models.

In addition to data treatment, we adopted standardized methods of handling the data, including feature engineering, model selection, and evaluation metrics. By following established practices and widely accepted methodologies, we aimed to enhance the reproducibility and comparability of our results. Researchers and practitioners can confidently replicate our experiments and evaluate our proposed methods using consistent benchmarks and evaluation criteria.

Moreover, we made a concerted effort to document our research process comprehensively. This documentation included clear explanations of our methodology, data preprocessing steps, model development, and hyperparameter tuning. By providing detailed descriptions and code examples, we aimed to facilitate the reproducibility of our work and enable others to build upon our findings.

By prioritizing the practicability and reproducibility of our work, we aimed to bridge the gap between academic research and real-world applications. Our rigorous treatment of the data, adherence to established standards, and comprehensive documentation contribute to the robustness and reliability of our findings, allowing for wider adoption and practical implementation.

6.2 Perspectives

In this section, we discuss the perspectives and potential future work that can build upon the findings and contributions of our research. By exploring new avenues and addressing emerging challenges, researchers and practitioners can further advance the field of CTR prediction for advertising and click fraud detection. We highlight key areas such as ethical considerations and fairness, long-term performance monitoring, interpretability and explainability of models, as well as exploring novel data sources and features. These perspectives provide a roadmap for future investigations, enabling the development of more robust and responsible solutions that cater to the evolving needs of the advertising industry and ensure a trustworthy and transparent digital ecosystem.

Ethical Considerations and Fairness:

Ethical considerations and fairness in advertising and fraud detection algorithms are crucial aspects that deserve careful attention and further exploration in future research. As these algorithms play a significant role in shaping user experiences and can have a profound impact on individuals and communities, it is essential to ensure that they operate in a fair and unbiased manner:

1. Identifying Potential Biases: Future work could involve investigating potential biases present in advertising and fraud detection algorithms. Biases can arise from various sources, including the data used for training the models, the features selected, or the assumptions made during model development. It is important to examine whether these biases disproportionately affect certain user groups, leading to unfair or discriminatory outcomes.

2. Fair Treatment of User Groups: Researchers can explore techniques and methodologies to ensure fair treatment of different user groups while maintaining high accuracy in prediction and fraud detection. This could involve developing algorithms that explicitly consider fairness metrics and constraints during the model training and optimization process. For example, employing fairness-aware learning approaches, such as adversarial learning or equalized odds, to reduce disparate impact or treatment across different demographic or socioeconomic groups.

3. Ethical Data Collection and Usage: Another critical aspect is ensuring that the data used for training and evaluating advertising and fraud detection algorithms is obtained and utilized ethically. This includes obtaining informed consent, protecting user privacy, and being transparent about data collection and usage practices. Future research can explore methods to minimize data biases and develop frameworks that prioritize ethical considerations when collecting and processing user data.

4. Mitigating Unintended Consequences: Researchers should investigate potential unintended consequences that may arise from advertising and fraud detection algorithms. For example, these algorithms could inadvertently reinforce stereotypes or create filter bubbles that limit users' exposure to diverse information. Future work could focus on developing techniques to mitigate these unintended consequences and promote fairness, diversity, and user autonomy in algorithmic decision-making.

5. User-Centric Design and Feedback: Future research can emphasize a user-centric approach to the design and deployment of advertising and fraud detection systems. Engaging users and obtaining their feedback can help identify and address potential ethical concerns and biases in the algorithms. Incorporating user perspectives into the development process can contribute to fairer and more inclusive systems that align with user values and preferences.

6. Regulatory and Policy Considerations: Ethical considerations and fairness in advertising and fraud detection algorithms also extend to the regulatory and policy domain. Future work can involve collaborations with policymakers and industry stakeholders to develop guidelines and standards that ensure ethical practices in algorithmic advertising and fraud detection. This could include the development of auditing mechanisms, transparency requirements, and guidelines for addressing biases and promoting fairness.

By delving into ethical considerations and fairness aspects, future research can contribute to the development of more responsible and accountable advertising and fraud detection algorithms. This will not only improve user experiences but also promote trust and transparency in the digital advertising ecosystem.

Long-Term Performance Monitoring:

Long-term performance monitoring is a critical aspect to ensure the practicality and effectiveness of the proposed methods in real-world applications. While initial evaluations and experiments provide valuable insights, it is essential to assess the models' performance over an extended period to understand their stability, adaptability, and continued effectiveness:

1. Data Collection and Retraining: Long-term performance monitoring involves regularly collecting new data and updating the models accordingly. This ensures that the models are trained on the most recent and relevant information, capturing the evolving dynamics of data imbalance and click fraud patterns. By periodically retraining the models with the updated data, researchers can assess their ability to adapt to changing conditions and maintain accurate predictions over time.

2. Evaluating Model Performance: The collected data can be used to evaluate the long-term performance of the models. Researchers can analyze various performance metrics, such as accuracy, precision, recall, and F1-score, to assess the models' effectiveness in handling data imbalance and detecting click fraud. By monitoring these metrics over an extended period, insights can be gained into the models' consistency, robustness, and their ability to maintain high-quality predictions.

3. Identifying Performance Decay: Long-term monitoring helps researchers identify potential performance decay or degradation over time. It allows them to detect any shifts in the data distribution, emerging fraud patterns, or evolving user behaviors that might impact the models' accuracy. By detecting performance decay, researchers can investigate the underlying causes and develop strategies to mitigate its effects, such as updating the models' algorithms or incorporating new features to capture emerging patterns.

4. Adaptive Model Updates: Long-term monitoring can guide researchers in making adaptive updates to the models. If performance degradation is detected, it may be necessary to revise the models' architecture, re-evaluate feature significance, or incorporate new techniques to handle emerging challenges. By continuously monitoring performance, researchers can iteratively improve the models, ensuring their relevance and effectiveness in combating data imbalance and click fraud over an extended period.

5. Benchmarking against Baselines: Long-term performance monitoring also facilitates benchmarking the models against established baselines or competing approaches. By comparing the models' performance with other state-of-the-art methods, researchers can gain insights into their relative strengths and weaknesses. This benchmarking process can help identify areas for further improvement and guide future research directions.

6. Real-World Validation and User Feedback: In addition to quantitative evaluation, long-term monitoring provides an opportunity for real-world validation and user feedback. Researchers can collaborate with industry partners and deploy the models in production environments to assess their performance in practical settings.

Engaging with advertisers, ad platforms, and end-users can provide valuable insights and feedback, further enhancing the models' effectiveness and applicability in real-world scenarios.

By including long-term performance monitoring as part of future research efforts, the proposed methods can be thoroughly evaluated for their stability, adaptability, and long-term effectiveness. This iterative process of data collection, evaluation, and model updates ensures that the models remain relevant and capable of addressing the challenges posed by data imbalance and click fraud detection over extended periods.

More Potential Perspectives

1. **Addressing Emerging Imbalance Challenges:** While the research conducted provides valuable insights into the impact of data imbalance on CTR prediction and evaluates various sampling techniques, it is essential to recognize that the field of advertising is dynamic and constantly evolving. Future work could focus on identifying emerging challenges and addressing them. For example, exploring the impact of evolving user behavior or changes in ad platforms on data imbalance and developing innovative techniques to handle these challenges.

2. **Advanced Feature Selection Methods:** The research identifies key features that significantly contribute to accurate CTR prediction. However, future work could explore advanced feature selection methods, such as genetic algorithms or deep learning-based techniques, to identify the most informative and relevant features more efficiently. This could enhance prediction accuracy and provide advertisers with more precise insights for optimizing their advertising strategies.

3. **Ensemble Techniques for Fraud Detection:** The research presents a stacking algorithm as an effective approach for click fraud detection. Future work could explore other ensemble techniques, such as boosting or bagging, to further improve the performance of fraud detection models. Investigating the combination of different machine learning algorithms and ensembling methods could lead to even higher accuracy in identifying fraudulent clicks.

4. **Real-Time Fraud Detection:** The research acknowledges the need for realistic application scenarios but focuses on a growing and pruning approach. Future work could extend the evaluation to real-time fraud detection, where the model is continuously updated and validated using streaming data. This would allow for the detection of new and evolving fraud patterns, providing advertisers with real-time protection against click fraud.

5. **Generalizability to Other Domains:** The research primarily focuses on advertising and click fraud detection. However, the insights and methodologies developed could potentially be generalized to other domains facing similar challenges of data imbalance and fraud detection. Future work could explore the applicability of the proposed approaches in domains such as cybersecurity, finance, or healthcare, where imbalanced data and fraud detection are significant concerns.

By considering these perspectives and pursuing potential future work, researchers can further advance the field of CTR prediction, click fraud detection, and data imbalance

analysis, leading to more effective and reliable solutions in the advertising industry and beyond.

List of Publications

We list all publications that arise from this Thesis organized by chronological order:

1. Sahllal, N., & Souidi, E. M. (2023). A Comparative Analysis of Sampling Techniques for Click-Through Rate Prediction in Native Advertising. In IEEE Access (Vol. 11, pp. 24511–24526). Institute of Electrical and Electronics Engineers (IEEE). https://doi.org/10.1109/access.2023.3255983.

2. N. Sahllal and E. Souidi, "Forecasting Click Fraud via Machine Learning Algorithms," in *4th International Conference C2SI-2023 on Codes, Cryptology and Information Security*,Vol. 13874, 2023, Springer.

3. N. Sahllal and E. Souidi, "Click fraud prediction by stacking algorithm," *Intelligenza Artificiale*, 2023

And these are papers we published over the course of our PhD

1. Belabssir, S., & Sahlal, N. (2021). Tensor Product And Linear Error Block Codes. IAENG International Journal of Applied Mathematics, 51(2).

2. Belabssir, S., Sahllal, N., & Souidi, El. M. (2019). Cyclic linear error-block codes. In AIP Conference Proceedings. 2ND INTERNATIONAL CONFERENCE ON APPLIED MATHEMATICS, ICAM'2018. Author(s). https://doi.org/10.1063/1.5090622

Bibliography

[1] https://www.readpeak.com. *readpeak*.

[2] Digital advertising soared 35internet advertising revenue report, May 2022.

[3] Shaza M Abd Elrahman and Ajith Abraham. Class imbalance problem using a hybrid ensemble approach. *International journal of hybrid intelligent systems*, 12(4):219–227, 2015.

[4] Zahra Putri Agusta et al. Modified balanced random forest for improving imbalanced data prediction. *International Journal of Advances in Intelligent Informatics*, 5(1):58–65, 2019.

[5] Rahbar Ahsan, Wei Shi, and Jean-Pierre Corriveau. Network intrusion detection using machine learning approaches: Addressing data imbalance. *IET Cyber-Physical Systems: Theory & Applications*, 7(1):30–39, 2022.

[6] Qasem Al-Tashi, Helmi Md Rais, Said Jadid Abdulkadir, Seyedali Mirjalili, and Hitham Alhussian. A review of grey wolf optimizer-based feature selection methods for classification. *Evolutionary Machine Learning Techniques: Algorithms and Applications*, pages 273–286, 2020.

[7] Paulo S Almeida and João JC Gondim. Click fraud detection and prevention system for ad networks. *Journal of Information Security and Cryptography (Enigma)*, 5(1):27–39, 2018.

[8] Sergio A Alvarez. An exact analytical relation among recall, precision, and classification accuracy in information retrieval. *Boston College, Boston, Technical Report BCCS-02-01*, pages 1–22, 2002.

[9] Mahesh Bathula, Rama Chaithanya Tanguturi, and Srinivasa Rao Madala. Click fraud detection approaches to analyze the ad clicks performed by malicious code. In *Journal of Physics: Conference Series*, volume 2089, page 012077. IOP Publishing, 2021.

[10] Gustavo EAPA Batista, Ronaldo C Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29, 2004.

[11] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[12] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks*, 106:249–259, 2018.

[13] BusinessInsider. Report: Ad fraud is 'second only to the drugs trade' as a source of income for organized crime, 6 2016.

[14] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[15] Nitesh V Chawla. Data mining for imbalanced datasets: An overview. *Data mining and knowledge discovery handbook*, pages 875–886, 2010.

[16] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[17] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, Tianyi Zhou, et al. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4, 2015.

[18] Damien Dablain, Bartosz Krawczyk, and Nitesh V Chawla. Deepsmote: Fusing deep learning and smote for imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[19] Anshuman Dash and Satyajit Pal. Auto-detection of click-frauds using machine learning. *Int. J. Eng. Sci. Comput.*, 10:27227–27235, 2020.

[20] Xuhua Ding. A hybrid method to detect deflation fraud in cost-per-action online advertising. In *Applied Cryptography and Network Security: 8th International Conference, ACNS 2010, Beijing, China, June 22-25, 2010. Proceedings 8*, pages 545–562. Springer, 2010.

[21] Charles Elkan. The foundations of cost-sensitive learning. In *International joint conference on artificial intelligence*, volume 17, pages 973–978. Lawrence Erlbaum Associates Ltd, 2001.

[22] Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C Prati, Bartosz Krawczyk, and Francisco Herrera. *Learning from imbalanced data sets*, volume 10. Springer, 2018.

[23] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[24] Alejandro G. Martín, Alberto Fernández-Isabel, Isaac Martín de Diego, and Marta Beltrán. A survey for user behavior analysis based on machine learning techniques: current models and applications. *Applied Intelligence*, 51(8):6029–6055, 2021.

[25] Kun Gai, Xiaoqiang Zhu, Han Li, Kai Liu, and Zhe Wang. Learning piece-wise linear models from large scale data for ad click prediction. *arXiv preprint arXiv:1704.05194*, 2017.

[26] Mikel Galar, Alberto Fernandez, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4):463–484, 2011.

[27] Mona Gandhi, Markus Jakobsson, and Jacob Ratkiewicz. Badvertisements: Stealthy click-fraud with unwitting accessories. *Journal of Digital Forensic Practice*, 1(2):131–142, 2006.

[28] Vicente García, José Salvador Sánchez, and Ramón A Mollineda. Exploring the performance of resampling strategies for the class imbalance problem. In *Trends in Applied Intelligent Systems: 23rd International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2010, Cordoba, Spain, June 1-4, 2010, Proceedings, Part I 23*, pages 541–549. Springer, 2010.

[29] Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. Learning from class-imbalanced data: Review of methods and applications. *Expert systems with applications*, 73:220–239, 2017.

[30] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *Advances in Intelligent Computing: International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August 23-26, 2005, Proceedings, Part I 1*, pages 878–887. Springer, 2005.

[31] Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328, 2008.

[32] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

[33] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.

[34] https://imbalanced learn.org. imbalanced-learn documentation!

[35] Jin Huang and Charles X Ling. Using auc and accuracy in evaluating learning algorithms. *IEEE Transactions on knowledge and Data Engineering*, 17(3):299–310, 2005.

[36] Wattana Jindaluang, Varin Chouvatut, and Sanpawat Kantabutra. Under-sampling by algorithm with performance guaranteed for class-imbalance problem. In *2014 International Computer Science and Engineering Conference (ICSEC)*, pages 215–221. IEEE, 2014.

[37] Justin M Johnson and Taghi M Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):1–54, 2019.

[38] Yuchin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. Field-aware factorization machines for ctr prediction. In *Proceedings of the 10th ACM conference on recommender systems*, pages 43–50, 2016.

[39] Fumihiro Kanei, Daiki Chiba, Kunio Hato, and Mitsuaki Akiyama. Precise and robust detection of advertising fraud. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, volume 1, pages 776–785. IEEE, 2019.

[40] Fumihiro Kanei, Daiki Chiba, Kunio Hato, Katsunari Yoshioka, Tsutomu Matsumoto, and Mitsuaki Akiyama. Detecting and understanding online advertising fraud in the wild. *IEICE TRANSACTIONS on Information and Systems*, 103(7):1512–1523, 2020.

[41] Mehmed Kantardzic, Chamila Walgampaya, and Wael Emara. Click fraud prevention in pay-per-click model: Learning through multi-model evidence fusion. In *2010 International Conference on Machine and Web Intelligence*, pages 20–27. IEEE, 2010.

[42] Kyungwon Kim, Eun Kwon, and Jaram Park. Deep user segment interest network modeling for click-through rate prediction of online advertising. *IEEE Access*, 9:9812–9821, 2021.

[43] Brendan Kitts, Jing Ying Zhang, Gang Wu, Wesley Brandi, Julien Beasley, Kieran Morrill, John Ettedgui, Sid Siddhartha, Hong Yuan, Feng Gao, et al. Click fraud detection: adversarial pattern recognition over 5 years at microsoft. *Real World Data Mining Applications*, pages 181–201, 2015.

[44] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[45] Miroslav Kubat, Stan Matwin, et al. Addressing the curse of imbalanced training sets: one-sided selection. In *Icml*, volume 97, page 179. Citeseer, 1997.

[46] Miron B Kursa and Witold R Rudnicki. Feature selection with the boruta package. *Journal of statistical software*, 36:1–13, 2010.

[47] Jorma Laurikkala. Improving identification of difficult small classes by balancing class distribution. In *Artificial Intelligence in Medicine: 8th Conference on Artificial Intelligence in Medicine in Europe, AIME 2001 Cascais, Portugal, July 1–4, 2001, Proceedings 8*, pages 63–66. Springer, 2001.

[48] Paul H Lee. Resampling methods improve the predictive power of modeling in class-imbalanced datasets. *International journal of environmental research and public health*, 11(9):9776–9789, 2014.

[49] Dongfang Li, Baotian Hu, Qingcai Chen, Xiao Wang, Quanchang Qi, Liubin Wang, and Haishan Liu. Attentive capsule network for click-through rate and conversion rate prediction in online advertising. *Knowledge-Based Systems*, 211:106522, 2021.

[50] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[51] Dengpan Liu and Vijay Mookerjee. Advertising competition on the internet: Operational and strategic considerations. *Production and Operations Management*, 27(5):884–901, 2018.

[52] Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):539–550, 2008.

[53] Rushi Longadge and Snehalata Dongre. Class imbalance problem in data mining review. *arXiv preprint arXiv:1305.1707*, 2013.

[54] Manuel Lopez-Martin, Belen Carro, and Antonio Sanchez-Esguevillas. Variational data generative model for intrusion detection. *Knowledge and Information Systems*, 60:569–590, 2019.

[55] Abdul Majid, Safdar Ali, Mubashar Iqbal, and Nabeela Kausar. Prediction of human breast and colon cancers from imbalanced data using nearest neighbor and support vector machines. *Computer methods and programs in biomedicine*, 113(3):792–808, 2014.

[56] Inderjeet Mani and I Zhang. knn approach to unbalanced data distributions: a case study involving information extraction. In *Proceedings of workshop on learning from imbalanced datasets*, volume 126, pages 1–7. ICML, 2003.

[57] H Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1222–1230, 2013.

[58] Ahmed Metwally, Divyakant Agrawal, Amr El Abbad, and Qi Zheng. On hit inflation techniques and detection in streams of web advertising networks. In *27th International Conference on Distributed Computing Systems (ICDCS'07)*, pages 52–52. IEEE, 2007.

[59] AlAli Moneera, AlQahtani Maram, AlJuried Azizah, Taghareed AlOnizan, Dalia Alboqaytah, Nida Aslam, and Irfan Ullah Khan. Click through rate effectiveness prediction on mobile ads using extreme gradient boosting. *Comput. Mater. Continua*, 66(2):1681–1696, 2021.

[60] Shishir Nagaraja and Ryan Shah. Clicktok: click fraud detection using traffic analysis. In *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, pages 105–116, 2019.

[61] Nurulfitrah Noorhalim, Aida Ali, and Siti Mariyam Shamsuddin. Handling imbalanced ratio for class imbalance problem using smote. In *Proceedings of the Third International Conference on Computing, Mathematics and Statistics (iCMS2017) Transcending Boundaries, Embracing Multidisciplinary Diversities*, pages 19–30. Springer, 2019.

[62] Richard Oentaryo, Ee-Peng Lim, Michael Finegold, David Lo, Feida Zhu, Clifton Phua, Eng-Yeow Cheu, Ghim-Eng Yap, Kelvin Sim, Minh Nhut Nguyen, et al. Detecting click fraud in online advertising: a data mining approach. *The Journal of Machine Learning Research*, 15(1):99–140, 2014.

[63] Zahra Pooranian, Mauro Conti, Hamed Haddadi, and Rahim Tafazolli. Online advertising security: Issues, taxonomy, and future directions. *IEEE Communications Surveys & Tutorials*, 23(4):2494–2524, 2021.

[64] Joseph Prusa, Taghi M Khoshgoftaar, David J Dittman, and Amri Napolitano. Using random undersampling to alleviate class imbalance on tweet sentiment data. In *2015 IEEE international conference on information reuse and integration*, pages 197–202. IEEE, 2015.

[65] Xiaokang Qiu, Yuan Zuo, and Guannan Liu. Etcf: An ensemble model for ctr prediction. In *2018 15th International Conference on Service Systems and Service Management (ICSSSM)*, pages 1–5. IEEE, 2018.

[66] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1:81–106, 1986.

[67] Ismael Ramos-Pérez, Álvar Arnaiz-González, Juan J Rodríguez, and César García-Osorio. When is resampling beneficial for feature selection with imbalanced wide data? *Expert Systems with Applications*, 188:116015, 2022.

[68] Juniper Reasearch. Ad fraud to cost advertisers 19 billion in 2018, representing 9total digital advertising spend.

[69] Steffen Rendle. Factorization machines. In *2010 IEEE International conference on data mining*, pages 995–1000. IEEE, 2010.

[70] Juan D Rodriguez, Aritz Perez, and Jose A Lozano. Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE transactions on pattern analysis and machine intelligence*, 32(3):569–575, 2009.

[71] Nadir Sahllal and El Mamoun Souidi. Click fraud prediction by stacking algorithm. *Intelligenza Artificiale*, 2023.

[72] Nadir Sahllal and El Mamoun Souidi. A comparative analysis of sampling techniques for click-through rate prediction in native advertising. *IEEE Access*, 11:24511–24526, 2023.

[73] Nadir Sahllal and El Mamoun Souidi. Forecasting click fraud via machine learning algorithms. In *4th International Conference C2SI-2023 on Codes, Cryptology and Information Security*. Springer, 2023.

[74] JVV Sriram Sasank, G Ram Sahith, K Abhinav, and Meena Belwal. Credit card fraud detection using various classification and sampling techniques: a comparative study. In *2019 International Conference on Communication and Electronics Systems (ICCES)*, pages 1713–1718. IEEE, 2019.

[75] Nadine Schröder, Andreas Falke, Harald Hruschka, and Thomas Reutterer. Analyzing the browsing basket: a latent interests-based segmentation tool. *Journal of Interactive Marketing*, 47(1):181–197, 2019.

[76] Chris Seiffert, Taghi M Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. Rusboost: Improving classification performance when training data is skewed. In *2008 19th international conference on pattern recognition*, pages 1–4. IEEE, 2008.

[77] Aliaksei Severyn and Alessandro Moschitti. Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 959–962, 2015.

[78] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[79] Michael R Smith, Tony Martinez, and Christophe Giraud-Carrier. An instance level analysis of data complexity. *Machine learning*, 95:225–256, 2014.

[80] Angela A Sodemann, Matthew P Ross, and Brett J Borghetti. A review of anomaly detection in automated surveillance. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):1257–1272, 2012.

[81] Mervyn Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical society: Series B (Methodological)*, 36(2):111–133, 1974.

[82] Ahmad S Tarawneh, Ahmad B Hassanat, Ghada Awad Altarawneh, and Abdullah Almuhaimeed. Stop oversampling for class imbalance learning: A review. *IEEE Access*, 10:47643–47660, 2022.

[83] Fadi Thabtah, Suhel Hammoud, Firuz Kamalov, and Amanda Gonsalves. Data imbalance in classification: Experimental evaluation. *Information Sciences*, 513:429–441, 2020.

[84] G. S. Thejas, Kianoosh G. Boroojeni, Kshitij Chandna, Isha Bhatia, S. S. Iyengar, and N. R. Sunitha. Deep learning-based model to fight against ad click fraud. In *Proceedings of the 2019 ACM Southeast Conference*, ACM SE '19, page 176–181, New York, NY, USA, 2019. Association for Computing Machinery.

[85] GS Thejas, Surya Dheeshjith, SS Iyengar, NR Sunitha, and Prajwal Badrinath. A hybrid and effective learning approach for click fraud detection. *Machine Learning with Applications*, 3:100016, 2021.

[86] GS Thejas, Jayesh Soni, Kianoosh G Boroojeni, SS Iyengar, Kanishk Srivastava, Prajwal Badrinath, NR Sunitha, Nagarajan Prabakar, and Himanshu Upadhyay. A multi-time-scale time series analysis for click fraud forecasting using binary labeled imbalanced dataset. In *2019 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, volume 4, pages 1–8. IEEE, 2019.

[87] Avraam Tsantekidis, Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Forecasting stock prices from the limit order book using convolutional neural networks. In *2017 IEEE 19th conference on business informatics (CBI)*, volume 1, pages 7–12. IEEE, 2017.

[88] Olga Ussatova, Aidana Zhumabekova, Yenlik Begimbayeva, Eric T Matson, and Nikita Ussatov. Comprehensive ddos attack classification using machine learning algorithms. *CMC-COMPUTERS MATERIALS & CONTINUA*, 73(1):577–594, 2022.

[89] Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478*, 2020.

[90] Dennis L Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, (3):408–421, 1972.

[91] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[92] Ying Xie, Dan Jiang, Xinmei Wang, and Rongbin Xu. Robust transfer integrated locally kernel embedding for click-through rate prediction. *Information Sciences*, 491:190–203, 2019.

[93] Haitao Xu, Daiping Liu, Aaron Koehl, Haining Wang, and Angelos Stavrou. Click fraud detection on the advertiser side. In *Computer Security-ESORICS 2014: 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7-11, 2014. Proceedings, Part II 19*, pages 419–438. Springer, 2014.

[94] Show-Jane Yen and Yue-Shi Lee. Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset. In *Intelligent Control and Automation: International Conference on Intelligent Computing, ICIC 2006 Kunming, China, August 16–19, 2006*, pages 731–740. Springer, 2006.

[95] M. Yuen. Native advertising industry 2022: Forecast, trends, and the rise of video, July 2022.

[96] Harry Zhang. The optimality of naive bayes. *Aa*, 1(2):3, 2004.

[97] Zhuoyuan Zheng, Yunpeng Cai, and Ye Li. Oversampling method for imbalanced classification. *Computing and Informatics*, 34(5):1017–1037, 2015.

[98] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)*, pages 207–212, 2016.

ROYAUME DU MAROC
Université Mohammed V
- RABAT -
جامعة محمد الخامس
– الرباط –
Faculté des sciences
كلية العلوم

Abstract

This research dissertation tackles the significant challenges of unbalanced datasets in advertising, focusing on Click-Through Rate (CTR) prediction and click fraud detection. It explores the impact of data imbalance on CTR prediction, revealing how it affects predictive accuracy and identifying key features crucial for precision. The study also delves into advanced algorithm development for optimized click fraud detection, employing machine learning techniques like XGBoost and ensemble methods. Emphasizing practicality and reproducibility, the research incorporates thorough data treatment and robust methodologies suitable for real-world applications. This comprehensive approach not only enhances the understanding of data imbalance in advertising but also provides practical solutions for effective CTR prediction and click fraud detection. The dissertation concludes with future perspectives, advocating for ongoing research in ethical considerations, long-term performance, advanced feature selection, and real-time fraud detection, aiming to broaden the applicability of these findings to other domains.

Keywords : Unbalanced data, Click-through rate (CTR), Click fraud, Advertising, Machine learning.

Resume

Cette thèse de recherche aborde les défis importants posés par les ensembles de données déséquilibrés dans le domaine de la publicité, en se concentrant sur la prédiction du taux de clics (CTR) et la détection de la fraude au clic. Elle explore l'impact du déséquilibre des données sur la prédiction du CTR, révélant comment cela affecte la précision prédictive et identifiant les caractéristiques clés essentielles pour la précision. L'étude se penche également sur le développement d'algorithmes avancés pour une détection optimisée de la fraude au clic, en employant des techniques d'apprentissage automatique comme XGBoost et des méthodes d'ensemble. Mettant l'accent sur la praticité et la reproductibilité, la recherche intègre un traitement approfondi des données et des méthodologies robustes adaptées aux applications du monde réel. Cette approche complète améliore non seulement la compréhension du déséquilibre des données dans la publicité, mais fournit également des solutions pratiques pour une prédiction efficace du CTR et la détection de la fraude au clic. La thèse se conclut avec des perspectives d'avenir, plaidant pour des recherches continues sur les considérations éthiques, le suivi des performances à long terme, la sélection avancée des caractéristiques et la détection de la fraude en temps réel, dans le but d'élargir l'applicabilité de ces résultats à d'autres domaines.

Mots-clés : Données déséquilibrées, Taux de clics (CTR), Fraude au clic, Publicité, Apprentissage automatique.