Prediction and Anomaly Detection for Enterprise Access Control

by

Andre D'Souza

April 2019

Supervised by Dr. Yuri Lawryshyn

ESC499 Engineering Science Thesis

Abstract

The objective of this thesis was to classify enterprise access control mapping for anomalies, as well as explore the prediction of future user resource permission requirements. To identify anomalies, a dataset containing access control mapping was analyzed using a combination of outlier classification models. Outlier ratings were created for each entry in the dataset by aggregating the models, and it was concluded that 5% of the resources mapped to users require review for necessity. Prediction models were also applied to the dataset, and it was determined recommendation systems are a viable method to predict resource request approvals. The significance of this work is to maximize enterprise cybersecurity, while reducing employee downtime and access control administration costs.

Acknowledgements

I would like to thank the project sponsor for the opportunity to work under their guidance, while solving a high impact cybersecurity problem. I would also like to thank Dr. Yuri Lawryshyn for his supervision and support throughout this thesis. Lastly, I would like to thank my family and friends for their encouragement.

Table of Contents

List of Tables and Figures

Chapter 1: Introduction

Background

The scale and complexity of large organizations causes the administration of network security to be time consuming, error prone and expensive [1]. The National Institute of Standards and Technology identified Role Based Access Control (RBAC) as the preferred network access control solution in organizations over 500 members [1]. RBAC designates various network roles to a predefined permission requirement list, and maps users to various roles based on their access needs. When organizations possess dynamic user bases with changing employee responsibilities, RBAC requires excess role creation to correctly grant permission access, causing the problem defined as role explosion [1].

Gap

Role Based Access Control is easier maintained when fewer generalized roles are defined, although network security is enhanced by minimizing the permissions granted to each user. Role explosion is mitigated by linking multiple roles to single users, leading to permission set covering issues [1]. The omission of necessary permissions creates barriers to user functionality, while granting excess permissions during user-role mapping creates security risks. An excess set of permissions increases data loss vulnerability if a user incorrectly alters systems outside their circle of competence, or has their system compromised through malware or theft.

Objective

The objective of this research is to detect anomalies in permissions granted to users for existing access control configurations, and apply recommendation systems to predict future access control mapping and maximize enterprise network security. To accomplish this goal, nonessential permissions in user resource access control mapping were identified, and a resource approval predictor for granting new permissions to users was applied.

Key Deliverables

Outlier classification methods were applied to access control data to detect a subset of anomalies. The effectiveness of recommender systems including collaborative filtering and tree

methods were also explored to predict employee resource mapping [2]. The key deliverables presented to the project sponsor, a Canadian financial institution (Bank), include a proposal statement and interim presentation. The final deliverable will be an extensive report on access control prediction and anomaly detection, as well as the implemented Python code and a presentation on methodology and results.

Significance

An improved access control mapping technique will better administer minimal permission granting to users, improving cybersecurity within large organizations. In the cases of employee error or system compromise, enterprise systems will be less vulnerable to data loss compared to the implementation of traditional Role Based Access Control. Predicting user resource permissions will also reduce access control administration costs and employee downtime.

Thesis Organization

Chapter 2 will discuss the literature review conducted, beginning with the identification of a gap in Role Based Access Control. Recommendation systems will be reviewed as a possible prediction technique, and methods for anomaly detection will be identified. Next, Chapter 3 will state the methodology performed for data preparation, anomaly detection, and access control mapping prediction. The results from each stage of the methodology will be presented and briefly discussed in Chapter 4. Finally, Chapter 5 will identify conclusions and recommendations based on the results in Chapter 4, as well as discuss extensions to the methodology in Chapter 3.

Chapter 2: Literature Review

The purpose of this literature review is to justify the need for access control anomaly detection and the use of recommendation systems to improve upon Role Based Access Control (RBAC) as an enterprise access control technique. Both the reasons for widespread use and limitations of RBAC will be identified. Following the recognition of a gap in enterprise access control, both the development and use of active recommendation systems will be explored and the applications to enterprise access control will be proposed. Finally, a Python toolkit for anomaly detection will be identified.

2.1 Enterprise Access Control

Enterprise Access Control is a critical aspect of organizational cybersecurity. To maximize network security, each employee should have access to precisely the resources essential to their work [3]. While the absence of permissions can prevent employees from performing their role, the over granting of access is not feasible for several reasons. If a workstation is compromised, infected users possessing minimal access reduces the damage potential from hackers and the data theft that may occur. Furthermore, excess access permissions may allow employees to modify data outside their circle of competence and create operational problems and risks for an enterprise [1]. Excess access can also create data privacy issues, and provide employees lacking security clearance access to confidential information involving other employees, customers, or the enterprise [3].

2.2 Role Based Access Control

Access Control Lists map permissions to each employee and can grant minimal access control when customized by network administrators, however, as organizations grow this method becomes a costly and intensive task [3]. Role Based Access Control has emerged as the preferred network access solution in organizations over 500 members due to administration cost reduction and efficient provisioning reducing employee downtime [1]. The U.S. Department of Commerce estimated that RBAC saved U.S. enterprises $1.8 billion in 2009 alone [1]. Rather than map permissions directly to users as in the case of Access Control Lists, permissions are mapped to predetermined roles created in the system [1]. These roles are then mapped to users, creating a tiered approach involving role-permission mapping, and user-role mapping [4].

Users can be mapped to multiple roles to expand access granted, and ease future access allocation for network administrators compared to specific permission granting. Furthermore, new employees joining the organization can have access easily created by replicating the RBAC profiles of employees performing similar job roles. [3] While RBAC is an efficient cost saving method to administer access in a generalized manner, it rarely provides optimal user-permission mapping in the manner Access Control Lists can [5]. In a dynamic organization with a growing employee base and evolving job role responsibilities, RBAC can create security vulnerabilities.

Specifically, the drawbacks that come with RBAC are mainly encompassed at the role level. As an organization expands and new employees require a variety of access, role explosion may occur when roles are overly customized [4]. If roles are specific to individual employees, the explosion of role quantities can transform RBAC into Access Control Lists and require immense effort from network administrators [4]. To counter role explosion, users are mapped to multiple roles, giving administrators the ability to easily provide users robust permissions through user-role mapping [1]. The overlapping of roles is also prone to set covering issues caused by under-extension or over-extension.

Set covering issues arise when there are insufficient roles available to map permissions to users [4]. Under-extension occurs when role-permission mapping is not inclusive of all permission sets [4]. Network administrators must manually grant access in a time intensive process, while employees lose productivity. To prevent under-extension, role-permission mapping may over-extend and encompass resource permissions outside the scope of the role's activities [1]. Over-extension creates the security vulnerabilities discussed in Section 2.1.

2.3 Attribute Based Access Control

While most design development of Role Based Access Control systems occurred over a decade ago [1], the clear upgrade has emerged as Attribute Based Access Control (ABAC) [5]. Attribute based access control still possesses user-role and role-permission mapping, however, it also includes additional attributes to profile access needs. These attributes could be based on users such as demographics or clearance level [5], or stem from permissions such as the enterprise ownership group of the resource a permission provides.

Context has also emerged as a significant attribute including request location, date and time, or network information such as IP address. The request context allows Attribute Based Access Control to operate as a more dynamic policy based method, as compared to the static permission granting in Role Based Access Control [5]. While there is a clear advantage to ABAC over RBAC, the Role Based version is the significantly more common method currently in use since upgrade costs outweigh the network security benefits from an operational outlook [1].

2.4 Recommendation Systems

Role Based Access Control has limited room to expand due to its structured approach requiring strict user-role and role-permission mapping. On the other hand, recommendation systems (engines) have undergone continuous development with the rise of the digital world, and have influential applications to e-commerce, digital media, search engines and more [6]. Recommendation systems are primarily used to rank items for suggestion to users. These items can take a variety of forms, including e-commerce retail products in the case of Amazon, digital media distributed by Netflix, or web results provided by search engines such as Google [7]. Much of the development and innovation within recommendation systems is driven by corporations looking to improve customer experience and boost sales [7]. The goal of recommendation systems is to use past data to identify patterns and provide improved item rankings to both new and existing users [7]. In an access control context, recommendation systems can by utilized to automate and enhance employee access control. Metrics are required to compare recommendation systems against other access control methods.

2.5 Success Measures

An access control recommendation system requires criteria for comparison to RBAC and other access control methods. Success measures for recommendation systems are often grouped into three categories: accuracy, user satisfaction and provider satisfaction [6]. Accuracy is the primary metric and can be defined depending on the product type and product owner preference. Digital recommendations can be quantified using implicit measures including click through rate or conversion rate, with conversion leading to a sale or adoption of the item [6]. The commonly used metrics to evaluate recommendation algorithms include Mean Squared Error (MSE) and Root Mean Squared Error (RSME) [8].

Accuracy also impacts user satisfaction, and while user experience is dependent on providing appropriate suggestions, the top recommendation systems also maximize serendipity [6]. User satisfaction is driven by the level of surprise created, influenced by whether the recommendation was obvious or unlikely to be found without the use of the recommendation system. While relevant, users can become displeased through obvious recommendations undermining user experience. Finally, the vital criteria for recommendation system success is provider satisfaction [6], which is largely driven by accuracy, user satisfaction, and whether the recommendation system increases sales and profit.

In the context of access control recommendation systems, success criteria can be transformed and simplified. Accuracy can be determined by validating whether permissions granted are within the scope of an employee's functional role. A dynamic determinant would involve using network usage logs to ensure the resources a user has been granted are actively used by that employee in accordance with expected frequencies. User satisfaction would be purely driven by whether employees have sufficient access to the resources they require for their functional role, and have downtime due to access control reduced. Provider satisfaction is driven by reducing the cost and time required for network administrators to manually customize access control at a single user level, as well as ensuring employees have minimal access in order to maximize network security.

2.6 Recommendation System Considerations

Most recommendation systems are subject to a generic set of limitations [9]. The major setback is the cold start problem for new recommendation engine users. When a new user first enters an organization, data is unavailable for that particular user's attributes. Insufficient data prevents bucketing with other users' patterns to identify relevant suggestions, and handicaps recommendation systems until new users have produced enough data to display a pattern [9]. A common solution to minimize the cold start problem is to survey users on their preferences before their introduction to the system [7]. However, these surveys are often neglected, causing data sparsity in addition to the cold start problem [9]. In the context of access control, employees could be surveyed on their access needs, or have attributes predefined by their managers as opposed to the actual users.

Another drawback of recommendation systems is the difficulty to scale across all users. As the user base expands, the large user datasets can become very computationally expensive, and impede recommendation engine performance [9]. Recommendation systems also suffer from incomplete user attributes causing data sparsity [10]. While segments of the user data may contain in depth organizational attributes, if a large portion of the user base has attributes undefined, the sparse matrices increase computational complexity for recommendations. Finally, recommendation systems should also aim to maximize recommendation diversity and serendipity, to creates more value for users than repetitive suggestions [7]. In access control, serendipity may involve forecasting resources and permissions needed for future projects before employees need to raise resource requests to network administrators.

2.7 Collaborative Filtering

The most implemented types of recommendation systems are based on collaborative filtering [11]. These systems operate by predicting future user decisions based on the past decisions made by many similar users. The basis is if two people agree on one decision, they are more likely to agree on another [11]. The larger the decision and attribute overlap between users, the more likely item adoption decisions unique to each person will be shared by the other. The grouping of similar users is the most direct link between RBAC role mapping and recommendation systems.

Employees with related positions and responsibilities should have similar RBAC profiles, so collaborative filtering would bucket permission sets for these employees. The major benefit to collaborative filtering is the dynamic nature. If a group of similar employees begins displaying a split pattern, groupings could be dynamically modified as opposed to a network administrator manually configuring new user-role mapping. However, collaborative filtering is especially susceptible to the cold start problem for new users [7].

2.8 Hybrid Recommender Systems

While user based collaborative filtering can achieve strong performance, issues often arise when the user base grows large enough to make the algorithms very computationally expensive [7]. Amazon developed an item-item filtering approach to aggregate ratings per product instead of user, adding the benefit of data stability [10]. When users greatly exceed the

number of products, dynamic and expanding user bases do not destabilize the consistent item matrices therefore easing computation significantly [10]. For enterprise access control, it is unlikely to require a solely item-item approach due to the significantly smaller user bases compared to most recommender systems. With the transition from Access Control Lists to Role Based Access Control usually occurring before 500 users [1], even the largest organizations span much smaller employee bases than the customer bases for large recommendation systems [8].

While collaborative filtering is the more common method, an alternative type of recommendation system uses a content based approach. The content based systems contain attributes for each item or product, and then match items to user profiles [12]. Rather than aggregating users together, similar products are aggregated and then user-item mapping connects users to product buckets [12]. This mapping makes the content based approach highly similar to Attribute Based Access Control, showcasing the application of recommendation systems to be used as an enterprise access control solution.

Adding complexity to recommendation systems can increase the scalability problem as algorithms become very computationally expensive [11]. However, a hybrid approach can combine several variations of recommendation systems to yield better results [13]. A first layer of models can function for feature extraction and manipulation, before the latter models cluster users and aggregate results [13]. With smaller user bases for access control applications, hybrid approaches can be less prone to scalability issues and be used to achieve higher accuracy. The benefits obtained from combining recommendation engines were demonstrated in the Netflix Prize.

2.9 Netflix Prize

The Netflix Prize that ran from 2006 to 2009 remains one of the most famous periods in Recommendation System development [8]. The competition created by Netflix offered a $1 Million USD prize for the most accurate predictor of user movie ratings, with a minimum 10% RMSE reduction over Netflix's current algorithm at the time [8]. Even a 1% improvement has a large effect on the top recommendations for users [14]. The competition showcased the value of ensemble learning, with the top finishers all implementing variations of ensemble algorithms that combine several different machine learning models [8]. The winner was a joint effort ensembling

preliminary submissions created by three separate teams, and used gradient boosted decision trees as the aggregation technique [13].

2.10 Amazon Employee Access Challenge

The search for access control datasets led to the discovery of the publically available Kaggle "Amazon.com - Employee Access Challenge" held in 2013 [15]. The goal of this competition was to create a predictor of employee resource requests as approved or denied, to minimize manual access control transactions. The anonymized dataset provided contains labelled (approved or denied) access control requests collected over a two-year basis. The submissions required classification algorithms determining whether new resource requests should be approved or denied for a particular employee [15]. The winning methodology includes code published under a MIT License, and utilized tree models, gradient boosted machines, and random forests [16]. The winning solution also required extensive feature engineering to convert the categorical text strings to numerical data that models could be trained on.

2.11 PyOD Toolkit

PyOD is the only open-source Python toolkit dedicated to outlier detection, and was recently released in 2019 [17]. It was explored as a possible method for classifying anomalies in present access control mapping. The advantages PyOD offers over alternative Python outlier detection methods include access to over 20 outlier detection algorithms, as well as ensemble methods that aggregate outputs of multiple algorithms [17]. Furthermore, the available algorithms span a variety of outlier detection classification categories, including linear models, proximity and ensembling [17]. The PyOD toolkit also contains clear documentation, extensively tested models, and access to sample code for applications [17].

2.12 Chapter Summary

Role Based Access Control is the dominant access control method in organizations over 500 members, however, permission set covering issues create a gap from an operational and security standpoint [1]. Cost savings would be obtained by reducing employee downtime due to access omissions, as well as automating and enhancing access control processes. Eliminating unnecessary permissions granted improves cybersecurity by reducing data loss vulnerabilities.

Recommendation systems are actively used to connect users to items based on patterns in user activity, attributes and item interaction [7]. The recommendation system approach may be used to predict employee access control mapping as an improved method over Role Based Access Control. The implementation of a hybrid recommender system through an ensemble algorithm is proven to have the most accurate output [13], and has the potential to become an enterprise access control solution. PyOD toolkit for Python may also be applied to detect anomalies in access control mapping. The implementation of methods explored throughout Chapter 2 will be explained in the following chapter.

Chapter 3: Methodology

The processes used in this thesis can be grouped into three stages: data preparation, prediction of permission mapping, and outlier identification. The latter two encompass the main goals, which were to predict future employee resource access requirements and classify current access control mapping for anomalies, defined as permissions granted to employees which they do not require. To accomplish both goals, a dataset with the project sponsor's existing access control mapping (Bank dataset) was structured to have similar composition to data from Kaggle's Amazon Access Challenge (Amazon dataset) in order to build on submissions.

While the influence of the Amazon dataset on this project will be discussed, all methodology conducted and results obtained pertain to the Bank dataset provided by the Canadian financial institution sponsoring this project. This chapter will first cover the data cleaning and feature engineering that was conducted on the Bank dataset. Following data preparation, the methods for anomaly detection will be addressed, followed by the process to predict future access control mapping. A high level overview of the total methodology is shown in Figure 3.1, with each step explained in subsequent sections.

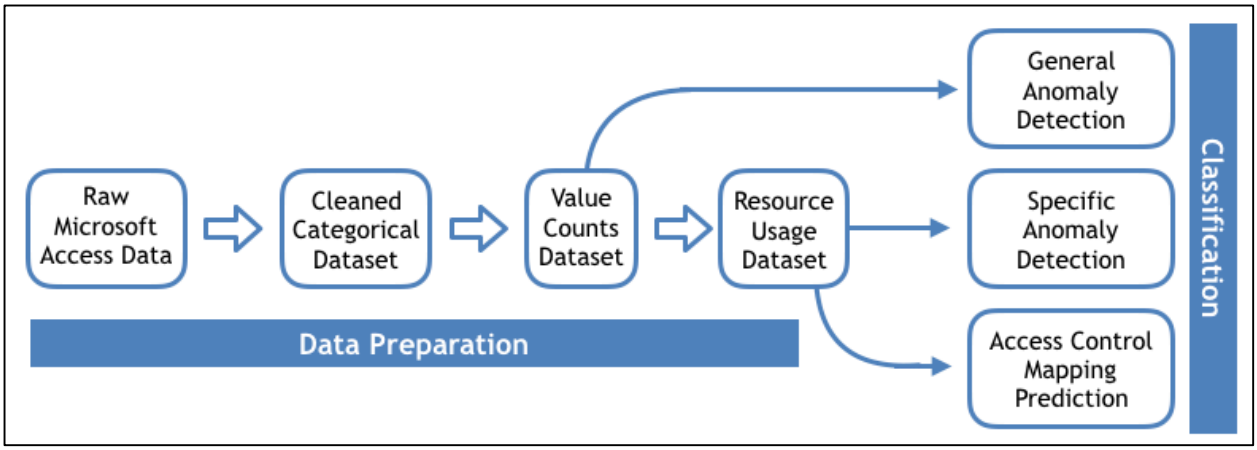

Figure 3.1 Methodology Overview

3.1 Amazon Access Challenge Data

The dataset provided by Kaggle's Amazon Access Challenge discussed in Section 2.10 will be explained. The structure was important to consider before transforming the project sponsor's access control data, in order to build on publically available solutions to Kaggle's

challenge. The Amazon dataset columns can be viewed in Table 1, along with a short description for each feature and the presence in both the Amazon and Bank datasets. It should be clarified that "resource" refers to a digital system as opposed to an employee. Each row represents an employee's access to a particular resource, therefore, an Action of 1 indicates a resource permission was granted to that employee. The remaining features are attributes containing information that describes an employee's role in the enterprise.

Table 3.1 Overview of the Amazon [15] and Bank Employee Access Datasets

Feature / Column	**Feature Description**	**Amazon Dataset**	**Bank Dataset**
Action	ACTION is 1 if the resource was granted, 0 if denied	Yes	No
Resource	An ID for each resource	Yes	Yes
Manager	The ID of the manager of the current employee record	Yes	Yes
Role Rollup 1 / Business	Company role grouping category id 1 (e.g. US Retail Engineering) / Business Role	Yes	Yes
Role Rollup 2 / Bufugu	Company role grouping category id 2 (e.g. US Retail) / Company business unit functional group (Bufugu)	Yes	Yes
Department	Company role department description (e.g. Retail)	Yes	Yes
Title	Company role business title (e.g. Senior Retail Manager)	Yes	Yes
Role Code / Ccode	Company code (Ccode) unique to each role (e.g. Manager)	Yes	Yes
Role Family Description	Company role family extended description (e.g. Retail Manager, Software)	Yes	No
Role Family	Company role family description (e.g. Retail Manager)	Yes	No
Employee ID (DN)	An ID for each employee	No	Yes

The Amazon dataset included roughly 33k rows of employee resource requests, with 94% of the requests resulting in resource approvals indicated by an Action of 1. Furthermore, the majority of resources had just a single occurrence in the dataset. The Amazon dataset's structure was presented to the project sponsor, as a guideline for organizing the Bank's access control data for analysis.

3.2 Project Sponsor's Microsoft Access Data

Microsoft Access is an enterprise database management software used by the project sponsor (Bank), that contains employee access control information. Microsoft Access allows access control data for up to 1000 employees per batch to be exported into XML flat file format. Each section of the export received contained an employee ID, their respective company attributes (Role, Title, Department…), as well as a comprehensive list of all resource permissions granted to that employee. The XML export was then hashed to transform the data into anonymous alphanumeric strings to protect the privacy of employees. A sample of this XML output can be viewed in Appendix A. As shown in Table 1, the Bank dataset lacks an "Action" feature, as the project sponsor does not store resource requests that were denied. All rows in the dataset indicate resources that have been granted to the user. The Bank dataset also contains an employee ID for each resource permission granted.

A data parser based on text identifiers was coded using Python to convert the Bank XML flat file into a tabular format matching the structure of the Amazon Access Challenge data. Each unique employee and resource pairing was stored to a row also containing that employee's categorical company attributes. A small portion of employees lacked company attributes and were removed, leaving the final Bank dataset with 956 employees and 66.3k rows of access control information. The ratio of unique entries per column to total rows was also reviewed and compared against the uniqueness ratios of the Amazon dataset to confirm the datasets had similar composition. The final Bank dataset identifies the resource granted, as well as the seven other categorical features associated with that employee (see Table 3.1).

3.3 Feature Engineering

With the Bank dataset cleaned and transformed into tabular data structured similar to the Amazon dataset, publically available solutions to Kaggle's Amazon Access Challenge were further explored as a building block for other access control applications. The purpose of feature engineering was to convert the categorical alphanumeric text strings (Role Title, Department…) into numerical data that algorithms could be applied to. In particular, the methodology and code of Benjamin Solecki's Kaggle Challenge solution [16] made significant contributions to the feature engineering performed on the Bank dataset.

The feature extraction used by Solecki on the Amazon dataset included appending data frame columns containing value counts of each entry [18]. Each row now contained the categorical variables, as well as the number of occurrences that particular entry appeared in the dataset. For example, if the Role Title was "US Engineering", the count of "US Engineering" out of all Role Title entries was added to the row. The numerical counts were then used to generate resource usage percentages pertaining to each categorical feature in that resource's particular row, as depicted in Figure 3.1 above. Calculating resource usage percentages was the most computationally extensive stage in data preparation.

The usage percentage indicates for each company attribute (Title, Department, Business…), what percentage of that specific attribute's occurrences in the dataset were related to that particular resource out of all 66.3k rows in the Bank dataset. The categorical data was now converted to a normalized numerical dataset. The final Bank datasets for analyses were constructed as lists of all resource permission entries, with indices matching onto two 66.3k x 7 data frames. The first data frame contained the value counts generated and the other contained the resource usage percentages. The seven columns pertain to the categorical features: Manager, Department, Title, Business, Business Unit Functional Group, Company Role Code and Employee ID.

3.4 Anomaly Detection

The preliminary anomaly detection performed was exploratory data analysis on the value counts generated for each feature. A count of the number of resources each employee has been granted revealed likely outliers due to an overabundance of access provided. A similar procedure was performed on manager counts, to possibly identify managers that are overly lenient when granting their underlying employees access. However, this initial analysis is unable to identify which resources in these large baskets granted were the actual outliers, and which lie within the scope of that employee's access needs. The general anomaly detection is a list of likely employees possessing excess access.

The main stage of outlier classification aimed to use a more granular approach, and classify individual rows pertaining to a specific resource granted to an employee as an anomaly or ordinary. The resource usage counts data frame containing the seven normalized numerical

features was analyzed using PyOD. As discussed in Section 2.11, the PyOD toolkit for Python was selected due to its dedicated application for outlier detection, and comprehensive set of models available. Out of the available models, seven were implemented and tested against the dataset. The tested models included: Angle Based Outlier Detection, Cluster Based Local Outlier Factor, Feature Bagging, Histogram Based Outlier Detection, Isolation Forest, K Nearest Neighbours and Average K Nearest Neighbours. The combination of classifiers was chosen based on their ability to handle high data dimensionality, as well as a mixture of proximity and ensembling classification techniques.

A Python algorithm based on PyOD sample code [19] was created to combine and classify each unique pairing of the seven features in the Bank dataset, creating 21 possible combinations. The seven classifiers were used to detect outliers in each feature pairing. Using visualizations generated from each classification, it was clear that Isolation Forest, Histogram Based Outlier Detection, and Cluster Based Local Outlier Factor were the most applicable models. The remaining classifiers were unable to detect distinct outliers and deemed unsuitable for the Bank dataset. The visualizations generated from classifying the Bank dataset's feature pairing are presented in Chapter 4 as well as in Appendix B.

The final anomalies were determined using two methods, with the first method producing three sets of outlier classifications for each of the Bank dataset's entries in the full dimension resource usage dataset (all seven features). The three classifications applied were Isolation Forest, Histogram Based Outlier Detection, and Cluster Based Local Outlier. The second method involved creating an ensemble of the same three classifiers on feature pairs, and was possible since the models were not deemed computationally expensive. The three classifiers were applied to each of the 21 feature combinations used to generate outlier graphs, outputting a total of 63 classifications. The 63 outputs were ensembled using voting aggregation by summation, with the final output containing an Outlier Score ranging from 0 to 63 for each of the 66.3k rows. The methodology using Outlier Score for anomaly detection is depicted in Figure 3.2.

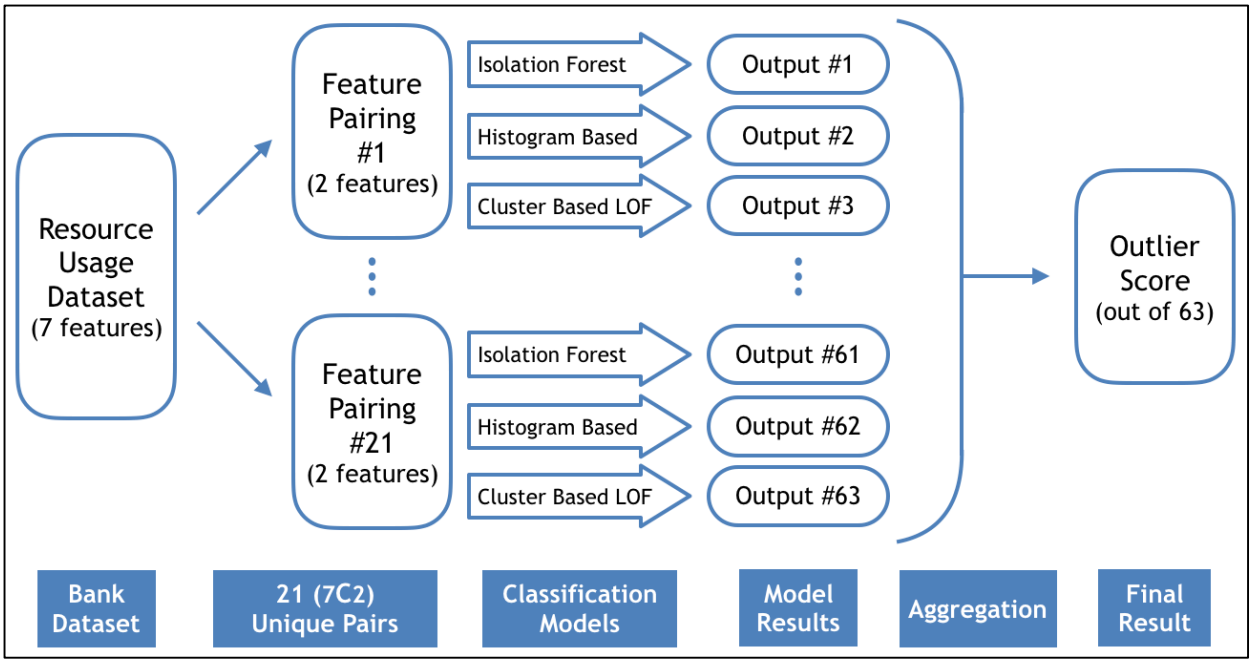

Figure 3.2 Methodology for Outlier Score Classification

3.5 Prediction of Employee Resource Mapping

The process to predict employee resource needs once again expanded on Benjamin Solecki's solution to Kaggle's Amazon Access Challenge. The Amazon dataset was a form of supervised learning, as resource requests were labeled by Action as approved or denied. However, the Bank dataset was exclusively composed of granted resources to employees. The project sponsor's Microsoft Access does not store denied resource requests, hence rows possessing an Action of 0 were synthesized.

As discussed in Section 3.1, 94% of requests in the Amazon dataset were approved. An action column was appended to the Bank dataset with an entry of 1 (approved) for all 66.3k rows. An additional 4k synthetic rows with an Action of 0 were then created using resource and employee pairing that were absent from the original Bank dataset, meaning the employee lacked access to the resource. The drawbacks of using synthetic rows will be discussed in Section 5.1. The total modified Bank dataset was now sized at 70.3k rows, with 94% resource approvals to mimic the Amazon dataset. While synthetic rows are an imperfect workaround, the Bank dataset was able to be run through Benjamin Solecki's ensemble.py predictor [18] which is a form of collaborative filtering. Minor modifications were made to adjust the predictor to the Bank

dataset. The ensemble.py predictor labels the target (Action) of each row with a probability of resource approval. The probabilities are determined using three classifiers: Random Forest, Extra trees and Gradient Boosting.

3.6 Chapter Summary

The Microsoft Access data provided by the project sponsor contained access control mapping, and was converted from a flat file into a tabular format matching the structure of the Amazon dataset. Both datasets included rows pertaining to individual employee resource permission mapping, as well as company attributes for each employee. The categorical text strings were transformed into numerical entries by generating value counts and normalized resource usage percentages for each unique entry. The transformed Bank dataset was then analyzed using PyOD toolkit for Python to classify each employee resource mapping as an anomaly or ordinary. The Bank dataset was also appended with synthetic resource denial rows, and run through collaborative filtering predictors created as solutions to Kaggle's Amazon Access Challenge.

Chapter 4: Results and Analysis

The results from Chapter 3's methodology will be presented and described in order to guide interpretation. The stages performed include data preparation, access control mapping anomaly classification, and future employee resource mapping prediction. It should be noted that outliers and anomalies identified in the Bank dataset do not necessarily confirm a gap in cybersecurity or access control standards. There is the possibility of model misclassification, and explanations including administrators and senior employees requiring elevated access levels. Due to the anonymized nature of the dataset, possible outliers can only be confirmed after decryption.

4.1 Bank Dataset Overview

Table 4.1 provides a snapshot of the Bank dataset. Firstly, the original cleansed Bank data is presented with counts of all values and unique values per feature. The second data frame refers to the value counts for each categorical entry relative to its column, after transforming text strings into numerical data. The resource usage percentages refer to the normalized data identifying the percentage of each categorical attribute's appearances associated with that particular resource. For example, out of the 684 unique managers indicated by data frame A, the most a manager appears is 660 out of 66,288 rows as shown by the value counts in data frame B. The most a resource is associated with a particular manager is 3.03% of that manager's appearances in the Bank dataset, as shown by data frame C.

Table 4.1 Summary of Bank Dataset & Transformed Versions

	Resource	Manager	Department	Title	Bufugu	Business	Ccode	DN
Data Frame A: Cleaned Bank Dataset								
Count	66288	66288	66288	66288	66288	66288	66288	66288
Unique	7982	684	491	730	75	211	10	956
Data Frame B: Value Counts								
Median	104	112	184	91	5804	796	35248	68
Max	956	660	1983	942	11591	3571	47169	260
Data Frame C: Resource Usage Percentages								
Median	-	1.37%	1.27%	1.37%	0.66%	0.91%	0.21%	1.47%
Max	-	3.03%	2.86%	2.86%	2.56%	2.70%	2.08%	3.03%

First, validating data frame A indicated the cleaned Bank dataset contains 66,288 rows with no missing values. More importantly, there are 7982 unique Resources and 956 unique Employee IDs (DN). However, there are only 10 unique Company Codes (Ccode), indicating this feature may not offer insight. Moving on to value counts in data frame B, Company Code is confirmed as a generalized attribute by the high counts. Resource indicates the average resource is granted to 11% of employees (104 out of 956), while there also exists a resource provided to all 956 employees. The DN column shows that the average employee can access 68 resources, and there exists an employee with up to 260 resources granted. This deviation from the mean can be used to classify general anomalies. Data frame C does not offer outliers at a glance, however, the normalized resource usage percentages were analyzed by algorithmic classifiers as well.

4.2 Classifying Categorical Outliers

The preliminary metric used to flag employees possessing access they do not require was counts of total resources granted to each employee. Employees with resource counts far above the mean are likely outliers. Figure 4.1 is a violin plot portraying the distribution of the number of resources granted to each of the 956 employees. Violin plots visualize a combination of a box and whisker plot, as well as a kernel density plot. The box and whisker plot is used to portray data centroids and outliers, while the kernel density plot is a smoothed histogram indicating frequencies for each bucket. Various thresholds from Figure 4.1 are also quantified in Table 4.2.

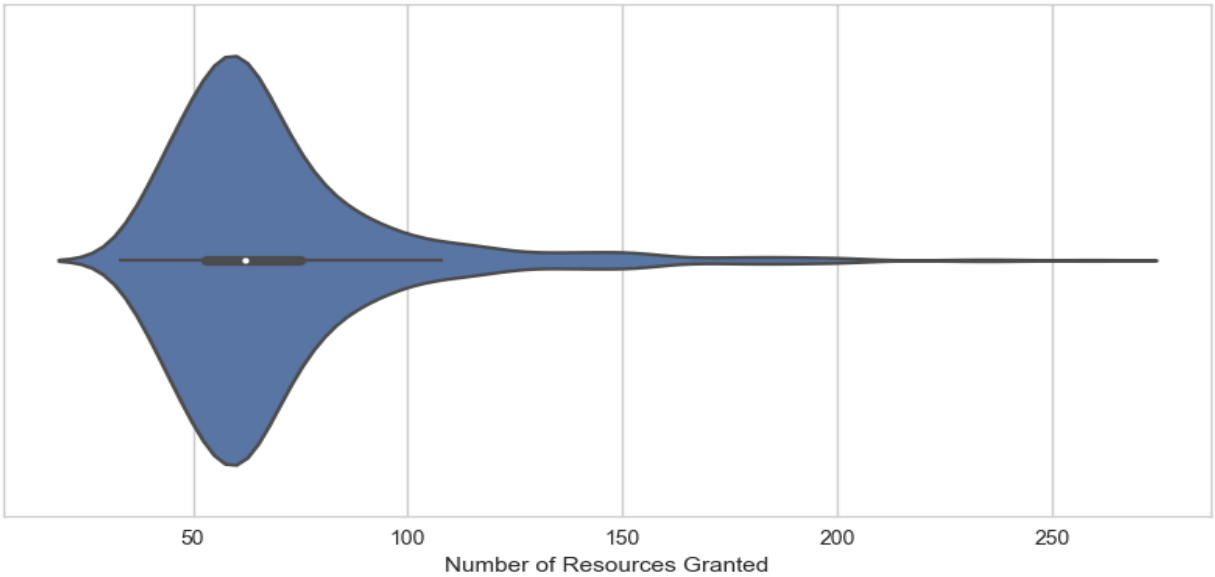

Figure 4.1 Violin Plot of Resources Granted per Employee

Table 4.2 Employees over the Resources Granted Threshold

Resources Granted Threshold	25	50	100	150	200
Frequency of Employees	956	770	95	27	5

Figure 4.1 indicates the vast majority of employees have access to between 30 and 80 resources. The peak stretches to a maximum over 260, with five employees possessing access permissions to over 200 resources as summarized in Table 4.2. All 27 employees over 150 resources granted are also part of Figure 4.1's long tail. It also interesting that the kernel distribution in Figure 4.1 begins to slightly widen again around 150 resources. This may indicate there is a small group of users such as network administrators or senior employees requiring high resource access levels compared to the average employee.

A violin plot was also generated for the value counts of managers, as portrayed in Figure 4.2 below. The purpose of this plot is to possibly identify managers that are overly lenient when granting access to their underlying employees. Outlying managers can be flagged based on a high number of occurrences, similar to employees in Figure 4.1. The majority of managers have less than 100 occurrences, while the maximum stretch to 660.

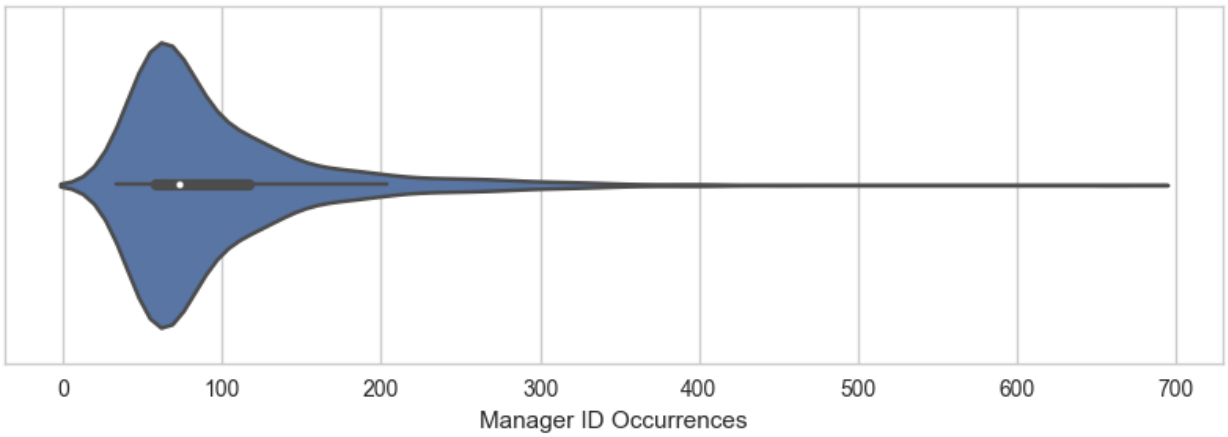

Figure 4.2 Violin Plot of Unique Manager ID Occurrences

The issue with the general classification drawn from Figures 4.1 and 4.2 is the inability to identify which resources involving the outlying employees or managers are the actual anomalies, and which are ordinary. Section 4.2 implemented a preliminary method to view baskets of resources, while a granular analysis of individual resource classification will be presented next.

4.3 Anomaly Detection using Ensemble of Feature Pairings

To classify individual rows in the Bank dataset as anomalies or ordinary, PyOD was used as discussed in Section 3.4. Three classifiers were run on resource usage percentages for each unique pairing of the seven features, creating 21 feature combinations and 63 classifications in total. A sample of the three classifiers applied to one feature pairing is portrayed by Figure 4.3, with outliers indicated by black dots lying near or outside edges of the orange inlier region. The red line depicts the calculated outlier decision function, while blue contours indicate outlier severity. The visualizations of all 21 feature combinations can be found in Appendix B.

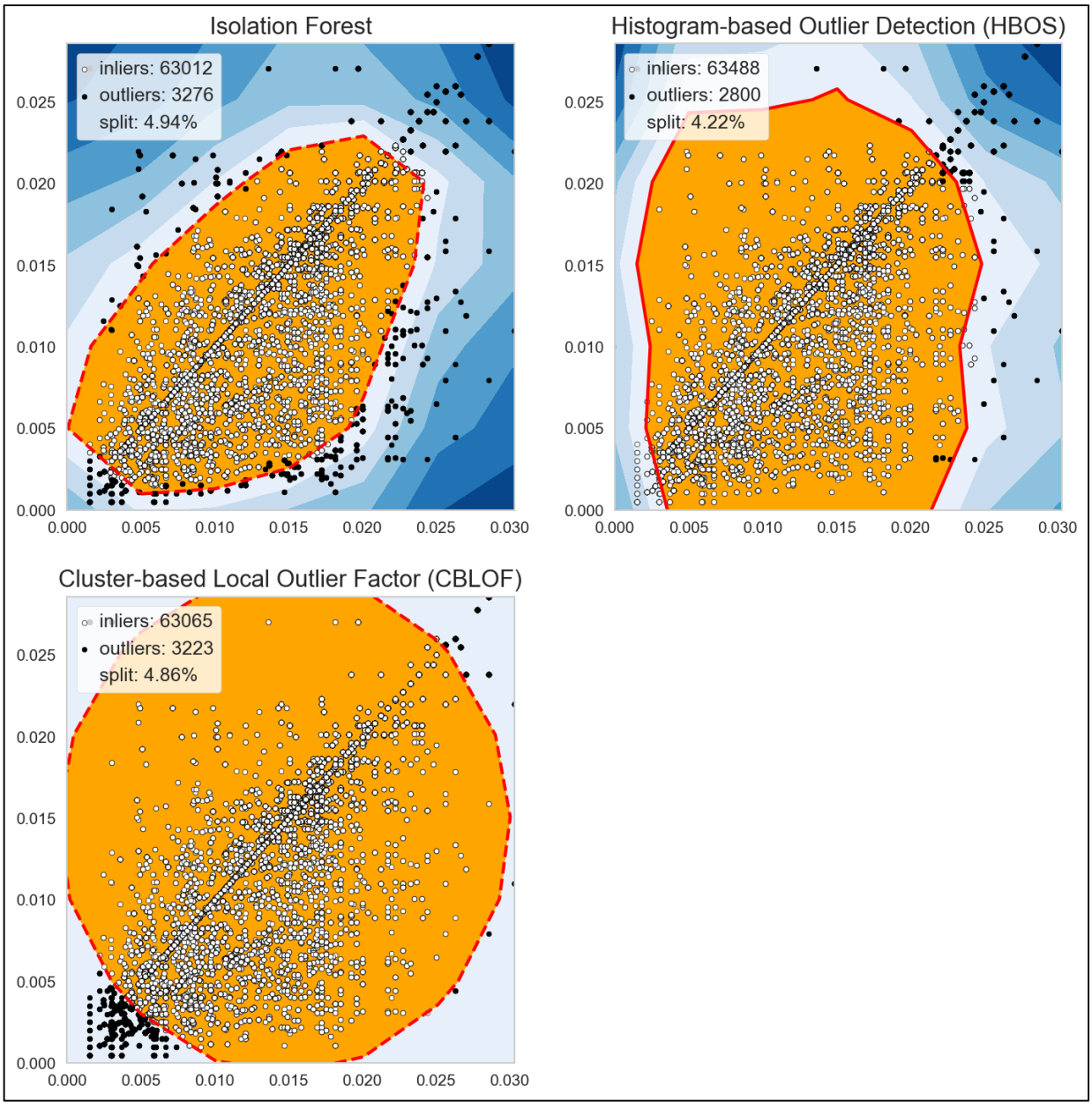

Figure 4.3 Manager vs Department - Resource Usage Outlier Classification

Isolation Forest is consistently the most sensitive of the algorithms since it has the highest split percentage, followed by Cluster Based Local Outlier Factor, and then Histogram Based Outlier Detection. The complete results of the Bank dataset from all 63 iterations can be found in Appendix C. An ensemble was formed by aggregating the results using voting summation. Each of the 63 classifiers identified a row as an outlier or ordinary by a 1 or 0 respectively. The Outlier Score represents the aggregation of classifiers up to a maximum of 63. These results are presented in Table 4.3, using various thresholds for the outlier score to label anomalies. The outlier scores are also visualized as a violin plot in Figure 4.4.

Table 4.3 Classification Ensemble Results (Max Score of 63)

Outlier Score Threshold	Outlier Count	Inlier Count	% Outliers
0	25820	40468	38.95
5	9546	56742	14.4
10	5459	60829	8.24
15	3684	62604	5.56
20	2491	63797	3.76
25	1848	64440	2.79
30	1300	64988	1.96
35	973	65315	1.47
40	718	65570	1.08
45	529	65759	0.8
50	312	65976	0.47
55	21	66267	0.03
60	0	66288	0

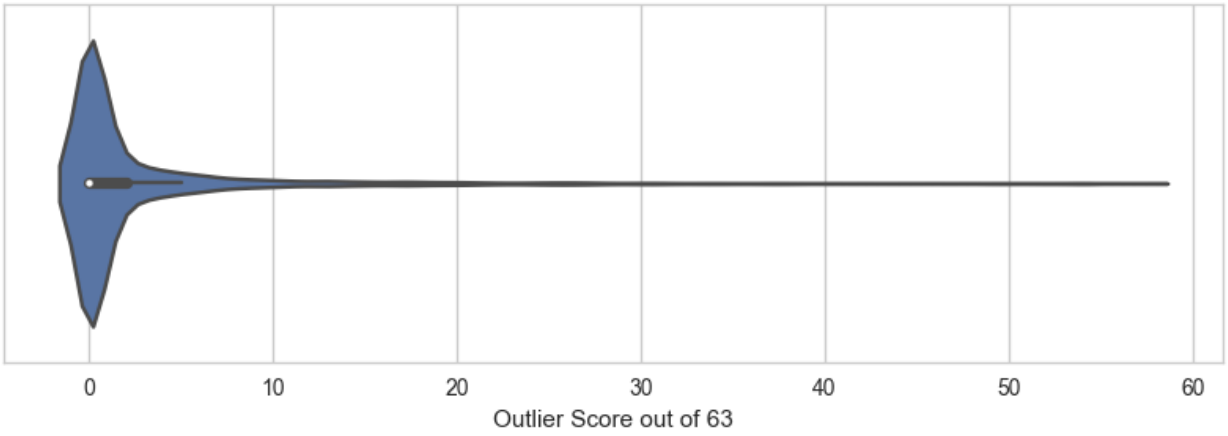

Figure 4.4 Violin Plot of Outlier Score determined by Ensembling Classifiers

The violin plot depicts that outlier scores above 10 out of 63 are rare, encompassing only 8.2% of access control mapping in the Bank dataset (Table 4.3). Over 60% of the 66.3k rows

were not flagged by a single classifier, determining the user resource permission mapping as ordinary. To maximize precision and avoid false outlier detections the threshold can be raised, increasing the likelihood that the resource mapping encompassed in outlying baskets are true anomalies. The maximum outlier score that an entry received on the Bank dataset was 57 out of 63 classifiers.

While possible to ensemble all seven tested classifiers as opposed to just three, Angle Based Outlier Detection, Feature Bagging, K- Nearest Neighbours (KNN) and Average KNN were deemed unsuitable for application to the Bank dataset. Comparable to Figure 4.3, Figure 4.5 visualizes the four omitted classifiers on Manager and Department resource usage as well. It is clear the black dots indicating outliers in Figure 4.5 are far less distinct from inlier regions.

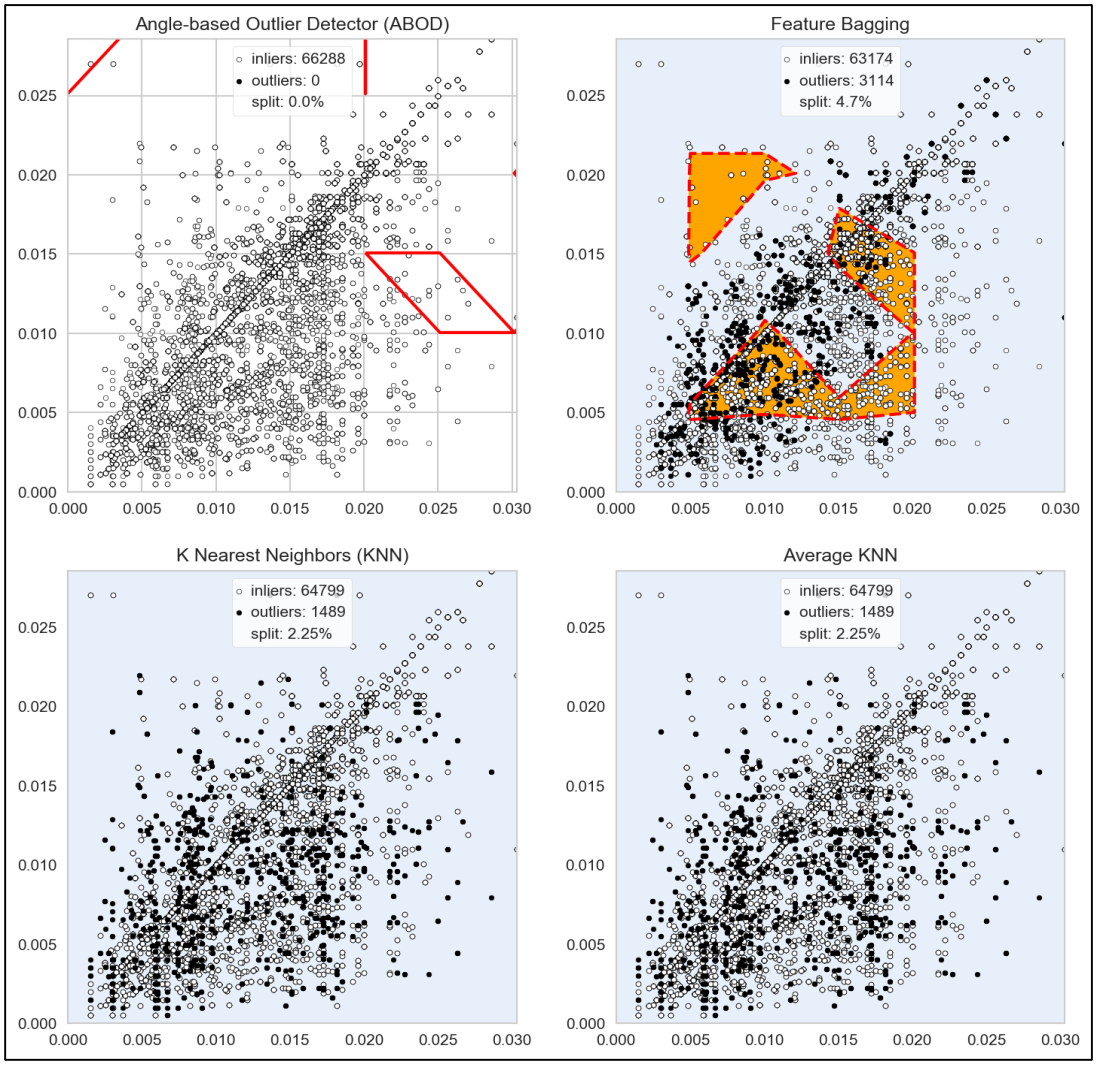

Figure 4.5 Unsuitable Classifiers - Manager vs Department Resource Usages

4.4 Anomaly Detection on Full Dimensionality Dataset

PyOD was also used to classify the entire 66.3k x 7 dataset of resource usage percentages, as opposed to classifying individual feature pairings as presented in Section 4.3. The results from this alternative analysis are shown below in Table 4.4. Feature Bagging, K- Nearest Neighbours and Average K- Nearest Neighbours were also deemed unsuitable models as a large portion of outliers detected in Figure 4.5 and other iterations were deemed ordinary. Angle Based Outlier Detection was unable to classify any anomalies.

Table 4.4 Results of PyOD Classifiers on Full Dimension Resource Usage Classification

Classification Algorithm	**Outliers**	**Inliers**	**% Outliers**
Isolation Forest	3314	62974	5.00
Histogram-base Outlier Detection (HBOS)	3313	62975	5.00
Cluster-based Local Outlier Factor (CBLOF)	3311	62977	4.99
Feature Bagging	*3121*	*63167*	*4.71*
K Nearest Neighbors (KNN)	*2774*	*63514*	*4.18*
Average KNN	*1429*	*64859*	*2.16*
Angle-based Outlier Detector (ABOD)	*0*	*66288*	*0*

The results from Isolation Forest, HBOS and CBLOF indicate that 5%, or 3.3k out of 66.3k resource access entries in the Bank dataset are possible anomalies. Comparing this count to the ensembling performed in Table 4.3, an Outlier Score Threshold of 16-17 out of the 63 classifications produced would replicate a similar outlier to inlier split.

4.5 Prediction of Future Access Control Mapping

The Bank dataset presented in data frame A of Table 4.1 (Section 4.1) was appended with an Action feature containing a value of 1 for approved. An additional 4k synthetic resource denial rows (Action of 0) were created as explained in Section 3.5. An 80/20 split was then used on both the original and synthetic data to yield the datasets overviewed in Table 4.5. The 80/20 split randomly selects 80% of the data for training, and uses the remaining 20% for testing.

Table 4.5 Bank Training and Testing Data with Synthetic Resource Denial Rows

	Resource	Bufugu	Business	Ccode	Department	DN	Manager	Title	Action
Training Dataset									
Count	56230	56230	56230	56230	56230	56230	56230	56230	56230
Unique	7114	75	211	10	491	956	684	730	2
Top Freq.	781	9786	3014	39968	1672	222	534	791	53030
Testing Dataset									
Count	14058	14058	14058	14058	14058	14058	14058	14058	*14058*
Unique	3108	75	211	10	491	956	684	730	*2*
Top Freq.	221	2486	760	10056	417	51	160	200	*13258*

Comparing Table 4.5 to the original cleaned Bank dataset in Table 4.1 shows that unique resources in the training dataset dropped from 7982 to 7114, indicating 868 resources were transferred solely to the testing set. The remaining 2240 resources in the testing set (3108 minus 868) are also present in the training set. Furthermore, all 956 employees from the original dataset appear in both the training and testing datasets, as expected since all employees can access at least 25 resources. The similar uniqueness indicates the randomization used to synthesize and select rows for training or testing was adequate. Finally, the Action column dictates 53k resource approvals out of 56.2k rows for the training set, or an approval rate around 94.3%. The Action entries in the testing dataset were deleted since Action is the target for prediction.

The datasets in Table 4.5 were analyzed by Benjamin Solecki's ensemble.py classifier as described in Section 3.5, with minor modifications to suit the Bank dataset. Random Forest, Extra Trees and Gradient Boosting models were each fit to the Bank training set. The trained models then predicted resource approval (Action) in the testing set. The model predictions on the test set are presented in Table 4.6, along with statistical summaries. The resulting percentages represent the probability that each row of the testing set would be approved with an Action of 1.

Table 4.6 Classifying Testing Set Approval Probability

Classifier	Random Forest	Extra Trees	Gradient Boosting
Count	14058	14058	14058
Mean	92.299%	90.974%	97.532%
Std. Deviation	9.614%	7.869%	12.499%
Minimum	7.091%	4.171%	0.013%
25th Percentile	89.633%	89.215%	99.668%
50th Percentile	95.873%	93.256%	99.692%
75th Percentile	98.086%	95.377%	99.696%
Maximum	100.000%	100.000%	99.699%

It is known that correct resource approval rate for the testing set is 94.3%, hence, the classifiers' mean approval rates fall within an acceptable range. Analyzing the means presented in Table 4.6, Random Forest and Extra Trees under predicted resource approval, while Gradient Boosting over approved resources to be granted to employees. These prediction results and possible improvements will be further discussed in Section 5.5.

4.6 Chapter Summary

Value counts were generated on the cleaned Bank dataset to identify general outlying IDs, including employees and managers. There existed employees granted access for up to 260 resources compared to the median of 68. Resource usage percentages for each attribute were generated to perform a granular analysis and label individual employee resource pairings. Feature pairings of the seven resource usage percentages were run through three outlier classifiers, yielding a total of 63 classifications. The iterations were aggregated into an outlier score out of 63, while any score above 10 qualifies for investigation as an outlier. This threshold includes up to 8.2% of access control mapping, however, increasing the threshold reduces the anomaly subset size. Three outlier classifiers were also applied to the full dimensionality dataset, detecting 5% of the 66.3k resource mappings as possible outliers.

Resource denial rows were also synthesized from the Bank dataset and used in resource mapping prediction models. Three classifiers were trained using an 80/20 split and applied to the testing set yielded approval rates ranging from 91% to 97.5%. The true value of the testing approval rate was 94.3%, demonstrating that the classifiers were viable predictors. Actionable recommendations from these results and future work will be explained next.

Chapter 5: Conclusions, Recommendations and Future Work

This section will further analyze the methodology and results presented in Chapters 3 and 4 respectively. Actionable recommendations based on the results will be stated, and future work will also be outlined for each stage, expanding on the methodology and analysis that has been already be conducted. The future work will be emphasized in certain areas, to better verify the validity of outlier classification results and improve accuracy in access control prediction. Dataset recommendations will be examined first, followed by access control anomaly classification, and finally access control mapping prediction.

5.1 Examination of Dataset Suitability

The MS Access XML data that was provided by the project sponsor was deemed sufficient for the purposes of this thesis. The cleaned Bank dataset summary in Table 4.1 indicates sufficient length and uniqueness for analysis, however, accuracy for both outlier classification and resource mapping prediction could benefit from more data points. While MS Access limits exports to 1000 employees, multiple exports can be aggregated. The conversion from XML flat file to tabular data is reproducible using the data parser created, as is the data transformation and feature engineering into value counts and resource usage percentages.

With regards to resource mapping prediction, it would be ideal to utilize documented resource request denial data as opposed to appending an Action feature (resource approval) and synthesizing the 4k denial rows. It is possible that several synthesized rows were severe anomalies involving employee resource combinations, as opposed to reasonable access requests an employee may make. However, synthetic rows are unlikely to be avoided as the project sponsor's resource requests are either a verbal transaction, or one occurring outside MS Access documentation capabilities. An alternate solution may involve gathering access control data from sources other than MS Access, which could provide approval and denial information.

Expanding on alternate datasets, MS Access stores a high quantity of attributes other than the eight features (see Table 4.1) examined in this thesis. Future experimentation should be performed to explore other attributes that may offer more insight than the selected subset of eight. In particular, Company Code is a replaceable feature due to a low uniqueness throughout the dataset. Expanding the attributes universe will also impact and improve feature engineering.

5.2 Feature Engineering

Feature engineering was a crucial stage in the methodology (Section 3.3), as the original dataset is composed of categorical text. The conversion to value counts and resource usage percentages created suitable data for both access control outlier classification, and the prediction of resource mapping. A further analysis of resource usage percentages is presented in Figure 5.1, which depicts correlations between the 21 unique pairings of the seven usage features utilized for prediction and anomaly detection. The high correlations indicate particular pairings may not have benefited the ensemble, and instead overweighed the importance of certain attributes.

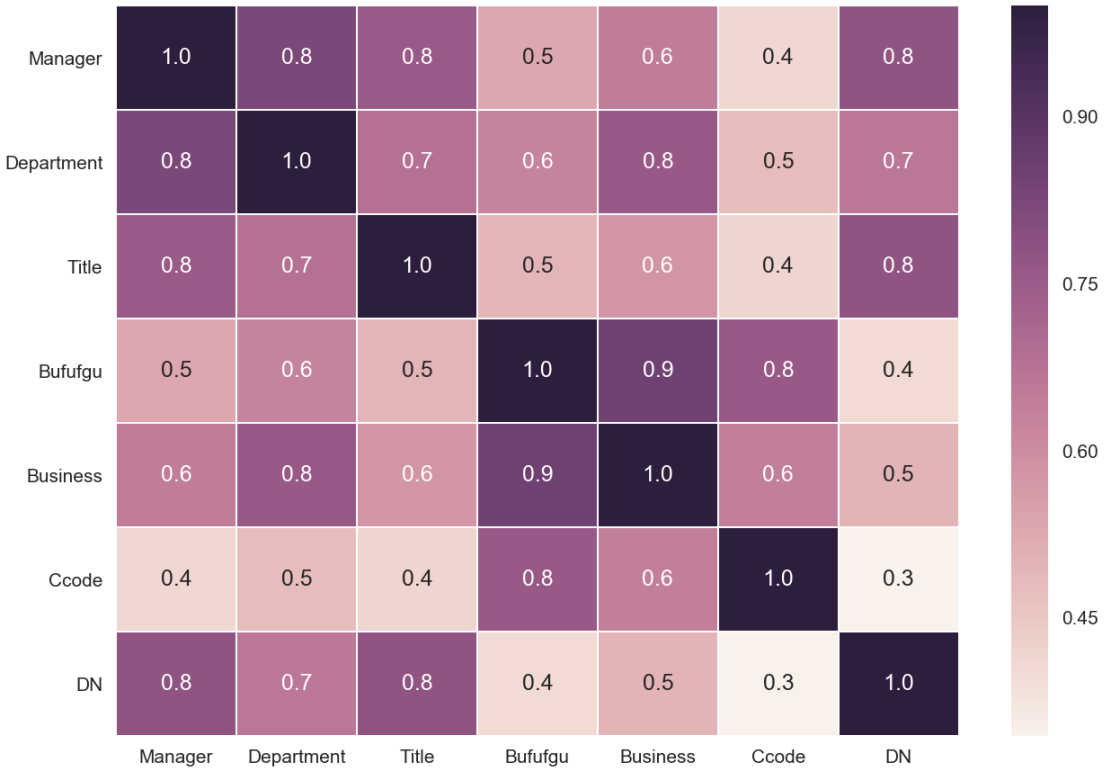

Figure 5.1 Correlation Heatmap of Resource Usage Percentages

Alternative feature engineering was tested, although the project timeline did not yield a complete and comprehensive examination. Experimentation involving a variety of feature engineering techniques is likely to yield superior datasets that could offer more insight, and is a highly recommended process for future work. The variety of solutions to Kaggle's Amazon Access Challenge can be referred to for inspiration [16].

5.3 Generalized Anomaly Detection

As discussed in the introduction to Chapter 4, outliers and anomalies identified do not indicate a gap in the project sponsor's access control. The results are a guidance for investigating existent access control mapping, and an opportunity to further maximize enterprise cybersecurity. If outliers are determined true anomalies, eliminating excess access control mapping reduces data loss vulnerabilities due to employee error, or workstation compromise through theft and malware.

The categorical outliers identified in Section 4.2 can be interpreted as employees granted an abundance of access compared to the average resources granted per user (68). All 27 employees above the 150 resources granted threshold in Table 4.2 should undergo a preliminary review to deem if their attributes fit their access profiles. Anonymized Employee IDs from any bucket grouping number of resources granted can be retrieved using filtration on the value counts dataset. These employee ID's can be decrypted with the hashing key used to anonymize the entries, leading to actionable insight. If employees are deemed to possess unnecessary resource permissions, eliminating these mappings improves enterprise network security.

Similarly, the occurrence counts of managers presented in Figure 4.2 can be analyzed to determine whether their underlying employees possess abundant access. If a manager is deemed overly lenient, further guidance on access control approval may be administered. While this generalized classification can aid a basic access control audit, it is recommended to perform a more granular analysis and review specific outlying employee resource pairings as well.

5.4 Individual User-Resource Anomaly Detection

The critical features analyzed in this thesis are the resource usage percentages generated on the remaining seven features. The results of applying three outlier classification algorithms to all unique resource usage feature pairings can be viewed in Section 4.3. The visualization of all 63 resource usage feature pairings may be viewed in Appendix B, while the results of each are located in Appendix C.

All seven features were paired into 21 unique combinations, although it is clear from the visualizations in Appendix B that segments of the 63 classifications may not have yielded

beneficial results. Poor feature pairing is particularly true for combinations involving company code. In future applications, more feature selection could occur to reduce noise added by unsuitable classifiers. The classification algorithms applied were not were not computationally expensive, hence feature extraction was chosen as a focus over feature selection.

Isolation Forest is consistently the most sensitive of the classifiers. It is highly effective when data is wide spread over both axes, since outlying data points are rather evenly spread throughout the Isolation Forest plots [17]. While Cluster Based Local Outlier Factor is nearly as sensitive as Isolation Forest, outlying data points are collected in more distinct regions towards a corner or edge of the plots. Histogram Based Outlier Detection has the lowest sensitivity of the three classifiers used, and is likely to yield less false positives defined as labelled outliers that are ordinary data points. To minimize the scope of access control anomalies to be reviewed, the histogram based model offers the smallest subset of outlying cases. However, it is recommended that an ensemble of all three algorithms is used to minimize false negatives defined as missed outliers labelled as ordinary.

The ensembled results of the three outlier classification algorithms can be found in Table 4.3 and Figure 4.4, based on Outlier Scores out of 63. The results of each of the 63 iterations from Appendix C can be linked by classification key (e.g. B2) to an outputted dataset containing the non-aggregated classifications. The index for each of the 66.3k rows matches to a resource granted row in the original Bank dataset, which can be decrypted for review. When selecting a threshold for Outlier Score, the outlier counts in Table 4.3 may be used to control the scope of resources requiring review. It is recommended to first select a high threshold such as 50 out of 63, and confirm that a significant portion of the 312 outliers detected are indeed true outliers. If not, it is unlikely lower thresholds would have higher precision and outlier review would not be beneficial.

Furthermore, possible outliers should be reviewed by descending Outlier Score, in order to prioritize the most anomalous access control mapping. The stopping point for Outlier Score can be determined by the repetitive presence of falsely labelled outliers, or using the results of outlier classification performed on the full 66.3k x 7 resource usage dataset. In future extensions, it is possible that stacking, which would combine the 63 classifier outputs using another classification

model, would be a better aggregation technique than the summation method used.

An alternative source of outlier classifications is presented in Section 4.4, by applying classifiers to the full resource usage dataset without creating feature pairs. The alternate approach determined that 5% of the entries were detected outliers by all three classifiers applied (Table 4.4), which pairs to an ensembled Outlier Score between 16 and 17. The 5% split indicates a stopping point for reviewing labelled outliers. It is recommended that if a large portion of high Outlier Score threshold anomalies labelled by the ensemble are true outliers, at least 3.3k out of 66.3k access control mappings (Table 4.4) should be reviewed. Future work would include testing outlier classification techniques outside the three algorithms selected and the four algorithms deemed unsuitable for this project (Table 4.4). PyOD toolkit for Python offers over 20 outlier classification models [17], or other outlier detection toolkits could be applied.

5.5 Verifying the Prediction of Access Control Mapping

The results for attempting to predict access control mapping with collaborative filtering used three classifiers presented in Table 4.6. The classifiers applied include Random Forest, Extra Trees and Gradient Boosting. The true approval rate of the testing set was 94.3%, which is known since the denial rows were synthesized. Random Forest was the closest classifier at an average approval rate of 92.3%, while Extra Trees under predicted at 91.0%, and Gradient Boosting over predicted at 97.5%. Interestingly, the 25th percentile indicates a 99.7% approval probability from Gradient Boosting, which is nearly the maximum. The high probability is expected with only 6% of rows true resource request denials. However, the similarity between Gradient Boosting's 25th percentile and maximum showcases the decisive nature of the classifier, as compared to Random Forest and Extra Trees.

As mentioned in the dataset examination of Section 5.1, the main improvement for the prediction stage requires using real resource request denial data instead of synthetic rows. Although, there is significant future work to be performed on model development as well. Firstly, the dataset with synthesized rows appended was divided into a training and testing set using an 80/20 split. The Amazon dataset provided an additional testing set that was a magnitude larger than the training set. Hence, the models drawn from Benjamin Solecki's solution to Kaggle Amazon Access Challenge may have over fit the Bank training set.

Furthermore, the Bank dataset was cleaned and transformed to mimic the structure of the Amazon dataset, in order to build on relevant challenge solutions. However, the predictors in these solutions were not optimized for the Bank dataset. Future work should test alternative hyper-parameter choices that may increase prediction accuracy. Experimentation should also test a further array of prediction technique including the full Amazon Access Challenge winning solution. The winning solution ensembled Benjamin Solecki's solution [18] applied in this project, as well as classification and ensembling algorithms created by Paul Duan [16]. Paul Duan's contributions were experimented with for this project, however, timelines did not allow for presentable results to be produced.

The Kaggle Amazon Access Challenge occurred in 2013 [15], while machine learning techniques have undergone significant development since that period. The preliminary prediction conducted validates recommendation systems as a viable access control mapping technique. The main business case for the project sponsor was the detection of anomalies in current access control mapping. Therefore, focus was placed on outlier classification as opposed to access control mapping prediction due to project timelines. More importantly, emerging techniques that have become popularized since 2013 can yield prediction accuracy beating Solecki and Duan's algorithms on both the Amazon and Bank datasets. Spectral Clustering [20] and Restricted Boltzmann Machines have been explored as applicable recommendation systems for access control prediction [14], and can be expanded upon in future work.

5.6 Chapter Summary

The conclusion of this project is that 5% of the existent access control mapping analyzed were determined actionable outliers. It is recommended that the identified anomalies be reviewed in order of descending Outlier Score, assigned by ensembling 63 feature pairing classifications aggregated by summation. If a large portion of high Outlier Scores are deemed ordinary access control mapping, the remainder of the 5% is unlikely to be classified as anomalous and access control review would be unproductive. The removal of unnecessary employee resource mapping would maximize enterprise cybersecurity. Minimal access granting reduces vulnerabilities and data loss caused by employee miscues, or situations in which a workstation is subject to malware or theft.

Recommendation systems can also be applied to datasets possessing existent access control mapping and user attributes. Resource request approval prediction was classified with high accuracy, proving recommendation systems as a viable method for future access control mapping. The benefits include providing a dynamic method to grant and eliminate resource access control, helping prevent employee downtime and reduce network administration costs.

Works Cited

[1] C. O. a. R. J. Loomis, "Economic Analysis of Role-Based Access Control: Final Report," CSRC, 19 December 2010. [Online]. Available: https://csrc.nist.gov/publications/detail/white-paper/2010/12/19/economic-analysis-of-rbac-final-report/final. [Accessed 15 October 2018].

[2] Wang, H., Guo, X., Fan, Y. and Bi, J. (2014). Extended Access Control and Recommendation Methods for Enterprise Knowledge Management System. IERI Procedia, 10, pp.224-230.

[3] Michael Backes and Peng Ning, Computer Security - ESORICS 2009. Springer Nature., 2009.

[4] A. A. Elliott & G. S. Knight. "Role Explosion: Acknowledging the Problem". Proceedings of the 2010 International Conference on Software Engineering Research & Practice. 2010.

[5] Hu, Vincent C.; Ferraiolo, David; Kuhn, Rick; Schnitzer, Adam; Sandlin, Kenneth; Miller, Robert; Scarfone, Karen. "Guide to Attribute Based Access Control (ABAC) Definition and Considerations", 2014.

[6] Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger. “Research Paper Recommender Systems: A Literature Survey.” Proceedings of the International Workshop on Reproducibility and Replication in Recommender Syst:1–34. doi:10.1007/s00799-015-0156-0., RepSys, 2015.

[7] F. Ricci, L. Rokach, and B. Shapira, Recommender Systems Handbook. Boston, MA: Springer US, 2015.

[8] "Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition - NETFLIX 08," 2008.

[9] Rubens, Neil; Elahi, Mehdi; Sugiyama, Masashi; Kaplan, Dain. "Active Learning in Recommender Systems". Springer US., 2016.

[10] Smith, B., & Linden, G., "Two Decades of Recommender Systems at Amazon.com," 2017.

[11] Takács, G.; Pilászy, I.; Németh, B.; Tikk, D. "Scalable Collaborative Filtering Approaches for Large Recommender Systems". Journal of Machine Learning Research. 10: 623–656,

2009.

[12] C. C. Aggarwal, Recommender Systems The Textbook. Cham: Springer International Publishing, 2018.

[13] Robert M. Bell, Yehuda Koren and Chris Volinsky, "The BellKor solution to the Netflix Prize," AT&T Labs, 2010.

[14] R. Salakhutdinov, A. Mnih, G. Hinton, "Restricted Boltzmann Machines for Collaborative Filtering," 2016.

[15] “Amazon.com - Employee Access Challenge.” Kaggle, www.kaggle.com/c/amazon-employee-access-challenge/data.

[16] "Winning Solution Code and Methodology.” Kaggle, www.kaggle.com/c/amazon-employee-access-challenge/discussion/5283.

[17] " Zhao, Y., Nasrullah, Z. and Li, Z., 2019. PyOD: A Python Toolbox for Scalable Outlier Detection. arXiv preprint arXiv:1901.01588.".

[18] Solecki, Benjamin. “Bensolucky/Amazon.” GitHub, 27 July 2015, github.com/bensolucky/Amazon.

[19] "Lakshay Arora. “Tutorial on Outlier Detection in Python Using the PyOD Library.” Analytics Vidhya, 15 Feb. 2019, www.analyticsvidhya.com/blog/2019/02/outlier-detection-python-pyod/.".

[20] Li, X., Wang, Z., Hu, R. et al. "Recommendation algorithm based on improved spectral clustering and transfer learning", Springer, 2017.

Appendices

Appendix A: Microsoft Access Sample Access Control Mapping

A sample of the hashed (anonymized) XML flat file exported from the project sponsor's Microsoft Access is shown below. All entries have been shortened and resources have been removed to reduce length. Attribute names have also been altered to provide interpretable information.

```
...
...
 <searchResultEntry dn="5335de625c5187ee45e64c23427d2bfd215184f03cfd6581dcf">
    <attr name="manager">
       <value>87c8ff02614621d46a91427c9342c049b16a8673fe929f7554feecb8fe4b0</value>
    </attr>
    <attr name="bufugu">
       <value>39a5a276d3830e76d768535f613b99b2dd7db528009e8cafd13cb9d6d68f4</value>
    </attr>
    <attr name="business">
       <value>698a9d279fd2ee19e6acfa6fcc9a73eefc6a82182421de406f3e0766c5522</value>
    </attr>
    <attr name="ccode">
       <value>275f24097e3fa70b6466eba9d87d6608d2f2bacdd5b9ca88b9af1bcd079df</value>
    </attr>
    <attr name="resource">
       <value>15799d193ec684e362291df55c4f4130867f9b670f8340865c81cc0796fe5</value>
       <value>007d8dcaf543bf3d77dba311c1c560c67f5d4b6d637353444713d8a718202</value>
       <value>39184e6db22e685f5573c16e1aed2ade6e28c301d0f897004816d32665790</value>
       <value>149a14775581e8846a3cef02aeca1ce0f3654514136a7edf1ae6763460779</value>
       <value>ec3828a140e71b9e75afa1c255d55fe27139b59d4abea3dab74504e2fed4d</value>
       <value>e920bf9f47995906ef1aed0566fce3424cb59069381e6cc63bca00f1241b8</value>
    </attr>
    <attr name="department">
       <value>e374fd69c708edf754bc3cb6df51b0c1181ef927e4a6c001156dc254bb84d</value>
    </attr>
    <attr name="title">
       <value>407de5add8cfa1328be2af5ed58d17642f2442966c61cc116b6be71b24f48</value>
    </attr>
 </searchResultEntry>
 ...
 ...
```

Figure 6.1 Sample of MS Access XML Export

Appendix B: Classification Plots of Resource Usage Feature Pairs

The following graphs visualize outlier detection using PyOD. Three classifiers were applied to 21 unique pairings of the seven resource usage percentage features in the Bank dataset, yielding a total of 63 classifications. The visualizations shown below are grouped by feature pairing. The results of each iteration may be found in Appendix C.

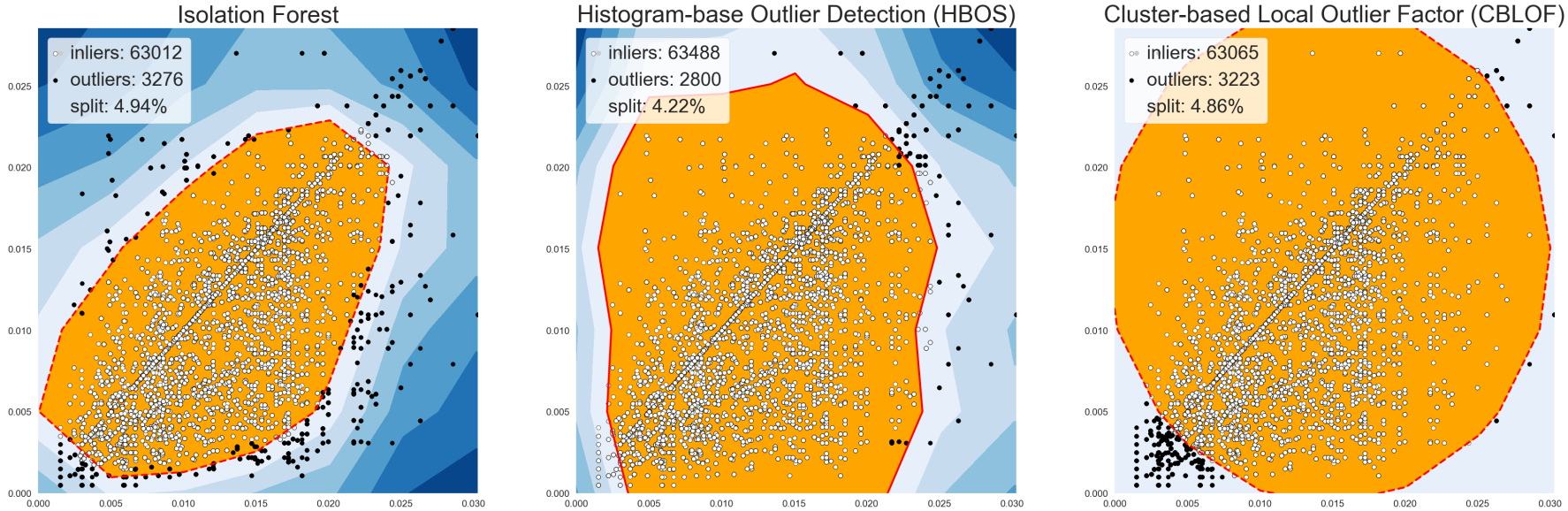

Figure 6.2 Manager vs Department – Resource Usage Outlier Classification

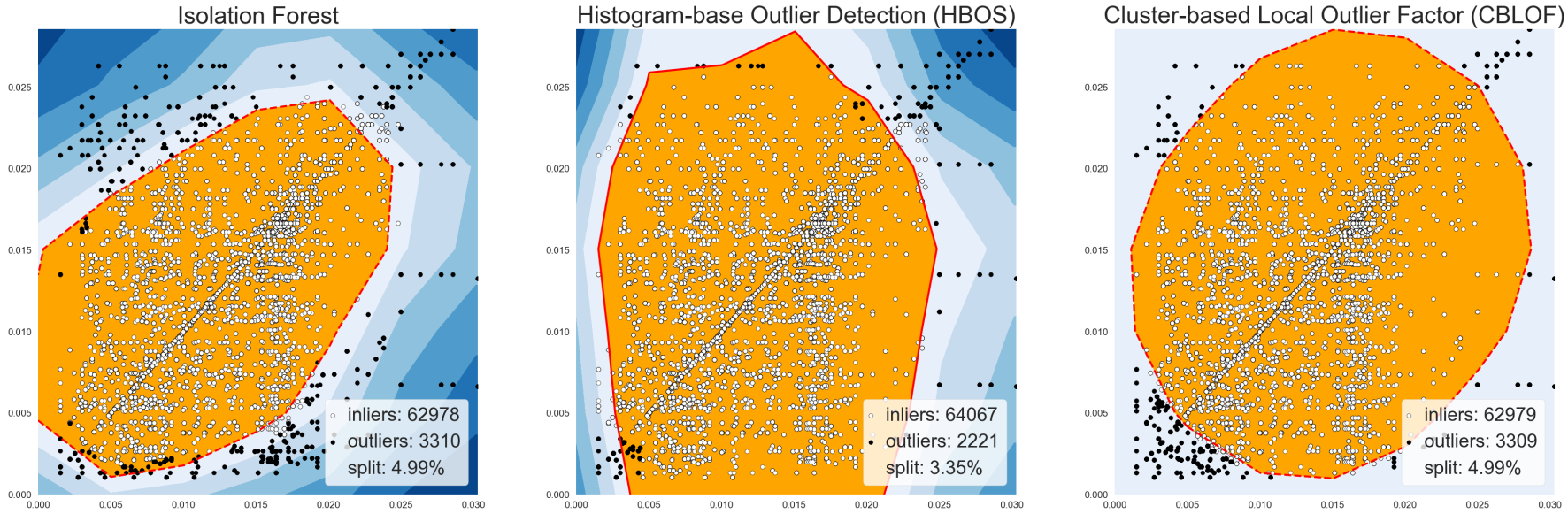

Figure 6.3 Manager vs Department – Resource Usage Outlier Classification

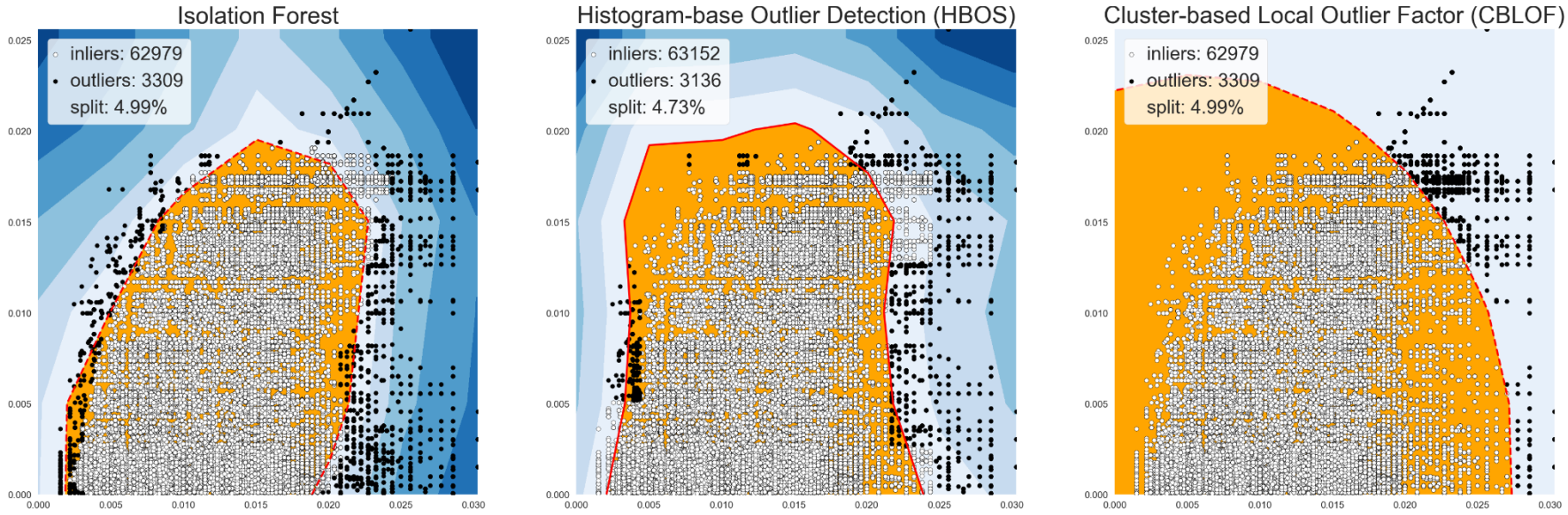

Figure 6.4 Manager vs Business Unit Group – Resource Usage Outlier Classification

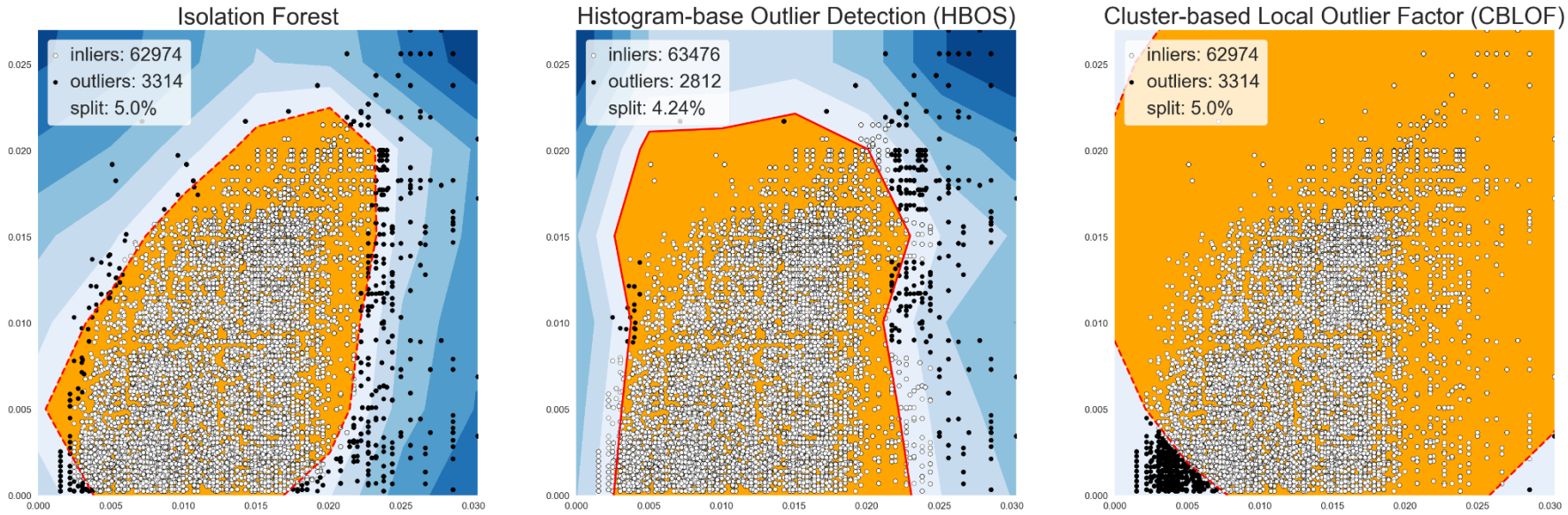

Figure 6.5 Manager vs Business – Resource Usage Outlier Classification

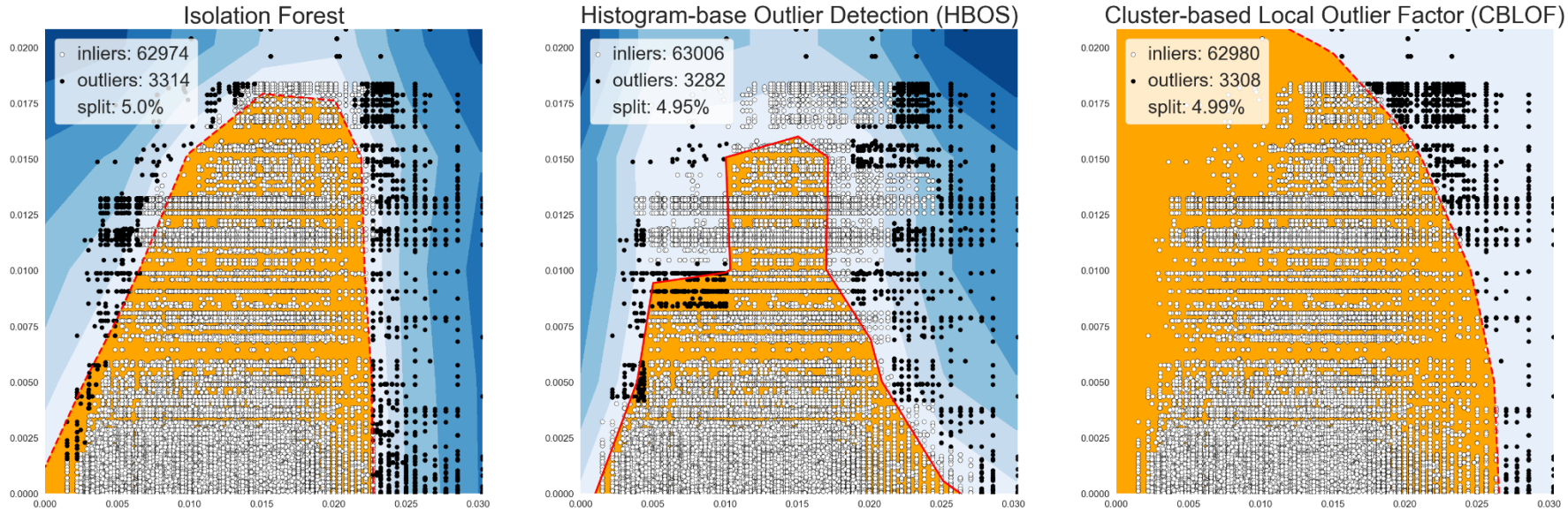

Figure 6.6 Manager vs Company Code – Resource Usage Outlier Classification

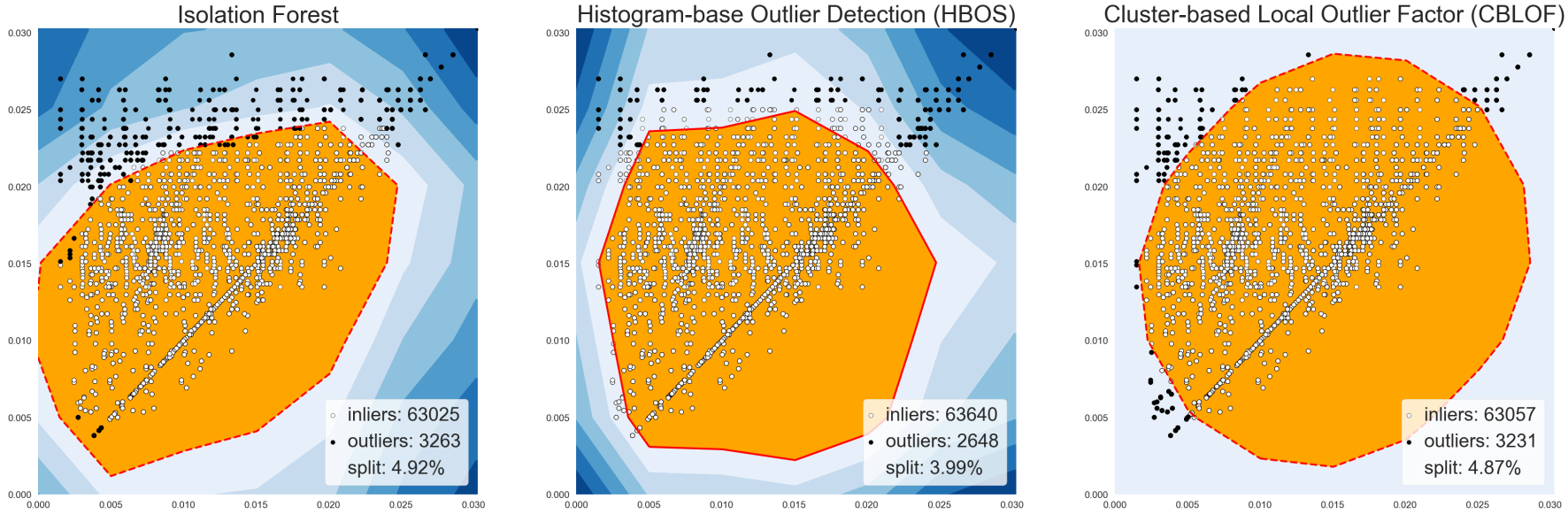

Figure 6.7 Manager vs Employee ID – Resource Usage Outlier Classification

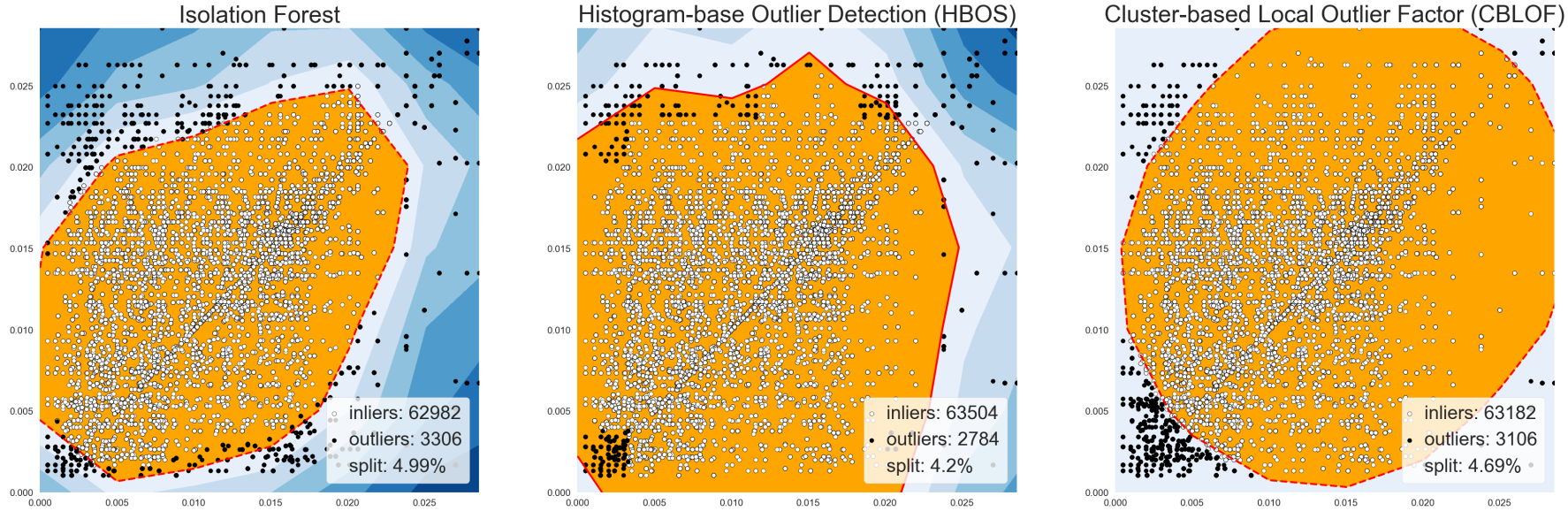

Figure 6.8 Department vs Title – Resource Usage Outlier Classification

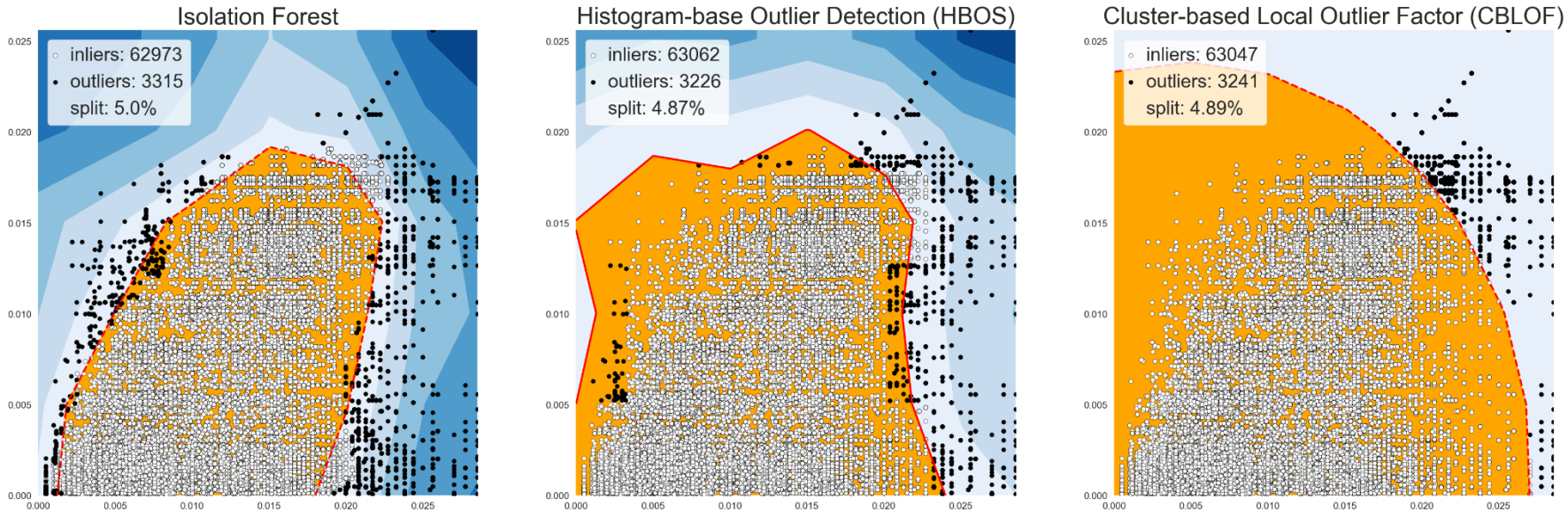

Figure 6.9 Department vs Business Unit Group – Resource Usage Outlier Classification

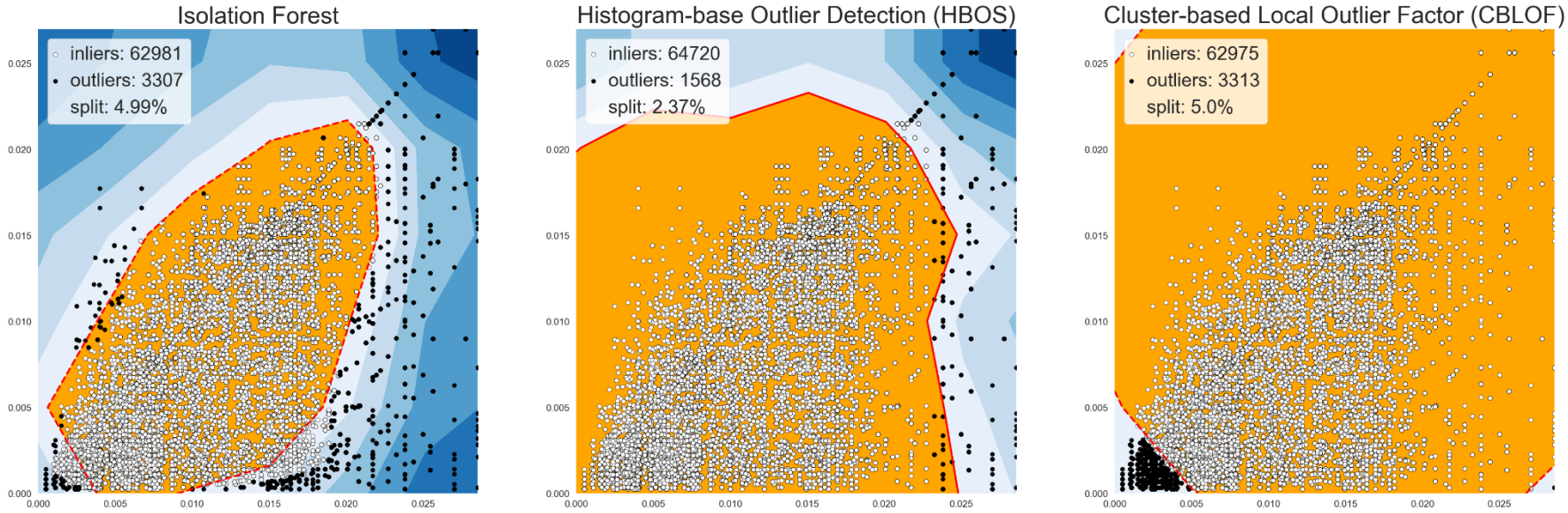

Figure 6.10 Department vs Business – Resource Usage Outlier Classification

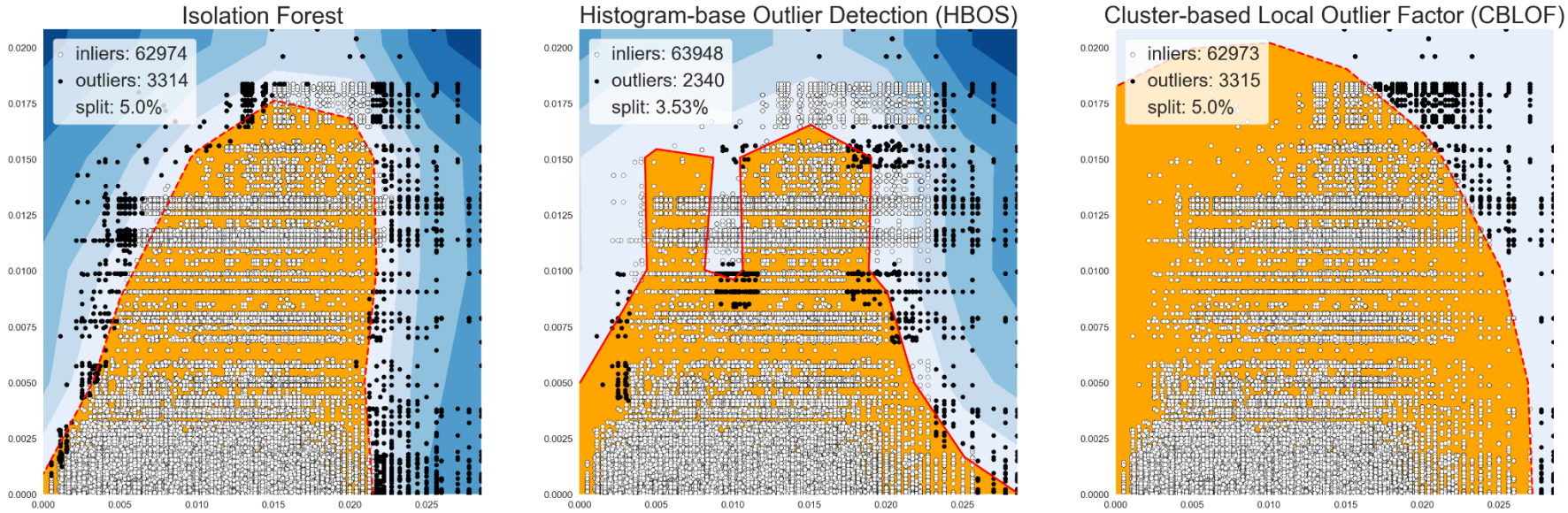

Figure 6.11 Department vs Company Code – Resource Usage Outlier Classification

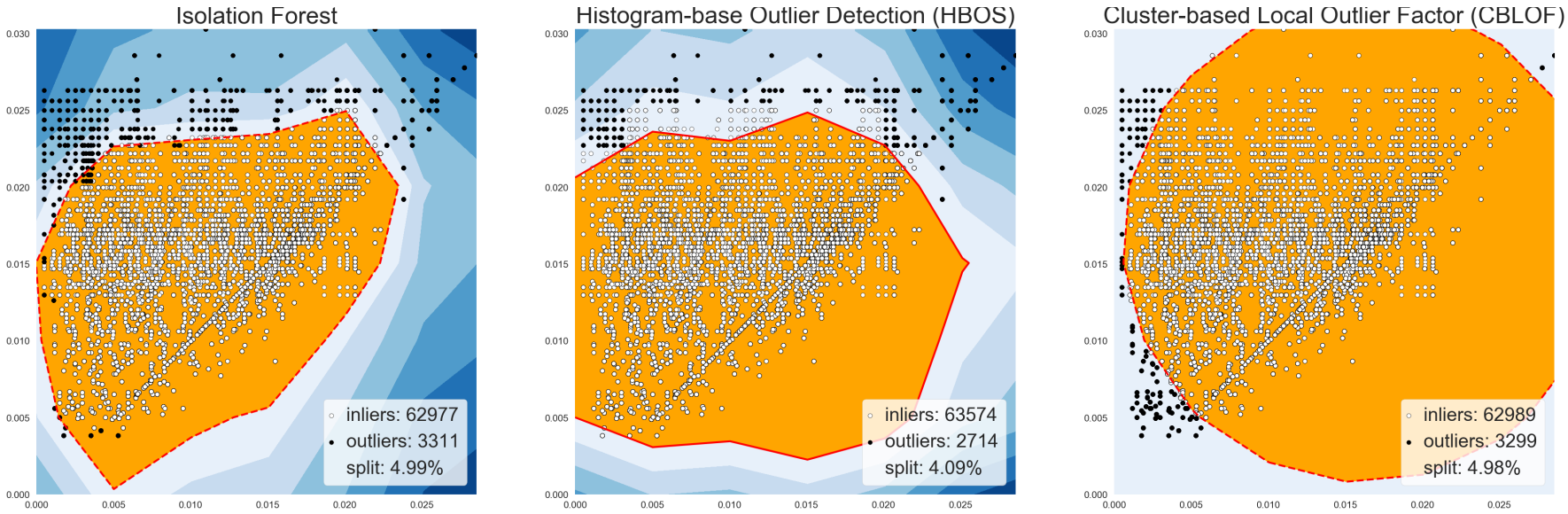

Figure 6.12 Department vs Employee ID – Resource Usage Outlier Classification

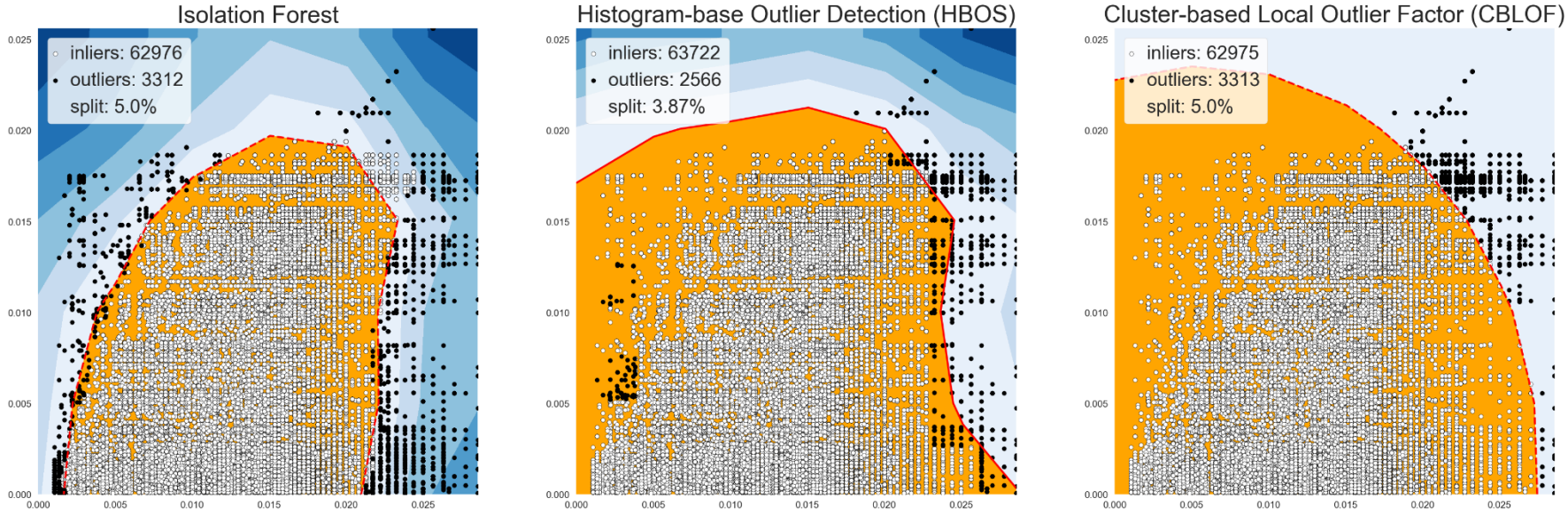

Figure 6.13 Title vs Business Unit Group – Resource Usage Outlier Classification

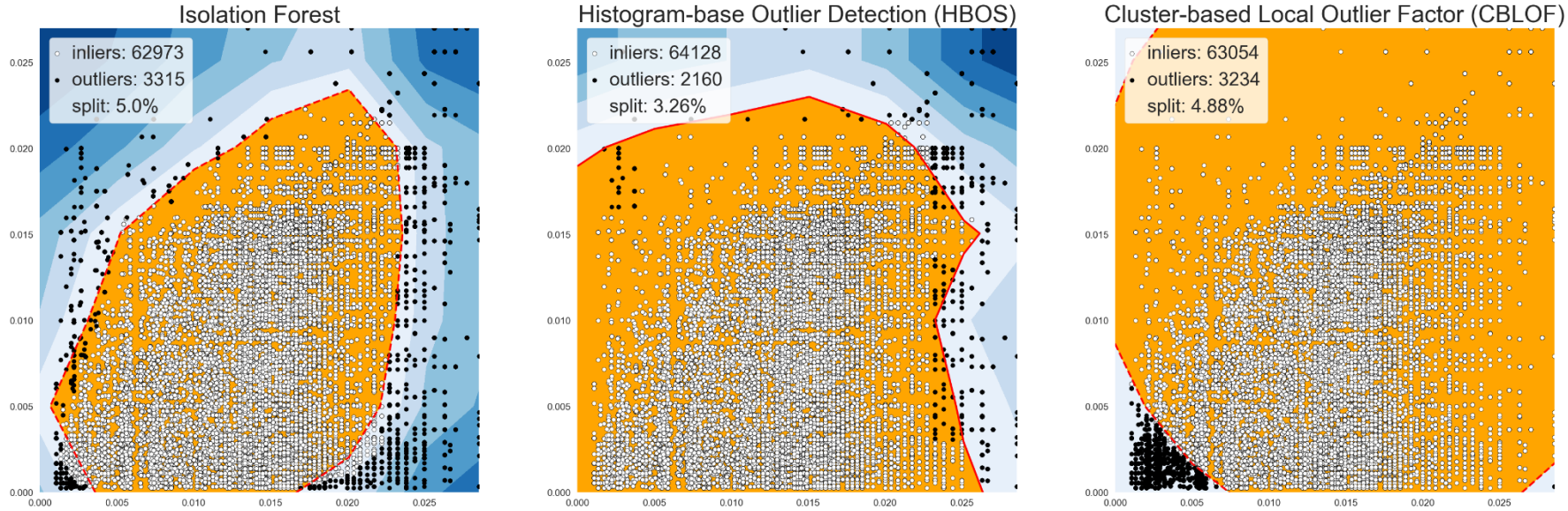

Figure 6.14 Title vs Business – Resource Usage Outlier Classification

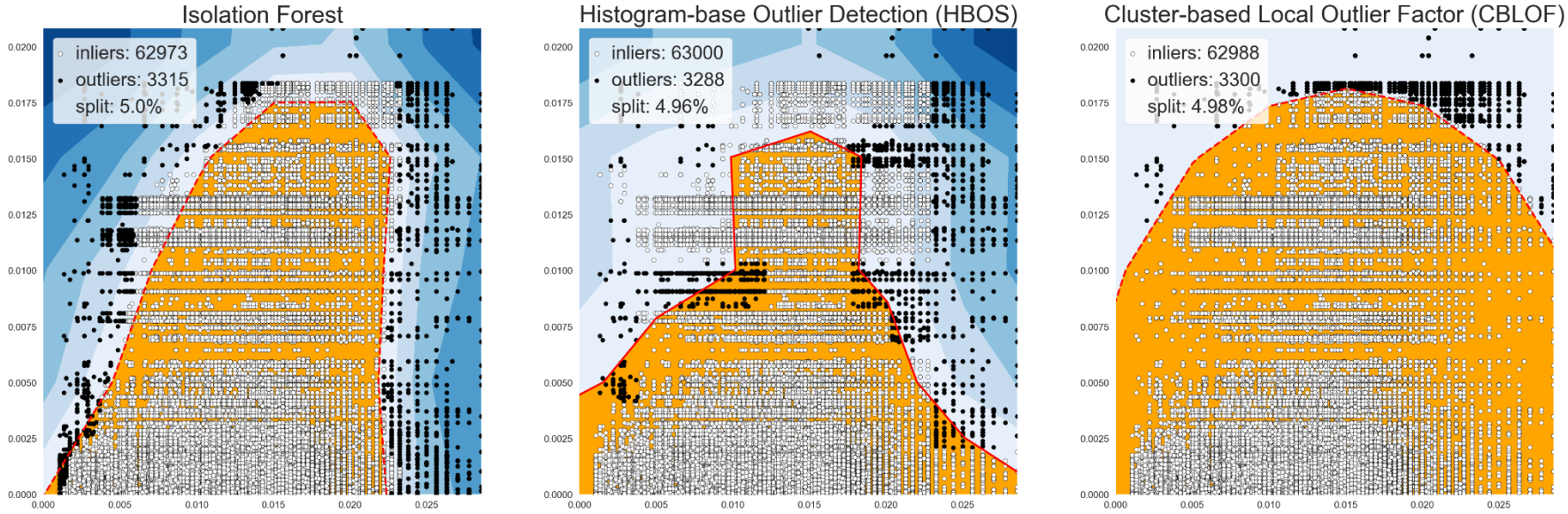

Figure 6.15 Title vs Company Code – Resource Usage Outlier Classification

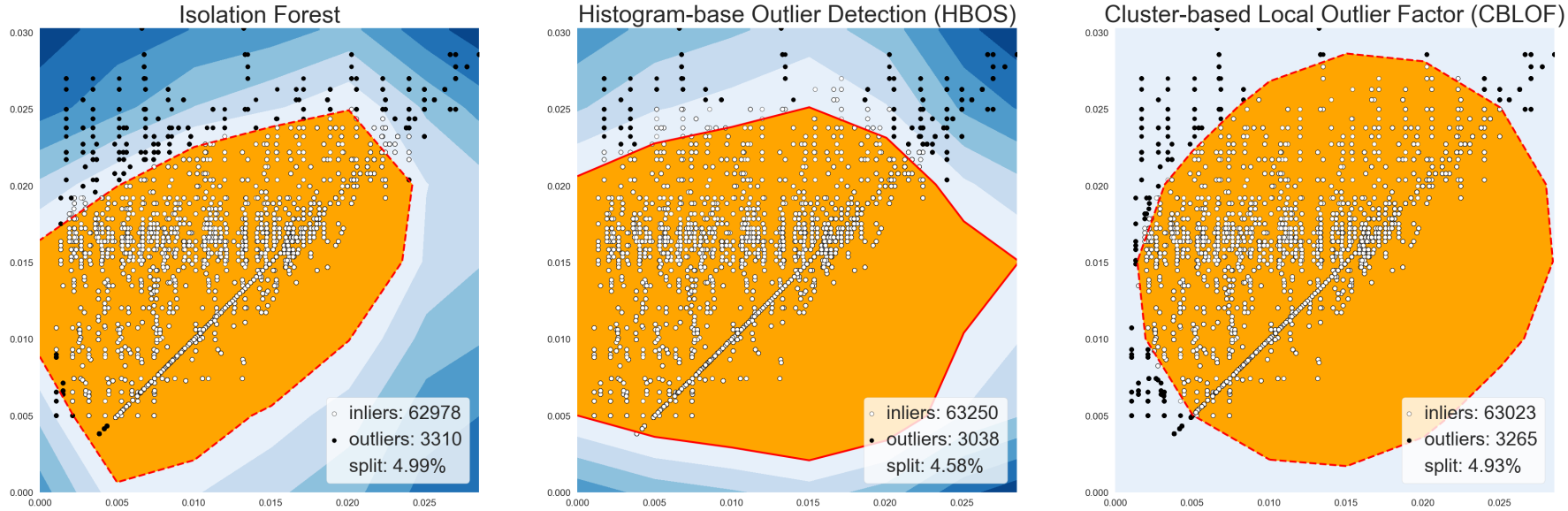

Figure 6.16 Title vs Employee ID – Resource Usage Outlier Classification

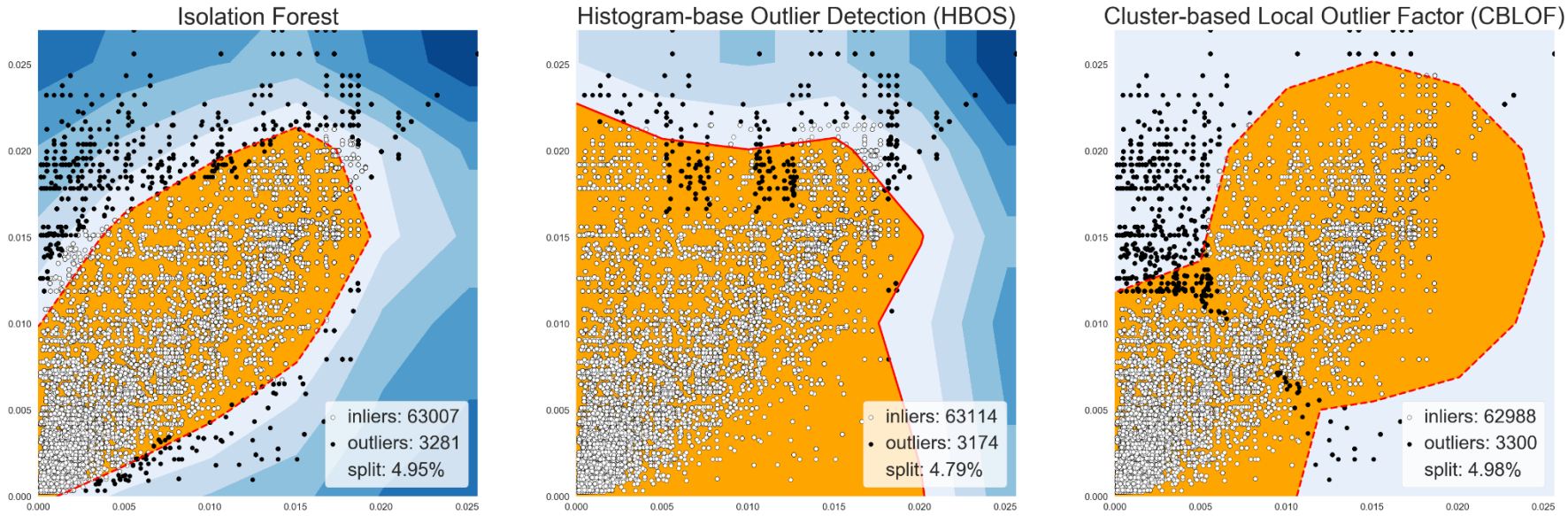

Figure 6.17 Business Unit Group vs Business – Resource Usage Outlier Classification

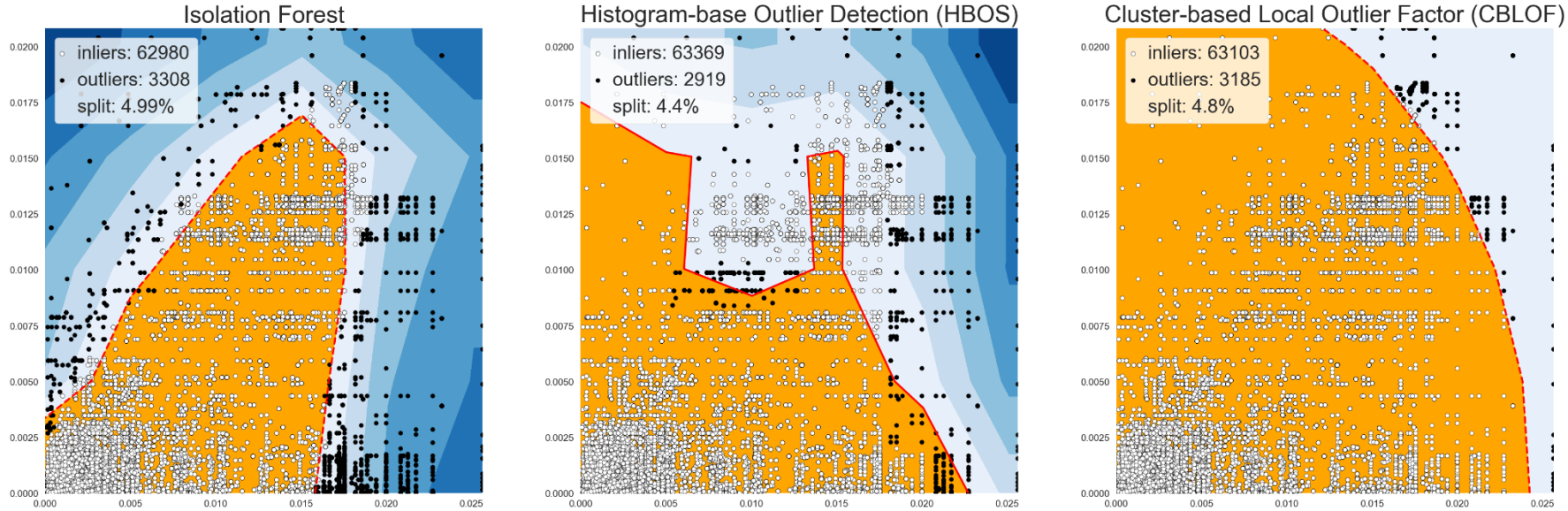

Figure 6.18 Business Unit Group vs Company Code – Resource Usage Outlier Classification

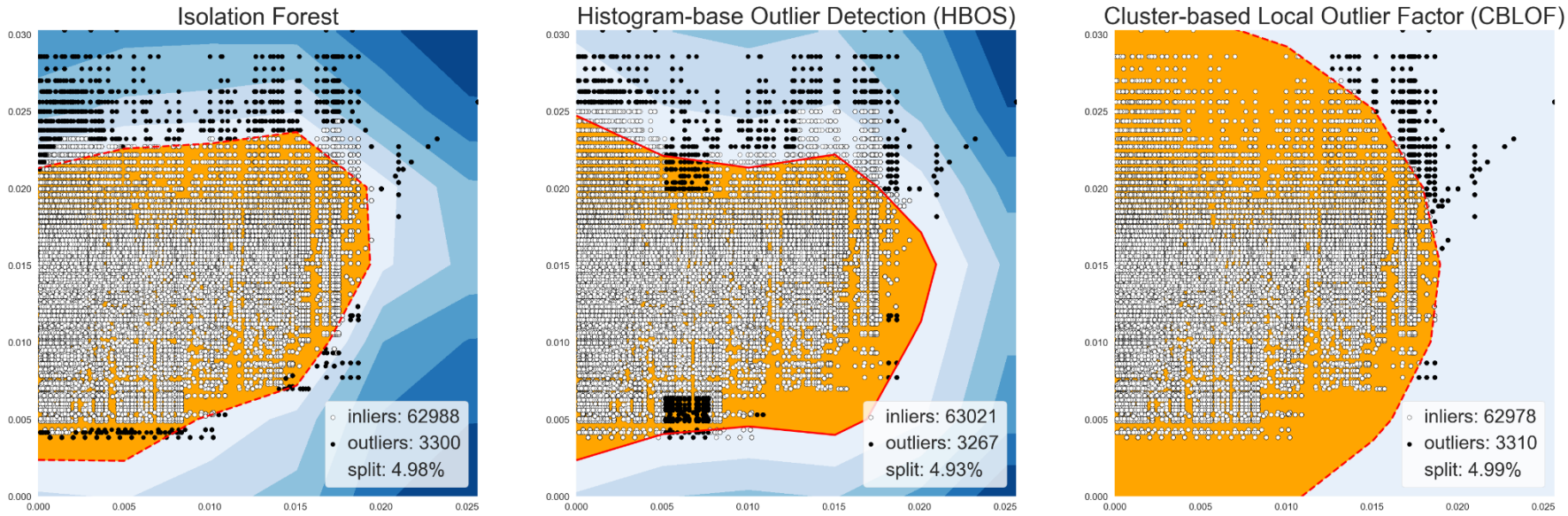

Figure 6.19 Business Unit Group vs Employee ID – Resource Usage Outlier Classification

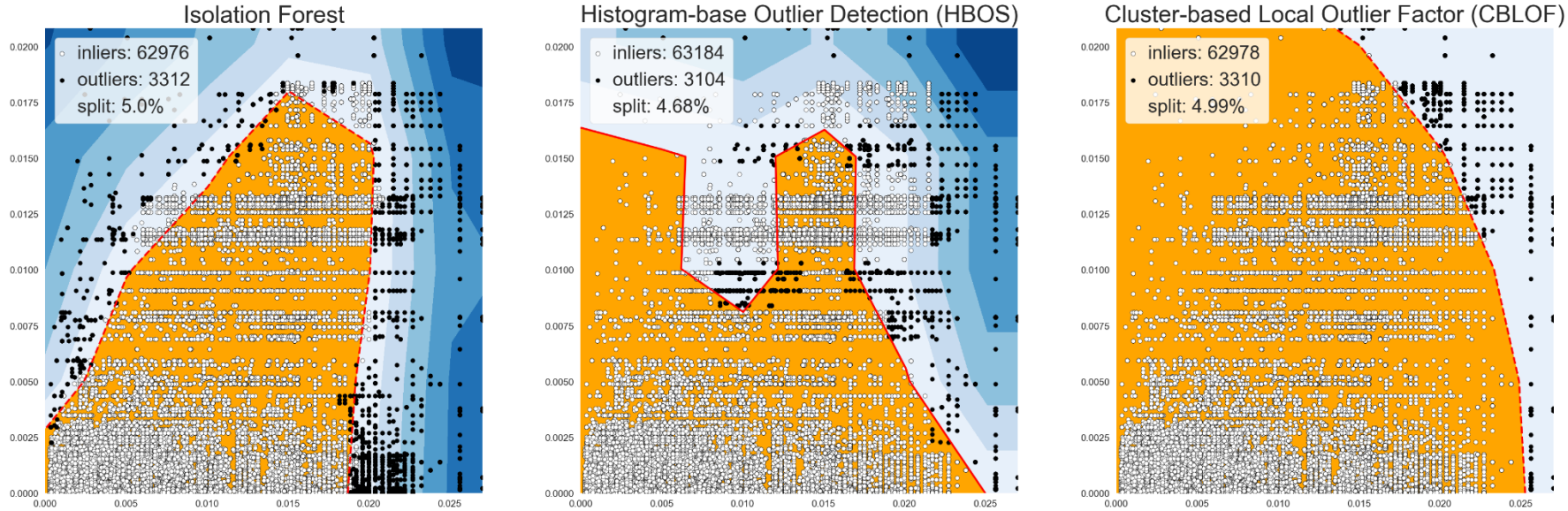

Figure 6.20 Business vs Company Code – Resource Usage Outlier Classification

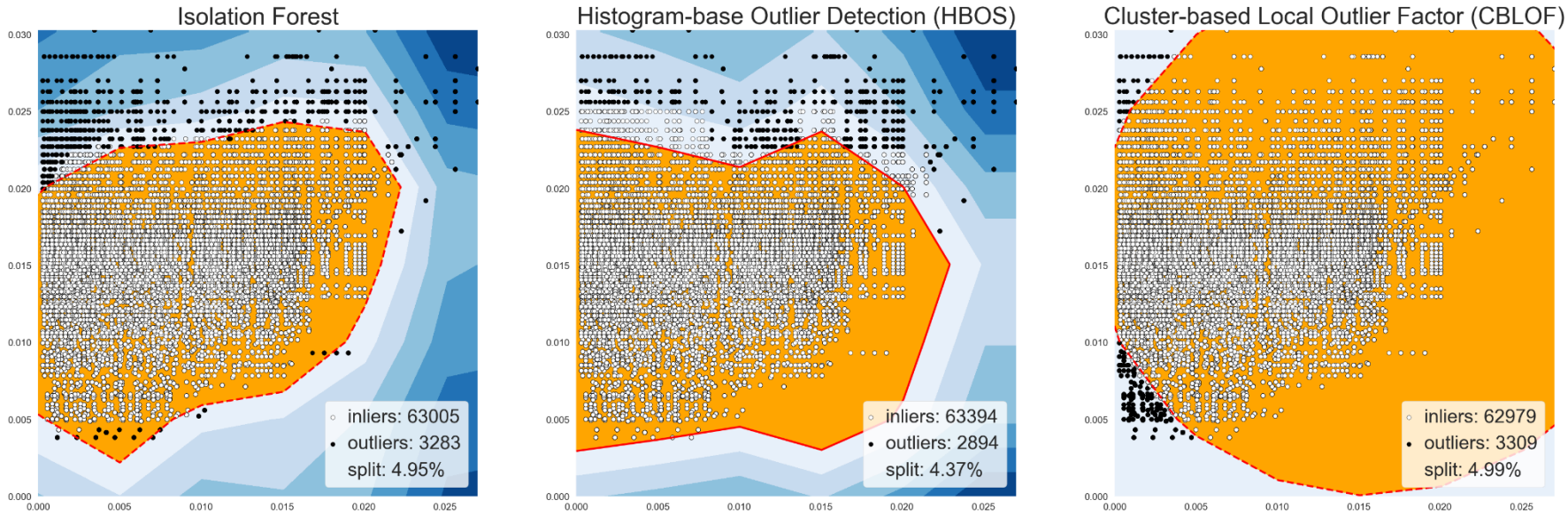

Figure 6.21 Business vs Employee ID – Resource Usage Outlier Classification

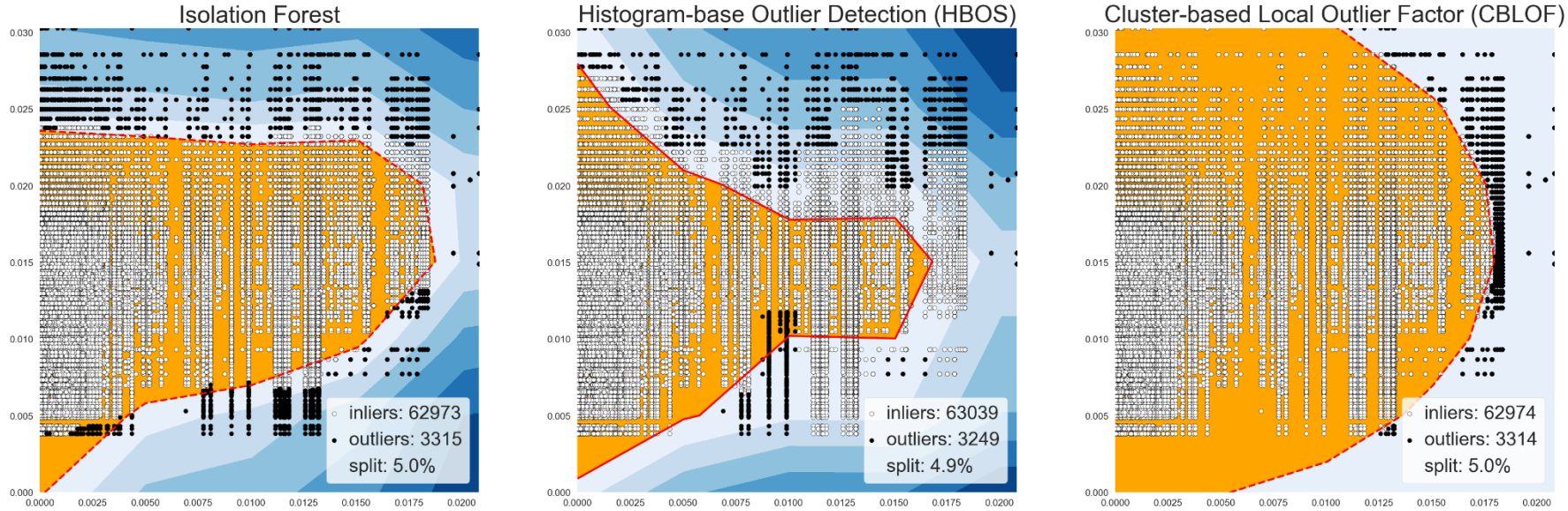

Figure 6.22 Company Code vs Employee ID – Resource Usage Outlier Classification

Appendix C: Results of Resource Usage Feature Pairing Classification

Three classifiers were applied to 21 unique pairing of the seven features in the Bank dataset. The graphs visualizing each of these classification iterations are presented in Appendix B, following the same ordering as Table 6.1 below.

Table 6.1 Feature Pairing Outlier Classification Results

#	Feature A	Feature B	Classifier	Outliers	Inliers	% Split
A1	Manager	Department	Isolation Forest	3276	63012	4.94
A2	Manager	Department	Histogram Based	2800	63488	4.22
A3	Manager	Department	Cluster Based LOF	3223	63065	4.86
B1	Manager	Title	Isolation Forest	3310	62978	4.99
B2	Manager	Title	Histogram Based	2221	64067	3.35
B3	Manager	Title	Cluster Based LOF	3309	62979	4.99
C1	Manager	Bufugu	Isolation Forest	3309	62979	4.99
C2	Manager	Bufugu	Histogram Based	3136	63152	4.73
C3	Manager	Bufugu	Cluster Based LOF	3309	62979	4.99
D1	Manager	Business	Isolation Forest	3314	62974	5
D2	Manager	Business	Histogram Based	2812	63476	4.24
D3	Manager	Business	Cluster Based LOF	3314	62974	5
E1	Manager	Ccode	Isolation Forest	3314	62974	5
E2	Manager	Ccode	Histogram Based	3282	63006	4.95
E3	Manager	Ccode	Cluster Based LOF	3308	62980	4.99
F1	Manager	DN	Isolation Forest	3263	63025	4.92
F2	Manager	DN	Histogram Based	2648	63640	3.99
F3	Manager	DN	Cluster Based LOF	3231	63057	4.87
G1	Department	Title	Isolation Forest	3306	62982	4.99
G2	Department	Title	Histogram Based	2784	63504	4.2
G3	Department	Title	Cluster Based LOF	3106	63182	4.69
H1	Department	Bufugu	Isolation Forest	3315	62973	5
H2	Department	Bufugu	Histogram Based	3226	63062	4.87
H3	Department	Bufugu	Cluster Based LOF	3241	63047	4.89
I1	Department	Business	Isolation Forest	3307	62981	4.99
I2	Department	Business	Histogram Based	1568	64720	2.37
I3	Department	Business	Cluster Based LOF	3313	62975	5
J1	Department	Ccode	Isolation Forest	3314	62974	5
J2	Department	Ccode	Histogram Based	2340	63948	3.53
J3	Department	Ccode	Cluster Based LOF	3315	62973	5

#	Feature A	Feature B	Classifier	Outliers	Inliers	% Split
K1	Department	DN	Isolation Forest	3311	62977	4.99
K2	Department	DN	Histogram Based	2714	63574	4.09
K3	Department	DN	Cluster Based LOF	3299	62989	4.98
L1	Title	Bufugu	Isolation Forest	3312	62976	5
L2	Title	Bufugu	Histogram Based	2566	63722	3.87
L3	Title	Bufugu	Cluster Based LOF	3313	62975	5
M1	Title	Business	Isolation Forest	3315	62973	5
M2	Title	Business	Histogram Based	2160	64128	3.26
M3	Title	Business	Cluster Based LOF	3234	63054	4.88
N1	Title	Ccode	Isolation Forest	3315	62973	5
N2	Title	Ccode	Histogram Based	3288	63000	4.96
N3	Title	Ccode	Cluster Based LOF	3300	62988	4.98
O1	Title	DN	Isolation Forest	3310	62978	4.99
O2	Title	DN	Histogram Based	3038	63250	4.58
O3	Title	DN	Cluster Based LOF	3265	63023	4.93
P1	Bufugu	Business	Isolation Forest	3281	63007	4.95
P2	Bufugu	Business	Histogram Based	3174	63114	4.79
P3	Bufugu	Business	Cluster Based LOF	3300	62988	4.98
Q1	Bufugu	Ccode	Isolation Forest	3308	62980	4.99
Q2	Bufugu	Ccode	Histogram Based	2919	63369	4.4
Q3	Bufugu	Ccode	Cluster Based LOF	3185	63103	4.8
R1	Bufugu	DN	Isolation Forest	3300	62988	4.98
R2	Bufugu	DN	Histogram Based	3267	63021	4.93
R3	Bufugu	DN	Cluster Based LOF	3310	62978	4.99
S1	Business	Ccode	Isolation Forest	3312	62976	5
S2	Business	Ccode	Histogram Based	3104	63184	4.68
S3	Business	Ccode	Cluster Based LOF	3310	62978	4.99
T1	Business	DN	Isolation Forest	3283	63005	4.95
T2	Business	DN	Histogram Based	2894	63394	4.37
T3	Business	DN	Cluster Based LOF	3309	62979	4.99
U1	Ccode	DN	Isolation Forest	3315	62973	5
U2	Ccode	DN	Histogram Based	3249	63039	4.9
U3	Ccode	DN	Cluster Based LOF	3314	62974	5