Telecommunication Systems
https://doi.org/10.1007/s11235-023-01085-3

ORIGINAL PAPER

Insider threat detection using supervised machine learning algorithms

Phavithra Manoharan[1] · Jiao Yin[1] · Hua Wang[1] · Yanchun Zhang[1,2,3] · Wenjie Ye[1]

Accepted: 25 November 2023

Abstract
Insider threats refer to abnormal actions taken by individuals with privileged access, compromising system data's confidentiality, integrity, and availability. They pose significant cybersecurity risks, leading to substantial losses for several organizations. Detecting insider threats is crucial due to the imbalance in their datasets. Moreover, the performance of existing works has been evaluated on various datasets and problem settings, making it challenging to compare the effectiveness of different algorithms and offer recommendations to decision-makers. Furthermore, no existing work investigates the impact of changing hyperparameters. This paper aims to objectively assess the performance of various supervised machine learning algorithms for detecting insider threats under the same setting. We precisely evaluate the performance of various supervised machine learning algorithms on a balanced dataset using the same feature extraction method. Additionally, we explore the impact of hyperparameter tuning on performance within the balanced dataset. Finally, we investigate the performance of different algorithms in the context of imbalanced datasets under various conditions. We conduct all the experiments in the publicly available CERT r4.2 dataset. The results show that supervised learning with a balanced dataset in RF obtains the best accuracy and F1-score of 95.9% compared with existing works, such as, DNN, LSTM Autoencoder and User Behavior Analysis.

Keywords Cybersecurity · Insider threat · Imbalanced dataset · Supervised learning

1 Introduction

Nowadays, insider threats present a serious concern to cybersecurity. Insider threats are more vulnerable than outsider threat attacks, and while rare, they can cause considerable damage [1–3]. In recent years, much research has focused on understanding insiders and developing methods to detect insider threats [4–7]. Insider threats can include data exfiltration, espionage and fraud, exposure of classified information, IT sabotage, and theft of intellectual property [8, 9].

A report shows that insider threats have increased by 47% over the last two years.[1] Statistics show that insider attacks occur in 70% of businesses [10]. According to the 2022 Cost of Insider Threats: Global Report, both the frequency of insider threat occurrences and the associated expenses per incident have increased, with the latter reaching $15.38 million. In a recent survey, 55% of companies reported that the most significant insider threat risk was attributed to a select group of privileged users.[2] For example, in the 2019 Capital One breach, a former Amazon employee who participated in the attack was convicted in 2022.[3]

Insider threats are frequently perpetrated by individuals with significant technical expertise and knowledge of the organization's internal networks and security protocols [11–13]. Insider threats can bypass conventional security measures and a broader range of actions than external

✉ Jiao Yin
jiao.yin@vu.edu.au

Phavithra Manoharan
phavithra.manoharan@live.vu.edu.au

Hua Wang
hua.wang@vu.edu.au

Yanchun Zhang
yanchun.zhang@vu.edu.au

Wenjie Ye
wenjie.ye@vu.edu.au

1 School of Computer Science and Technology, Zhejiang Normal University, Jinhua, Zhejiang 321004, China

2 Department of New Networks, Peng Cheng Laboratory, Shenzhen 518055, China

3 Institute for Sustainable Industries and Liveable Cities, Victoria University, Melbourne, VIC 3011, Australia

[1] https://techjury.net/blog/insider-threat-statistics/#gref.

[2] https://techjury.net/blog/insider-threat-statistics/#gref.

[3] https://www.theverge.com/2022/6/18/23173727/former-amazon-employee-convicted-over-2019-capital-one-hack-paige-thompson.

Published online: 28 December 2023

threats [14–17]. Machine learning approaches have gained widespread use in various cybersecurity domains, including insider threat detection and cyber-attack prediction [18–22]. However, assessing the effectiveness of existing approaches under different datasets and diverse settings poses certain challenges.

Despite their growing popularity, detecting and mitigating insider threats present significant challenges due to limited access to real-world insider threat datasets caused by privacy concerns [23]. Moreover, the complex interplay of activities within large organizations makes identifying insider threats challenging [24]. Furthermore, the effectiveness of machine learning models in generating predictions is compromised when dealing with imbalanced data [25, 26]. To address these problems, we compare various supervised machine learning algorithms in the same balanced dataset and settings. Additionally, we investigate the effects of changing the hyperparameters on the performance. Furthermore, we assess the effectiveness of several supervised machine learning techniques in addressing datasets with different levels of class imbalance.

In summary, this paper provides the following contributions:

1. We compare various supervised machine learning algorithms such as Random Forest (RF), XG Boost, k-Nearest Neighbor (KNN), Gaussian Naïve Bayes (GNB), Decision Tree (DT), Multi-Layer Perceptron classifier (MLP), Adaptive Boosting (AdB) and Quadratic Discriminant Analysis (QDA) in the CERT r4.2 balanced dataset.
2. We examine the impact of hyperparameter modifications on the performance of the machine learning models KNN, DT, and AdB in the balanced dataset.
3. We investigate the performance of various supervised ML methods for dealing with imbalanced dataset standards in many real-world scenarios. In particular, we examine these methods' performance in varying levels of class imbalance, such as 40%,30%,20%,10%,1%, and 0.5% of insiders.

The paper is organized as follows: Sect. 2 covers related work on insider threat detection, Sect. 3 outlines the methodology for detecting insider threats, Sect.4 describes the experimental setup and evaluation, and Sect.5 presents the conclusion.

2 Related work

This section has three subsections. The first subsection explains the classification of insiders and insider threats, while the second subsection introduces datasets for insider threat detection. The third section provides the literature view of insider threat detection.

2.1 Classification of insiders and insider threat

The following section presents a foundational understanding of insider threats by describing insiders, categorizing their various types, and detailing their potential activities.

2.1.1 Insider

The Rand Corp [27] described an insider as "someone who has access to, privilege over, or knowledge of information systems and services". They also defined a hostile insider as "motivated to deliberately damage an organization's mission" (e.g., damage, degrade, destroy). Kim et al. [28] defined an insider as "someone who has the authority to enter a company as an employee, contractor or guest, regardless of the authority of the information system". The US's Cyber and Infrastructure Security Agency (CISA)[4] defines an insider threat as "An insider might utilize their access or unique knowledge of a company to harm it. This harm may include deliberate, negligent, or unintentional actions that compromise the organization's data, workers, facilities, and other resources while compromising integrity, confidentiality, and availability."

2.1.2 Insider types

Theoretically, insiders can be classified into several categories or types based on their level of access and authority within the organization [26, 29].

1. Malicious Insider: An insider who maliciously uses legitimate credentials to steal information for their or another person's benefit.
2. Contractor: A type of insider with the privilege to sensitive systems and confidential data that can be traded.
3. Inadvertent Insider: An insider who consistently makes common mistakes and disregards the organization's security policies.
4. Disgruntled Employee: An employee in the organization who feels unappreciated or overlooked.

2.1.3 Insider threat activities

Insider threats refer to malicious activities carried out by authorized individuals, such as employees, contractors, or partners, who have access to an organization's resources

[4] https://www.cisa.gov/defining-insider-threats.

and sensitive information.[5] Based on the activity of insiders, insider threats are classified into five types:

1. Information Technology (IT) Sabotage- IT Sabotage describes the technical experts with the access and capability to launch an attack [30].
2. Intellectual Property (IP) Theft- illustrates the theft of sensitive data from the organization, such as client information or source code, which can be carried out by technical or non-technical staff.
3. Fraud - indicates data that has been modified, added to, or deleted without permission.
4. Espionage - is the hidden or unlawful act of spying on a company, person, or entity to get sensitive information [31].
5. Unintentional Insider Threat - a person has authorized access to a company's network, whether they are a current or past employee, contractor, or business partner.

2.2 Datasets

This section details the datasets categorized into five groups based on the type of insider activity: masquerader-based, traitor-based, malicious, substituted masqueraders, and authentication-based. Table 1 describes the insider threat datasets.

RUU dataset [32] consists of host-based events from 34 regular users, with the help of 14 volunteers who act as masqueraders to look for information having financial value. RUU is a masquerader-based dataset. The RUU dataset was introduced by Salem and Stolfo in 2009 and 2011.

Enron dataset [33] comprises 500,000 emails from 150 Enron Corporation employees over five years. It is a traitor-based dataset.

Schonlau dataset [34] encompasses 50 users in a substituted masquerader dataset, with each user generating 15,000 Unix shell commands. During masquerade sessions, random commands from unidentified users are introduced.

Greenberg dataset [35] provides complete Unix C shell commands from 168 users in an authentication-based dataset. In contrast to the Schonlau dataset, Greenberg's dataset contains arguments and time stamps in command instances.

TWOS dataset [36] features a variety of data, encompassing both traitors and masqueraders. The dataset incorporates behaviors from 24 users over five days, collected through a multiplayer game that simulates 12 masquerader sessions and five traitor sessions.

CERT dataset [6] is a synthetic dataset containing system logs labeled as involving insider threats. The dataset incorporates logon, email, HTTP, device, and file access details.

The reviewed literature employed diverse malicious datasets as they encompassed various scenarios.

2.3 Literature review

Insiders are often highly skilled computer professionals, well-versed in internal networks and security measures, which enables them to circumvent conventional security mechanisms and perform a broader range of actions than outside attackers. A survey report indicates that 68% of organizations believe they are moderately to highly vulnerable to insider threats.[7] Over the past decade, the detection of insider threats has attracted significant attention from researchers and organizations due to the alarming increase in insider threats, which have risen by 44% since the previous year. This heightened focus has drawn the interest of a wide range of researchers. Previous research has explored the application of machine learning algorithms to identify unusual behavior in cybersecurity actions. However, real-time datasets for insider threat detection are rare [37]. Various methods for detecting insider threats have been proposed in recent years. The following section will provide an overview of related work from recent literature.

2.3.1 Machine learning approaches

Liu et al. [38] conducted a review of insider threat-related cybersecurity issues such as malware and advanced persistent threats (APT). The survey is designed around the APT intrusion kill chain and focuses on the three main insider categories: traitor, masquerader, and unintentional perpetrator.

Jiang et al. [39] described an insider prediction method based on a user sentiment profile derived from the users' social media and email content. They applied sentiment analysis techniques to identify the emotions and attitudes expressed in the text and used machine learning algorithms to predict the motivation behind the insider threat.

One-Class Support Vector Machine (OCSVM) is one of the most commonly used semi-supervised techniques for anomaly detection due to its strong theoretical foundations [40]. Rashid et al. [41] employed HMMs to model users' weekly activity sequences and identify potential insider assaults from small variations in weekly user activities, as

[5] https://www.dhs.gov/science-and-technology/cybersecurity-insider-threat.

[6] https://kilthub.cmu.edu/articles/dataset/Insider_Threat_Test_Dataset/12841247/1.

[7] https://www.globenewswire.com/news-release/2022/01/25/2372208/35374/en///Global-Cybersecurity%20-Study-Insider-Threats-Cost-Organizations-%2015-4-Million-Annually-up-34-Percent-from-2020.html#:~:text=This%20%20year's%202022%20Cost%20of,percent%20in%20just%20two%20years.

Table 1 Datasets for insider threat detection

Dataset	Threat types	Description
RUU [32] (2009)	Masquerader	14 Masqueraders and 34 Normal users
Enron [33] (2015)	Traitor	500,000 emails from 150 employees
Schonlau [34] (2001)	Substituted Masquerader	50 users' Unix shell commands
Greenberg [35] (1988)	Authentication	50 users' Unix C shell commands
TWOS [36] (2017)	Malicious	12 masqueraders users, 5 traitor sessions and 24 users
CERT (2013)	Malicious	5 insiders and 3995 normal users

indicated by HMM probabilities (of user sequences) below a predetermined threshold.

Le et al. [42] proposed a user-centric approach based on four supervised machine learning algorithms. The paper explored the effect of different data granularity levels on the accuracy of insider threat detection using machine learning. The authors investigated the use of different feature sets at different levels of granularity, such as user, session, and activity levels, and evaluated their performance using different classifiers.

In [43], the XG Boost algorithm was employed, and audit logs were used to extract features of behavior characteristics. Gamachchi et al. [44] presented a method for analyzing heterogeneous data and separating potential harmful users based on graph learning and Isolation Forest. The CERT r4.2 dataset is used to generate the graph attributes. Then, an Isolation Forest is used to generate anomaly scores for specific users using data directly retrieved from authentication, usage records of removable media, and psychometric scores.

2.3.2 Deep learning approaches

Le et al. [45] proposed supervised and unsupervised learning algorithms to detect the insiders. They also presented an insider threat detection workflow that combines supervised and unsupervised learning methods. The author evaluated the HMM, Decision Tree (DT), and Self Organizing Maps (SOM).

In [46] developed an unsupervised deep learning model using LSTM Auto-encoder that learns to imitate individual employees' behavior based on their day-by-day time-stamped activity sequence. The authors suggest that the LAC approach can be used as a complementary tool for insider threat detection in organizations. This paper [47] discussed a deep learning approach to detect insider threats based on user behavior. It proposed a framework that leveraged deep learning algorithms, including LSTM, CNN, and autoencoder, to capture and analyze user behavior.

Tuor et al. [48] proposed a novel approach for detecting insider threats in structured cybersecurity data streams. The proposed method utilizes a deep learning-based autoencoder to learn the normal behavior patterns of users and detect anomalous patterns that could indicate insider threats. Lu et al. [49] proposed a framework called "Insider Catcher" based on the deep learning technique, LSTM model, to represent system logs structured sequence.

In [50], a deep learning model consisting of CNN and LSTM models was proposed, based on the users' behaviors and character embeddings. In [51], a methodology was designed using a graph convolutional network (GCN) for anomaly detection in insider threat detection. The authors characterized the users' behavior and the relationship between users. GCNs can effectively model such relationships in a graph data structure.

3 Methodology

This paper evaluates eight supervised ML algorithms using the balanced CERT r 4.2 dataset. We also explore various hyperparameters for the KNN, RF, and AdB algorithms in the balanced dataset. In addition, we assess the effectiveness of various supervised ML techniques in addressing imbalanced datasets. Specifically, we evaluate these methods under different degrees of class imbalance, ranging from 40 to 0.5% of insiders. Figure 1 provides an overview of our approach.

3.1 Data collection and data pre-processing

This research uses the publicly available CERT r4.2 dataset for our implementation and evaluation. The CERT r4.2 dataset, collected by Carnegie Mellon University's CERT Insider Threat Center, originates from various sources, including case files, investigations, and forensic analysis [52]. The CMU institute has proposed multiple types of datasets (r1-r6).

The CERT r4.2 dataset comprises over 20 GB of system log files from 1000 users, spanning 500 days, and includes both normal and malicious behavior. It consists of 930 normal users and 70 malicious insiders, with only 0.03% of the incidents categorized as malicious and 99.7% as normal.

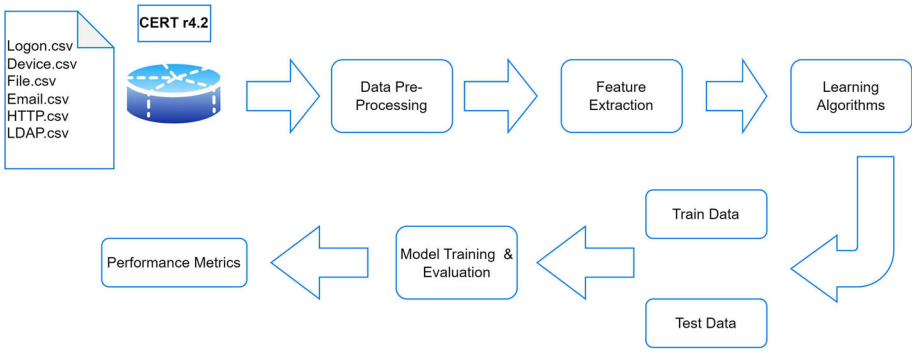


Fig. 1 Insider threat detection framework

The dataset includes seven csv files, among which are the following:

1. Device—Logs of connecting and disconnecting external devices (USB drives)
2. Logon—Users contain login and logout time users
3. Email—Email records of all employees
4. HTTP—contains users' browser logs(URLs)
5. File—user activity report (open, write, copy, delete).
6. Psychometric—includes information on user personality attributes- OCEAN (Openness, Conscientiousness, Extraversion, Agreeableness, Neuroticism).
7. LDAP (Lightweight Directory Access Protocol)- consists of user information and their job roles

In the pre-processing phase, each CSV file contains the raw data for all users, which is merged into a master file, except the psychometric.csv file. From this aggregated master file, a feature set was extracted that includes both text strings and integers. These values need to be correctly encoded to be used as input for our proposed approach. Since the dataset contains malicious internal attacks, a tag of "1" is assigned to every malicious event and "0" to normal events. Each row describes a particular event, including the user's name, role, event ID, date, PC ID, type of activity, specific operation details, and attribute details (such as sender, recipient, and email content). Event IDs have intentionally been omitted as features in the data. The data may contain null or missing values, which were pre-processed by substituting the estimated mean value of the relevant feature for any missing values.

The selected features from various CSV files contain both string and numerical values. However, our algorithm can only process numerical values. Therefore, the input values, such as Day, Time, User_ID, PC, User_Role, User_Functional_Unit, User_Department, and activity features,must be appropriately encoded for accurate predictions. A feature's presence is indicated by "1", while its absence is meant by "0". Regarding data labeling, a user-day is classified as an insider threat if the user has carried out at least one malicious activity on that day. Each day of the week is assigned a number from "0" for Monday to "6" for Sunday.

Logon activity is assigned the label "1", while logoff activity is indicated by the label "2". Similarly, device connection and disconnection events are distinguished by the labels "3" and "4", respectively. Email-related activities are represented by the label "5", and file-related activities are denoted by "6". Lastly, HTTP (URL) activity is labelled "7". Each user has a specific position within the organization. Table 2 mentions the feature values of the dataset.

3.2 Handling imbalanced datasets

The encoded data is imbalanced, and since it is a large dataset, the downsampling technique is used to address the imbalanced dataset problem. In downsampling, the number of samples is reduced by deleting some to achieve a balanced dataset for training the model. The downsampling technique is used for various levels of class imbalance, such as 40%, 30%, 20%, 10%, 1%, and 0.5% of insiders, to assess the effectiveness of different supervised machine learning techniques in handling imbalanced data using standard evaluation metrics.

3.3 Learning algorithms

3.3.1 Random forest

Random Forest (RF) is a supervised machine learning method that develops and combines various decision trees

Table 2 List of features and their possible values

Features	Values
Day	0–6
Time	1–24
User id	1–1000
Role	1–42
Functional unit	1–6
Department	1–7
PC	Unique number
Activity	1–7

to form a "forest". It can be applied to situations involving classification and regression. Each tree provides a classification or a "vote" when employing a Random Forest for classification. The forest chooses the classification with the most "votes". When performing regression with a Random Forest, the forest selects the average of all tree outputs.

3.3.2 XGBoost

Like RF, XG also uses decision trees as the classifiers in an ensemble [53]. However, XG assumes a gradient boosting approach, where optimizing a differentiable loss function directs the combining of straightforward decision trees into a potent ensemble. Furthermore, in boosting, the classifier's output predicts class labels using probabilities determined from a linear combination of each decision tree output. Other techniques can be combined with boosting to reduce overfittings, such as the random subspace approach and the random subset of training data available to each tree.

3.3.3 Decision tree

A Decision Tree (DT) is a supervised learning algorithm used for classification and regression tasks. It creates a model as a tree structure by recursively splitting the data into subsets based on the features that best separate the classes. The goal is to create a model that predicts the class of a new instance by traversing the tree from the root to a leaf node. DTs are easy to interpret and visualize and can handle both categorical and continuous data [54, 55].

3.3.4 Gaussian Naive Bayes

Gaussian Naive Bayes (GNB) is a probabilistic algorithm used in machine learning for classification tasks. It assumes that the features of a dataset are independent and normally distributed. GNB calculates the likelihood of a given class for a new data point by comparing its feature values to the means and variances of the feature values in each class. The class with the highest likelihood is assigned to the data point.

3.3.5 KNN

The k-nearest neighbors (KNN) technique is a straightforward and user-friendly supervised machine learning approach that can be applied to classification and regression tasks. KNN calculates the distances between a query and each example in the dataset, selects the K examples closest to the query, and casts votes for the most frequent label (for classification) or computes the label averages (for regression).

3.3.6 QDA

Quadratic Discriminant Analysis (QDA) is one of the types of Bayesian classifiers. QDA offers greater flexibility in modeling the covariance matrix. With a quadratic decision boundary, this classifier is generated by fitting class-conditional densities to the data and applying Bayes' rule.

3.3.7 AdB

AdB called for Adaptive Boosting, is a supervised machine-learning method employed within an ensemble framework. The most common AdB algorithm involves using a decision tree with just one level or a decision tree with a single split, often called decision stumps. While AdB excels in handling unbalanced datasets, it may encounter difficulties when dealing with noisy data.

3.3.8 MLP

The Multi-Layer Perceptron (MLP) is a neural network type composed of multiple layers of nodes. The initial layer receives input, while the final layer generates output. In the intermediate stages, one or more hidden layers aid in uncovering underlying data patterns. In each layer, all nodes are fully connected to those in the subsequent layer, with distinct weights assigned to each connection [56, 57].

4 Experiments and evaluation

This section presents the experimental settings and results. The balanced CERT r4.2 dataset was used to evaluate the performance of machine learning algorithms, including RF, XGBoost, KNN, GNB, DT, MLP, AdB, and QDA. Subsequently, the hyperparameters of KNN, DT, and XGBoost were compared to enhance performance. Lastly, the performance of different imbalanced CERT r4.2 datasets with status levels of 0.5%, 1%, 10%, 20%, 30%, and 40% was evaluated. The experiments were conducted using the Python programming language and the Sci-kit learn library on Google Colaboratory.

4.1 Performance metrics

In cybersecurity applications, detection rate (DR) and False Positive Rate (FPR) are widely used. We will evaluate the following metrics.

$$\text{True Positive Rate TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{1}$$

$$\text{False Positive Rate FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{2}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5}$$

$$F1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 * \text{TP}}{2 * \text{TP} + \text{FP} + \text{FN}} \tag{6}$$

TN and FN represent True Negative and False Negative, respectively, while TP and FP denote True Positive and False Positive. In this paper, TP signifies the number of correctly classified "malicious" samples, and FN represents the number of malicious samples incorrectly classified as "normal." TN (FP) denotes the number of correctly (or incorrectly) classified normal data samples. TPR is also known as sensitivity or recall.

4.2 Experiments on the balanced dataset

This experiment compared the performance of several supervised machine learning algorithms, including RF, XG Boost, KNN, GNB, DT, MLP, AdB, and QDA, on the balanced CERT r4.2 dataset. The dataset contains 32,770,227 events, of which 7323 are malicious instances. We assessed the performance of these classification algorithms on pre-processed data.

TTo conduct the experiments, we divided the dataset into a training dataset containing 70% of the data and a test dataset with the remaining 30%. The training dataset was used to train the machine learning models, while the test dataset was used for performance evaluation. This balanced dataset was divided into a 70:30 ratio, resulting in a training dataset comprising 10,252 samples and a test dataset comprising 4394 samples.

Precision and recall are equally important in a balanced dataset where both classes are equally represented. Consequently, in such cases, the F1 score becomes a valuable metric for assessing the overall performance of a classifier. A high F1 score indicates that the model effectively balances precision and recall, enabling accurate identification of positive and negative instances. Thus, the F1 score offers a means to compare the performance of different models when both precision and recall are crucial.

In this experiment, we used the following supervised learning classifiers and their parameters in the balanced CERT r4.2 dataset, as illustrated in Table 3. We used the overall performance of the test data in this experiment.

Table 4 presented the results of eight supervised machine learning classifiers evaluated for insider threat detection on a balanced dataset. The RF classifier achieved the highest accuracy score of 0.959, indicating that it accurately predicted 95.9% of the data points. The F1-score, which measures the balance between precision and recall, is also 0.959 for RF, suggesting high accuracy in both precision and recall. The precision score of RF is 0.9598, indicating that when it makes a positive prediction, it is correct 95.98% of the time. The recall score of RF is 0.959, indicating that it correctly identifies 95.9% of all positive instances. AdB also performs well, with an accuracy of 0.9554, a similar F1-score, and recall. Meanwhile, DT achieved both accuracy and F1-score at 0.9506. On the other hand, the GNB classifier appears to have performed the least favorably, with accuracy, F1-score, precision, and recall scores all at 0.5956. The QDA classifier also performed relatively poorly, with an accuracy of 0.6475 and a lower F1-score value of 0.6453, precision, and recall scores compared to other classifiers in Table 4. Overall, Fig. 2 shows that RF and AdB are the top performers. At the same

Table 3 Classifiers and their parameters

Classifier	Parameters
RF	n_jobs = -1, n_estimators=100, criterion='gini' max_depth=None, random_state=None
XGB	n_neighbors=5, metric='minkowski',p=2 max_depth=3, loss='log_loss', learning_rate=0.1
KNN	n_jobs=None, n_neighbors=5 p=2, metric='minkowski', algorithm='auto'
GNB	priors=None, var_smoothing=1e-09
DT	max_depth=50, max_features=None, random_state =None,max_leaf_nodes=None, criterion='gini'
MLP	random_state=1, max_iter=300, activation='relu' solver='adam', batch_size='auto'
AdB	base_estimator=DT, n_estimators=9 learning_rate=1.0, random_state=None
QDA	priors=None, reg_param=0.0

Table 4 Classification performance comparison on a balanced dataset

Classifier	Accuracy	F1-score	Precision	Recall
RF	0.959	0.959	0.9598	0.959
XGB	0.9106	0.9103	0.916	0.9106
KNN	0.8393	0.8383	0.848	0.8393
GNB	0.5956	0.5956	0.5956	0.5956
DT	0.9506	0.9506	0.9506	0.9506
MLP	0.8209	0.8204	0.8245	0.8209
AdB	0.9554	0.9554	0.9556	0.9554
QDA	0.6475	0.6453	0.6512	0.6475

time, KNN, GNB, and QDA had lower performance than the other classifiers on the CERT r4.2 balanced dataset. When comparing RF with other classifiers, it stands out with the highest accuracy and F1-score, highlighting its effectiveness in insider threat detection on a balanced dataset.

4.3 Hyperparameter impact analysis for AdB, KNN, and DT

In this section, we have demonstrated the results of the different hyperparameters for AdB, KNN, and DT on the balanced CERT r4.2 dataset. In this experiment, we used the balanced training dataset containing 10252 samples and the test dataset containing 4394 samples.

4.3.1 AdB model results

AdB is a boosting ensemble technique that turns several weak classifiers into robust classifiers. This experiment used various values of the hyperparameter 'n_estimator' ranging from 10 to 50 and the base estimator as DT on a balanced CERT insider threat dataset. The accuracy measures the proportion of correctly classified instances from the total number of instances in the dataset.

The hyperparameter results for AdB in Table 5 illustrate a substantial increase in the AdB classifier's performance as the number of estimators rises. Specifically, AdB 30 achieved the highest accuracy score at 0.9609. The F1-score values for the AdB classifier consistently remain high, with values ranging from 0.9581 to 0.9609 for various n_estimator values. AdB 40 exhibits an F1-score and recall value of 0.9602. As depicted in Fig. 3, it is evident that as the value of n_estimator increases, both F1-score and precision values decrease.

4.3.2 KNN model results

The experiment utilized a balanced CERT insider threat dataset with k n_neighbors ranging from 1 to 11 and classifier parameters set as metric = 'minkowski' and $p = 2$. Table 5 presents the performance metrics of the KNN classifier on the dataset. Initially, with $k = 1$, the accuracy stands at 0.8682. However, accuracy values decline from 0.8505 to 0.8211 as neighbors increase.

The F1-score values also decline, starting at 0.8680 for KNN1 and reaching 0.8196 for KNN11. Similarly, precision values follow a similar pattern, with KNN1 achieving the highest precision value of 0.8713 and KNN11 showing the lowest precision value of 0.8322. In contrast, recall values remain relatively consistent across all classifiers, with KNN1 recording the highest value of 0.8682 and KNN11 displaying the lowest value of 0.8211. As the value of k increases, accu-

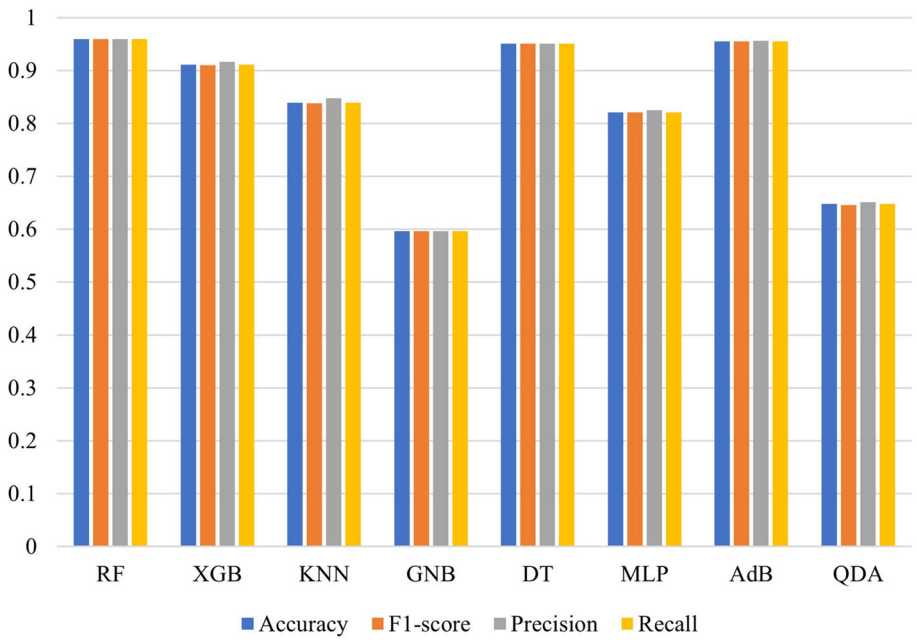


Fig. 2 Performance of supervised learning algorithms on a balanced dataset

Table 5 Performance comparison with different hyperparameters

Classifier	Accuracy	F1-score	Precision	Recall
AdB hyperparameters				
AdB10	0.9581	0.9581	0.9581	0.9581
AdB20	0.959	0.959	0.9591	0.959
AdB30	0.9609	0.9609	0.961	0.9609
AdB40	0.9602	0.9602	0.9602	0.9602
AdB50	0.9593	0.9593	0.9593	0.9593
KNN hyperparameters				
KNN1	0.8682	0.8680	0.8713	0.8682
KNN3	0.8505	0.8498	0.8566	0.8505
KNN5	0.8391	0.8381	0.8478	0.8391
KNN7	0.8336	0.8324	0.8439	0.8336
KNN9	0.8289	0.8275	0.8399	0.8289
KNN11	0.8211	0.8196	0.8322	0.8211
DT hyperparameters				
DT5	0.7745	0.7699	0.798	0.7745
DT10	0.9176	0.9173	0.9235	0.9176
DT20	0.9529	0.9529	0.9529	0.9529
DT30	0.9504	0.9504	0.9504	0.9504
DT40	0.9506	0.9506	0.9506	0.9506

racy and the F1-score, precision, and recall values decrease. The results in Fig. 4 illustrate that performance deteriorates when $k = 11$, resulting in an accuracy of 0.8211, an F1-score of 0.8196, and a recall of 0.8211.

4.3.3 DT model results

In this experiment, utilizing a balanced CERT insider threat dataset, we applied the DT classifier with a range of maximum depths from 5 to 40. The classifier was configured with parameters including max depth = 50, max features = None, random state = None, max leaf nodes = None, and criterion = 'gini'. The hyperparameters represent the decision tree's maximum depth, which determines the model's level of complexity.

Table 5 reveals that the decision tree with a maximum depth of 20 achieved the highest F1-score of 0.9529, indicating good overall performance. In contrast, the decision tree with a maximum depth of 5 had a significantly lower F1-score of 0.7699. The classifiers' accuracy scores, which range from 0.7745 for DT5 to 0.9529 for DT20 and DT30, demonstrate their effectiveness in classifying the data points.

All models have high precision, indicating that they effectively recognize positive events. As shown in Fig. 5, the DT10 model correctly predicted 92.35% of all occurrences to be positive, with a precision of 0.9235. Regarding recall, all hyperparameters perform equally well, scoring 0.9504 or 0.9506. DT20 performs well in accuracy, precision, recall, and F1 score.

4.4 Experiments on various imbalanced datasets

In this section, we examine the effectiveness of various supervised machine learning approaches for handling imbalanced datasets, which are common in many real-world scenarios. Table 6 illustrates the levels of data imbalance in the preprocessed CERT r4.2 dataset, with 30% of the data used for testing and 70% for training. The extremely unbalanced dataset with only 0.50% positive instances has 1,025,219 training samples.

4.4.1 Accuracy for various imbalanced data

Accuracy is a measure of the overall performance of a classification model, representing the proportion of correctly

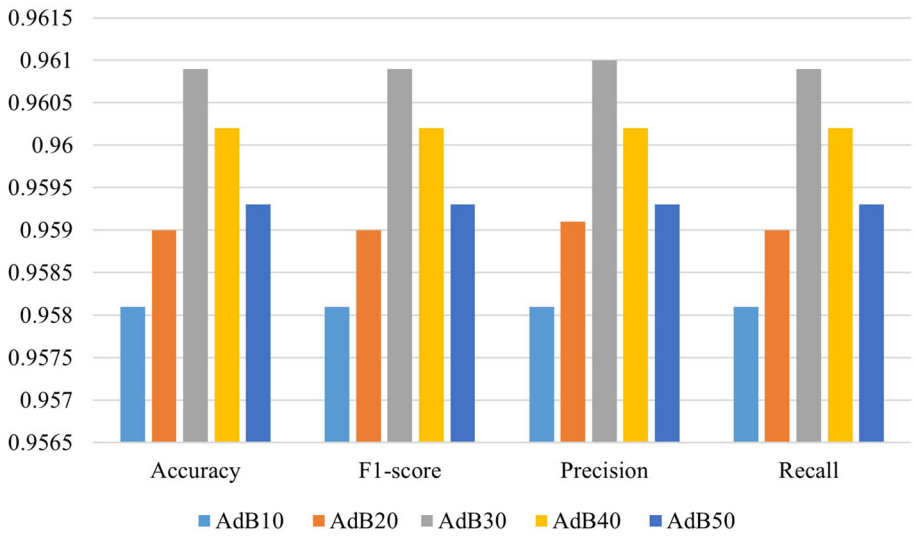


Fig. 3 Performance comparison of AdB with different hyperparameters

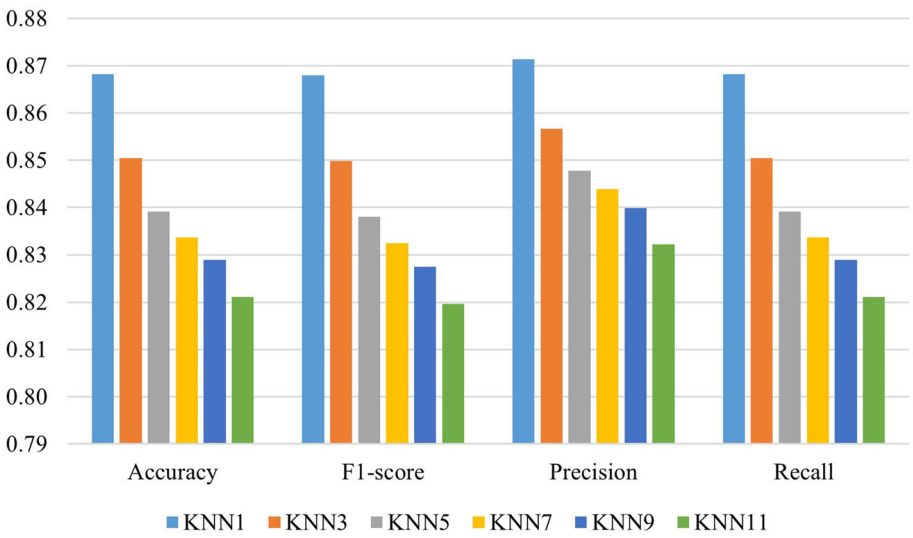


Fig. 4 Performance comparison of KNN with different hyperparameters

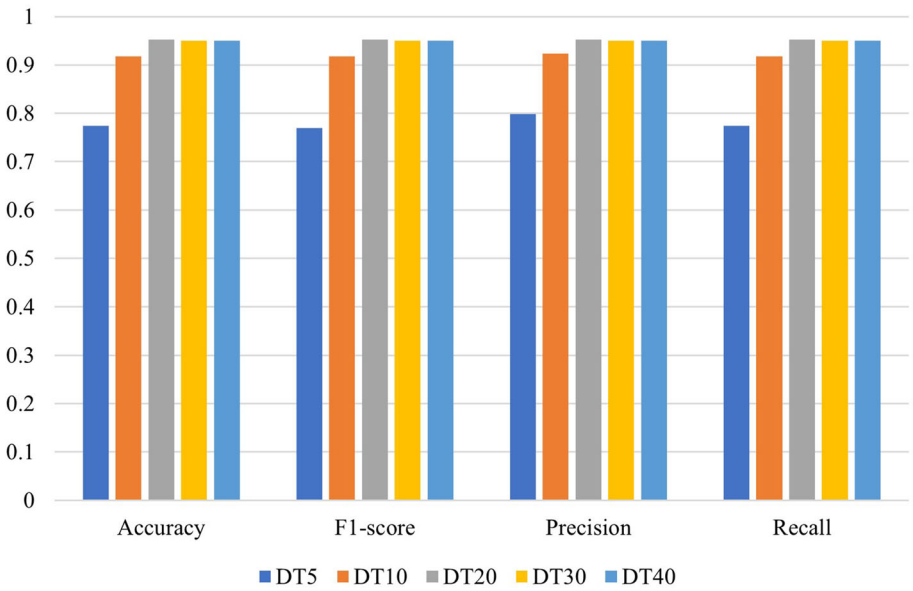


Fig. 5 Performance comparison of DT with different hyperparameters

classified instances out of the total number of instances in the dataset. The accuracy values of various machine learning methods are presented in Table 7. These values cover different data imbalance levels, from balanced datasets to highly unbalanced ones with as little as 0.50% of positive samples.

Table 7 shows that RF consistently demonstrated high accuracy, with values ranging from 0.9590 for a balanced dataset to 0.9933 for the imbalanced dataset of only 0.50% positive samples. XGB and KNN established good accuracy for moderately imbalanced datasets but struggled with highly imbalanced datasets with only 1% or 0.50% positive samples. GNB and QDA demonstrated poor accuracy for moderately to highly imbalanced datasets, with values ranging from 0.5956 to 0.8985.

For all levels of data imbalance, DT and AdB showed consistently good accuracy, with values ranging from 0.9476 to 0.9577 and 0.9504 to 0.9590, respectively. Employing moderately imbalanced datasets, MLP demonstrated great accuracy but struggled with highly imbalanced datasets. Figure 6 illustrates the accuracy for the different imbalanced datasets. The findings in the experiment show that DT, AdB, and RF are suitable for classification tasks using the CERT r4.2 imbalanced datasets.

4.4.2 F1-score for various imbalanced data

F1-score combines both precision and recall into a single metric. F1-scores range from 0 to 1, with 1 signifying perfect precision and recall and 0 signifying poor precision and recall. A high F1-score implies that the model is performing well in precision and recall, whereas a low F1-score suggests

Table 6 Sample size details of imbalanced datasets

	50%	40%	30%	20%	10%	1%	0.50%
Training data	10252	12814	17085	25630	51261	512609	1025219
Test data	4394	5493	7323	10985	21969	219691	306932

Table 7 Accuracy comparison of various algorithms on imbalanced data

Accuracy	Balanced	40%	30%	20%	10%	1%	0.50%
RF	0.9590	0.9598	0.9537	0.9496	0.9598	0.9903	0.9933
XGB	0.9106	0.9359	0.8914	0.8858	0.9188	0.9902	0.9928
KNN	0.8393	0.8529	0.8655	0.8859	0.9221	0.9874	0.9900
GNB	0.5956	0.6080	0.7038	0.8002	0.9000	0.9900	0.9928
DT	0.9506	0.9481	0.9478	0.9445	0.9577	0.9906	0.9934
MLP	0.8209	0.8358	0.8581	0.8751	0.9210	0.9900	0.9928
AdB	0.9554	0.9552	0.9504	0.9476	0.9590	0.9906	0.9934
QDA	0.6475	0.6738	0.7156	0.8025	0.8985	0.9900	0.9928

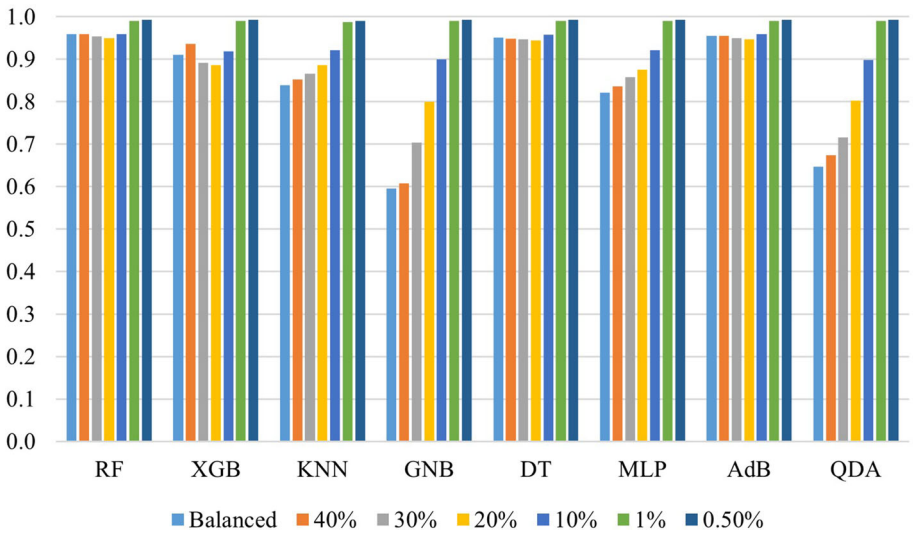


Fig. 6 Accuracy comparison of various algorithms on imbalanced data

that the model is not performing well in either precision or recall.

Table 8 presents F1 scores for a range of classifiers across a spectrum of imbalance levels, spanning from datasets with 50% to as low as 0.50% representation of the minority class. With values ranging from 0.9590 to 0.2926, the RF model has the highest F1 score across all imbalance levels. DT has F1 scores that are consistently high across all levels of imbalance, with the range of 0.9506 to 0.3010, but not as high as RF. In datasets with only 0.50% of the minority class, DT achieves the highest F1 score across all algorithms while still maintaining consistently high scores across all levels of imbalance, ranging from 0.9506 to 0.3010, although not as high as RF.

Conversely, the QDA model, with values ranging from 0.6453 to 0, has the lowest F1 scores across all imbalance levels. Figure 7 shows that the QDA model cannot perform well in precision and recall in highly imbalanced data. Similarly, KNN performs relatively better on imbalanced datasets as the level of imbalance increases. On the other hand, for XGB, the F1 score is 0.9103 on the balanced dataset but decreases to only 0.0009 for the highly imbalanced dataset. The drop in the F1 score indicates a significant reduction as the level of imbalance increases.

With the lowest F1 scores across all imbalanced levels, GNB is unsuitable for imbalanced datasets. When the level of imbalance rises, MLP's F1 scores decline, showing that it performs poorly on imbalanced datasets. Overall, the RF classifier was most effective in accurately identifying the samples, whereas the GNB and QDA classifiers performed the least effectively.

4.4.3 Precision for various imbalanced data

Precision is a measure used to assess the accuracy of a binary classification algorithm. Out of all the positive examples it

Table 8 F1 score comparison of various algorithms on imbalanced data

F1-score	Balanced	40%	30%	20%	10%	1%	0.50%
RF	0.9590	0.9507	0.9240	0.8782	0.8040	0.3654	0.2926
XGB	0.9103	0.9234	0.8031	0.6511	0.3764	0.0521	0.0009
KNN	0.8383	0.8281	0.7864	0.7221	0.6050	0.2745	0.2279
GNB	0.5956	0.1872	0.0339	0.0018	0.0000	0.0000	0.0000
DT	0.9506	0.9349	0.9129	0.8619	0.7843	0.3655	0.3010
MLP	0.8204	0.7959	0.7627	0.6729	0.4807	0.0135	0.0027
AdB	0.9554	0.9450	0.9175	0.8694	0.7933	0.3695	0.2934
QDA	0.6453	0.5203	0.2461	0.0936	0.0355	0.0000	0.0000

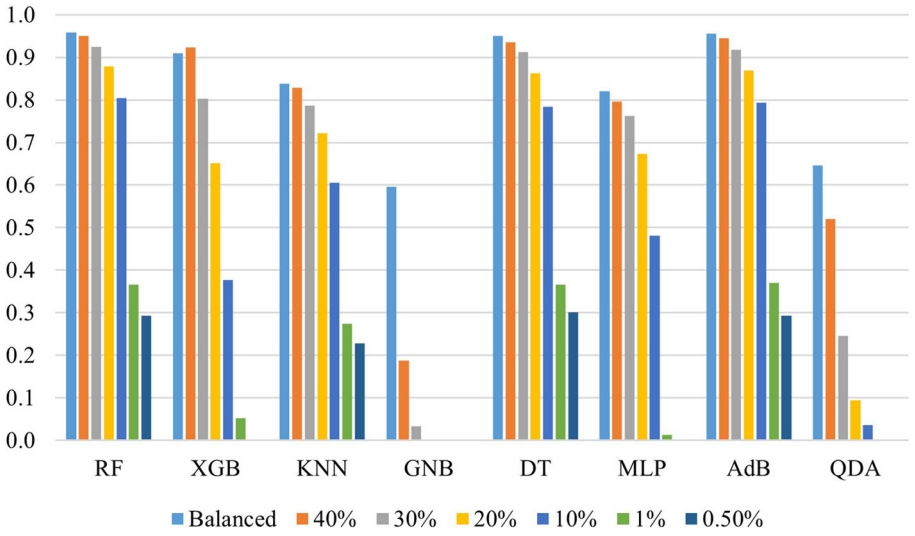


Fig. 7 F1 score comparison of various algorithms on imbalanced data

detects, precision indicates how well a classifier can identify true positive cases. Table 9 shows each classifier's precision values at different class imbalance levels, ranging from 0.5% to 40% of the minority class.

The RF and AdB models show the best precision values across all imbalance ratios, demonstrating that these models are more accurate at identifying true positives and minimizing false positives. On the other hand, the GNB classifier has deficient precision scores, particularly at high degrees of imbalance. The low precision score indicates that it is ineffective in identifying positive cases; it either correctly identified all negative cases or failed to identify any positive ones.

At the higher imbalance ratios (40% and 30%), the XGB model also performs well, with high precision values. Compared to the top-performing models, the precision values for the KNN and MLP models are considerably lower at 0.2546 and 0.3750, respectively, as shown in Fig. 8. Low precision levels show a lack of ability to recognize true positives and a propensity to classify negative events as positive mistakenly.

4.4.4 Recall for various imbalanced data

The performance of a binary classification model is measured using recall, also referred to as sensitivity or true positive rate. It measures the proportion of actual positive samples that the model correctly identifies. A high recall score indicates that the model can correctly identify most positive cases.

The accompanying Table 10 reports recall scores for each model at various thresholds, ranging from 0.5% to 50%. The RF and AdB models show significant recall scores across all thresholds, demonstrating their ability to accurately identify a significant portion of positive data in Table 10. At a 40% level of class imbalance, the RF model has a recall score of 0.9704, showing that it correctly identified 97.04% of all actual positive cases in the dataset.

However, the recall scores for QDA and GNB models are comparatively low, especially at higher thresholds, and these models may have difficulty appropriately identifying positive samples. Figure 9 shows that, at a 1% imbalance level, DT has

Table 9 Precision comparison of various algorithms on imbalanced data

Precision	Balanced	40%	30%	20%	10%	1%	0.50%
RF	0.9598	0.9318	0.9103	0.8494	0.7847	0.5327	0.5958
XGB	0.9160	0.8849	0.8810	0.8364	0.8127	0.8676	1.0000
KNN	0.8480	0.7775	0.7510	0.7042	0.6135	0.3250	0.2546
GNB	0.5956	0.5487	0.7917	1.0000	0.0000	0.0000	0.0000
DT	0.9506	0.9381	0.9150	0.8576	0.7995	0.5602	0.6181
MLP	0.8245	0.7913	0.7654	0.7066	0.7019	0.5172	0.3750
AdB	0.9556	0.9288	0.9164	0.8662	0.8006	0.5558	0.6166
QDA	0.6512	0.6316	0.6007	0.5685	0.3596	0.0000	0.0000

Fig. 8 Precision comparison of various algorithms on imbalanced data

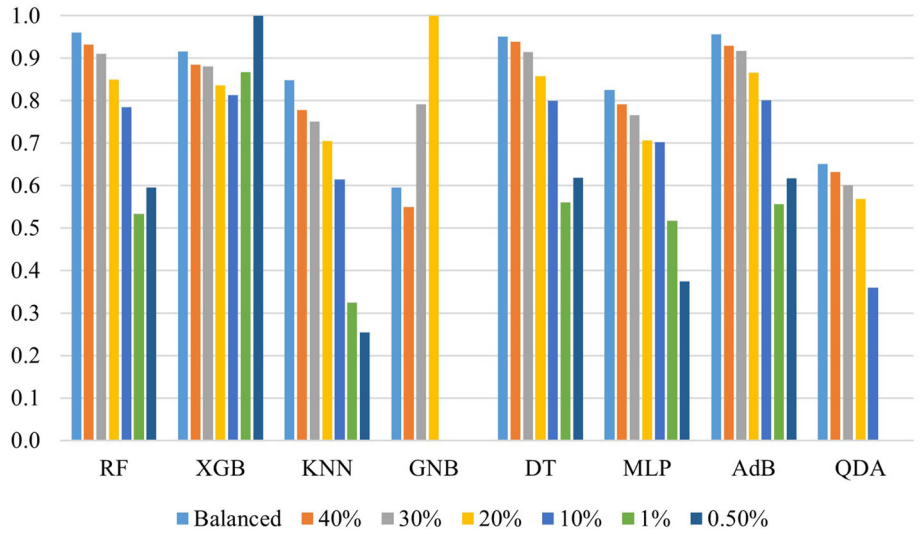


Table 10 Recall comparison of various algorithms on imbalanced data

Recall	Balanced	40%	30%	20%	10%	1%	0.50%
RF	0.9590	0.9704	0.9381	0.9090	0.8243	0.2781	0.1939
XGB	0.9106	0.9654	0.7378	0.5330	0.2449	0.0269	0.0005
KNN	0.8393	0.8858	0.8252	0.7410	0.5967	0.2376	0.2062
GNB	0.5956	0.1129	0.0173	0.0009	0.0000	0.0000	0.0000
DT	0.9506	0.9317	0.9108	0.8662	0.7697	0.2713	0.1989
MLP	0.8209	0.8006	0.7601	0.6422	0.3655	0.0068	0.0014
AdB	0.9554	0.9618	0.9185	0.8726	0.7861	0.2767	0.1925
QDA	0.6475	0.4424	0.1548	0.0510	0.0187	0.0000	0.0000

a recall score of 0.2713, whereas MLP has a recall score of 0.0068. As the level of class imbalance increases, the recall scores generally decrease for all supervised models, as it becomes more challenging to recognize the relatively few positive instances from the vast majority of negative cases.

4.5 Comparison with Existing work

In this Table 11, we have presented a comparison of existing work with the results of the proposed work using performance evaluation metrics, including accuracy, precision, True Negative Rate, Area Under Curve, and False Positive Rate. These results are then compared with those of four established approaches: DNN [58], OCSVM based on DBN [59], LSTM Autoencoder [60], and User Behavior Analysis [61]. The results demonstrate that supervised learning with a balanced dataset in RF achieves the highest accuracy and F1-score of 95.9% compared to the existing works.

Furthermore, it's important to note that, to the best of our knowledge, no existing work has conducted a comprehensive analysis of the impact of hyperparameters on the performance of AdB, KNN, and DT algorithms. Additionally, no extensive research currently investigates the effects of imbalanced datasets with varying class distribution percentages, includ-

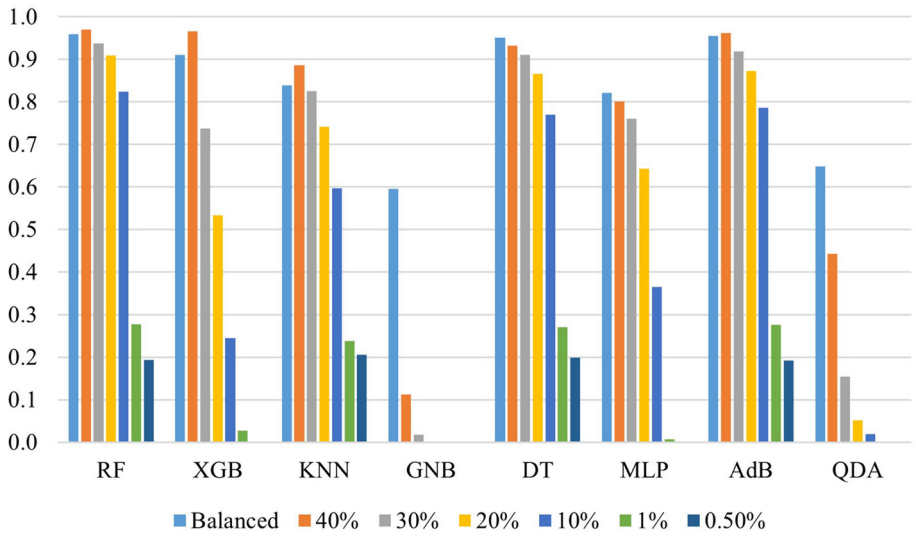


Fig. 9 Recall comparison of various algorithms on imbalanced data

Table 11 Comparison with existing work

Approach	M	DV	A	P	R	F1score	TNR	AUC	FPR
DNN [58]	S	4.2	N/A	N/A	N/A	N/A	N/A	0.944	N/A
OCSVM based on DBN [59]	U	4.2	87.79	N/A	81.04	N/A	N/A	N/A	12.18
LSTM Autoencoder [60]	U	4.2	90.17	N/A	91.03	N/A	90.15	N/A	9.84
User Behaviour Analysis [61]	U	4.2	87.3	84.9	81.7	81.9	N/A	0.89	N/A
Our Approach	S	4.2	95.9	95.98	95.9	95.9	N/A	N/A	N/A

S Supervised, *U* Unsupervised, *M* Method, *DV* Dataset Version, *A* Accuracy, *P* Precision, *R* Recall, *TNR* True Negative Rate, *AUC* Area Under Curve, *FPR* False Positive Rate

ing 40%, 30%, 20%, 10%, 1%, and 0.5%, on the performance of diverse supervised machine learning algorithms.

5 Conclusion and future work

This paper has offered a concise overview of insider threats and their significance in cybersecurity. It delved into extensive experiments involving various supervised machine learning algorithms within the CERT r4.2 balanced dataset. Moreover, when applied to the balanced dataset, the study explored the influence of hyperparameter adjustments on the performance of key machine learning models, specifically KNN, DT, and AdB. Additionally, the research probed the efficacy of diverse supervised machine learning techniques in handling imbalanced datasets, a common challenge in real-world applications. The results reveal that our proposed approach achieves the highest accuracy at 95.9, outperforming other existing works. XG Boost and AdB also performed well, while KNN and MLP had comparatively lower recall scores. GNB and QDA performed poorly, suggesting they may not be appropriate for the highly imbalanced CERT r4.2 insider threat dataset. While investigating the performance of the supervised ML methods in varying levels of class imbalance, such as balanced, 40%, 30%, 20%, 10%, 1%, and 0.5% of insiders showed that all algorithms' performance decreased as the class imbalance level increased. However, the Random Forest algorithm still performed the best in all scenarios. In future work, we will develop the algorithm to boot the minority class performance for the highly imbalanced dataset.

Author Contributions All authors made contributions to the study conception and design. Material preparation, data collection, and analysis were performed by PM and JY. HW and YZ provided resources. Investigation and validation were conducted by JY and WY. JY, HW, YZ, and WY provided supervision. The initial draft of the manuscript was written by Phavithra Manoharan, and all authors provided feedback on earlier versions of the manuscript. All authors have read and approved the final manuscript.

Funding The authors have not disclosed any funding.

Declarations

Conflict of interest The authors declare no competing interests.

References

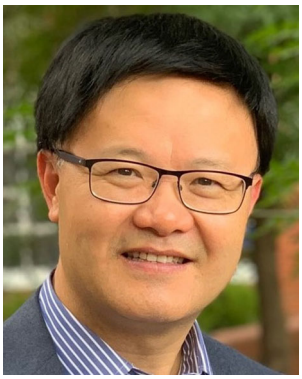
1. Hong, W., Yin, J., You, M., Wang, H., Cao, J., Li, J., & Liu, M. (2022). Graph intelligence enhanced bi-channel insider threat detection. In: *Network and system security: 16th International con-*

ference, NSS 2022, Denarau Island, Fiji, December 9–12, 2022, proceedings (pp. 86–102). Springer.
2. Feng, X., Zhu, X., Han, Q.-L., Zhou, W., Wen, S., & Xiang, Y. (2022). Detecting vulnerability on iot device firmware: A survey. *IEEE/CAA Journal of Automatica Sinica, 10*(1), 25–41.
3. Zhang, J., Pan, L., Han, Q.-L., Chen, C., Wen, S., & Xiang, Y. (2021). Deep learning based attack detection for cyber-physical system cybersecurity: A survey. *IEEE/CAA Journal of Automatica Sinica, 9*(3), 377–391.
4. Sun, N., Zhang, J., Rimba, P., Gao, S., Zhang, L. Y., & Xiang, Y. (2018). Data-driven cybersecurity incident prediction: A survey. *IEEE Communications Surveys and Tutorials, 21*(2), 1744–1772.
5. Lin, G., Wen, S., Han, Q.-L., Zhang, J., & Xiang, Y. (2020). Software vulnerability detection using deep neural networks: a survey. *Proceedings of the IEEE, 108*(10), 1825–1848.
6. Chen, X., Li, C., Wang, D., Wen, S., Zhang, J., Nepal, S., Xiang, Y., & Ren, K. (2019). Android hiv: A study of repackaging malware for evading machine-learning detection. *IEEE Transactions on Information Forensics and Security, 15*, 987–1001.
7. Qiu, J., Zhang, J., Luo, W., Pan, L., Nepal, S., & Xiang, Y. (2020). A survey of android malware detection with deep neural models. *ACM Computing Surveys (CSUR), 53*(6), 1–36.
8. Le, D. C., Zincir-Heywood, A. N., & Heywood, M. I. (2019). Dynamic insider threat detection based on adaptable genetic programming. In *2019 IEEE symposium series on computational Intelligence (SSCI)* (pp. 2579–2586). IEEE.
9. You, M., Yin, J., Wang, H., Cao, J., Wang, K., Miao, Y., & Bertino, E. (2022). A knowledge graph empowered online learning framework for access control decision-making. *World Wide Web,* 1–22.
10. Yin, J., Tang, M., Cao, J., You, M., Wang, H., & Alazab, M. (2022). Knowledge-driven cybersecurity intelligence: Software vulnerability co-exploitation behaviour discovery. *IEEE Transactions on Industrial Informatics.*
11. Huang, T., Gong, Y.-J., Kwong, S., Wang, H., & Zhang, J. (2019). A niching memetic algorithm for multi-solution traveling salesman problem. *IEEE Transactions on Evolutionary Computation, 24*(3), 508–522.
12. Wang, H., & Sun, L. (2010). Trust-involved access control in collaborative open social networks. In *2010 Fourth international conference on network and system security* (pp. 239–246). IEEE.
13. Rao, R. S., Umarekar, A., Pais, & A. R. (2022). Application of word embedding and machine learning in detecting phishing websites. *Telecommunication Systems,* 1–13.
14. Hu, X., Ma, W., Chen, C., Wen, S., Zhang, J., Xiang, Y., & Fei, G. (2022). Event detection in online social network: Methodologies, state-of-art, and evolution. *Computer Science Review, 46*, 100500.
15. Zhu, X., Wen, S., Camtepe, S., & Xiang, Y. (2022). Fuzzing: A survey for roadmap. *ACM Computing Surveys (CSUR), 54*(11s), 1–36.
16. Kavusi, H., Maghooli, K., & Haghipour, S. (2023). A novel and smarter model to authenticate and identify people intelligently for security purposes. *Telecommunication Systems, 82*(1), 27–43.
17. Humayun, M., Jhanjhi, N., Almufareh, M. F., & Khalil, M. I. (2022). Security threat and vulnerability assessment and measurement in secure software development. *Computers, Materials and Continua, 71*, 5039–5059.
18. Sun, X., Wang, H., & Li, J. (2009). Injecting purpose and trust into data anonymisation. In *Proceedings of the 18th ACM conference on information and knowledge management* (pp. 1541–1544).
19. Wang, H., Yi, X., Bertino, E., & Sun, L. (2016). Protecting outsourced data in cloud computing through access management. *Concurrency and Computation: Practice and Experience, 28*(3), 600–615.
20. Wang, H., Wang, Y., Taleb, T., & Jiang, X. (2020). Special issue on security and privacy in network computing. *World Wide Web, 23*, 951–957.
21. Phruksahiran, N. (2023). Improvement of source localization via cellular network using machine learning approach. *Telecommunication Systems,* 1–9.
22. Kok, S., Abdullah, A., Jhanjhi, N., & Supramaniam, M. (2019). Ransomware, threat and detection techniques: A review. *International Journal of Computer Science and Network Security, 19*(2), 136.
23. Sun, X., Wang, H., Li, J., & Zhang, Y. (2012). Satisfying privacy requirements before data anonymization. *The Computer Journal, 55*(4), 422–437.
24. Wang, H., Zhang, Y., Cao, J., & Varadharajan, V. (2003). Achieving secure and flexible m-services through tickets. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 33*(6), 697–708.
25. You, M., Yin, J., Wang, H., Cao, J., Miao, Y. (2021). A minority class boosted framework for adaptive access control decision-making. In *International conference on web information systems engineering* (pp. 143–157). Springer.
26. Sun, X., Li, M., Wang, H., & Plank, A. (2008). An efficient hash-based algorithm for minimal k-anonymity. In *Conferences in research and practice in information technology (CRPIT)* (Vol. 74, pp. 101–107).
27. Brackney, R. C., & Anderson, R. H. (2004). Understanding the insider threat. In *Proceedings of a march 2004 workshop*. Technical report, RAND CORP SANTA MONICA CA.
28. Kim, A., Oh, J., Ryu, J., & Lee, K. (2020). A review of insider threat detection approaches with iot perspective. *IEEE Access, 8*, 78847–78867.
29. Sun, X., Li, M., & Wang, H. (2011). A family of enhanced (l, α)-diversity models for privacy preserving data publishing. *Future Generation Computer Systems, 27*(3), 348–356.
30. Kabir, M. E., & Wang, H. (2009). Conditional purpose based access control model for privacy protection. In *Proceedings of the twentieth australasian conference on australasian database* (Vol. 92, pp. 135–142).
31. Wang, H., Cao, J., & Zhang, Y. (2020). Building access control policy model for privacy preserving and testing policy conflicting problems. *Access Control Management in Cloud Environments,* 225–247.
32. Salem, M. B., & Stolfo, S. J. (2011). Modeling user search behavior for masquerade detection. In *International workshop on recent advances in intrusion detection* (pp. 181–200). Springer.
33. Salem, M. B., & Stolfo, S. J. (2011). Modeling user search behavior for masquerade detection. In *International workshop on recent advances in intrusion detection* (pp. 181–200). Springer.
34. Schonlau, M., DuMouchel, W., Ju, W.-H., Karr, A. F., Theus, M., & Vardi, Y. (2001). Computer intrusion: Detecting masquerades. *Statistical Science,* 58–74.
35. Greenberg, S. (1988). Using UNIX: Collected traces of 168 users. PRISM. https://doi.org/10.11575/PRISM/10182, https://prism.ucalgary.ca/handle/1880/45929
36. Harilal, A., Toffalini, F., Castellanos, J., Guarnizo, J., Homoliak, I., & Ochoa, M. (2017). Twos: A dataset of malicious insider threat behavior based on a gamified competition. In *Proceedings of the 2017 international workshop on managing insider security threats* (pp. 45–56).
37. Yin, J., Tang, M., Cao, J., You, M., & Wang, H. (2022). Cybersecurity applications in software: Data-driven software vulnerability assessment and management. In *Emerging trends in cybersecurity applications* (pp. 371–389). Springer.
38. Liu, L., De Vel, O., Han, Q.-L., Zhang, J., & Xiang, Y. (2018). Detecting and preventing cyber insider threats: A survey. *IEEE Communications Surveys and Tutorials, 20*(2), 1397–1417. https://doi.org/10.1109/COMST.2018.2800740
39. Jiang, J., Chen, J., Choo, K.-K. R., Liu, K., Liu, C., Yu, M., & Mohapatra, P. (2018). Prediction and detection of malicious insid-

ers' motivation based on sentiment profile on webpages and emails. In *MILCOM 2018-2018 IEEE military communications conference (MILCOM)* (pp. 1–6). IEEE.
40. Le, T., Phung, D., Nguyen, K., & Venkatesh, S. (2015). Fast one-class support vector machine for novelty detection. In *Pacific-Asia conference on knowledge discovery and data mining* (pp. 189–200). Springer.
41. Rashid, T., Agrafiotis, I., & Nurse, J. R. (2016). A new take on detecting insider threats: Exploring the use of hidden Markov models. In *Proceedings of the 8th ACM CCS international workshop on managing insider security threats* (pp. 47–56).
42. Le, D. C., Zincir-Heywood, N., & Heywood, M. I. (2020). Analyzing data granularity levels for insider threat detection using machine learning. *IEEE Transactions on Network and Service Management, 17*(1), 30–44.
43. Gamachchi, A., & Boztas, S. (2017). Insider threat detection through attributed graph clustering. In *2017 IEEE Trustcom/BigDataSE/ICESS* (pp. 112–119). IEEE.
44. Gamachchi, A., Sun, L., & Boztas, S. (2018). A graph based framework for malicious insider threat detection. arXiv:1809.00141
45. Le, D. C., & Zincir-Heywood, A. N. (2018). Evaluating insider threat detection workflow using supervised and unsupervised learning. In *2018 IEEE security and privacy workshops (SPW)* (pp. 270–275). IEEE.
46. Paul, S., & Mishra, S. (2020). Lac: Lstm autoencoder with community for insider threat detection. In *2020 the 4th International conference on big data research (ICBDR'20)* (pp. 71–77).
47. Nasir, R., Afzal, M., Latif, R., & Iqbal, W. (2021). Behavioral based insider threat detection using deep learning. *IEEE Access, 9*, 143266–143274. https://doi.org/10.1109/ACCESS.2021.3118297
48. Tuor, A., Kaplan, S., Hutchinson, B., Nichols, N., & Robinson, S. (2017). Deep learning for unsupervised insider threat detection in structured cybersecurity data streams. arXiv:1710.00811
49. Lu, J., & Wong, R. K. (2019). Insider threat detection with long short-term memory. In: *Proceedings of the australasian computer science week multiconference* (pp. 1–10).
50. Saaudi, A., Al-Ibadi, Z., Tong, Y., & Farkas, C. (2018). Insider threats detection using cnn-lstm model. In *2018 International conference on computational science and computational intelligence (CSCI)* (pp. 94–99). IEEE.
51. Jiang, J., Chen, J., Gu, T., Choo, K.-K. R., Liu, C., Yu, M., Huang, W., & Mohapatra, P. (2019). Anomaly detection with graph convolutional networks for insider threat and fraud detection. In *MILCOM 2019–2019 IEEE military communications conference (MILCOM)* (pp. 109–114). IEEE.
52. Glasser, J., & Lindauer, B. (2013). Bridging the gap: A pragmatic approach to generating insider threat data. In *2013 IEEE security and privacy workshops* (pp. 98–104). IEEE.
53. Chen, T., & Guestrin, C. (2016) Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd Acm Sigkdd international conference on knowledge discovery and data mining* (pp. 785–794).
54. Wang, Y., Shen, Y., Wang, H., Cao, J., & Jiang, X. (2016). Mtmr: Ensuring mapreduce computation integrity with merkle tree-based verifications. *IEEE Transactions on Big Data, 4*(3), 418–431.
55. Zhang, F., Wang, Y., Liu, S., & Wang, H. (2020). Decision-based evasion attacks on tree ensemble classifiers. *World Wide Web, 23*, 2957–2977.
56. Yin, J., You, M., Cao, J., Wang, H., Tang, M., & Ge, Y.-F. (2020). Data-driven hierarchical neural network modeling for high-pressure feedwater heater group. In *Australasian database conference* (pp. 225–233). Springer.
57. Chen, Y., Han, S., Chen, G., Yin, J., Wang, K. N., & Cao, J. (2023). A deep reinforcement learning-based wireless body area network offloading optimization strategy for healthcare services. *Health Information Science and Systems, 11*(1), 8.
58. Yuan, F., Cao, Y., Shang, Y., Liu, Y., Tan, J., & Fang, B. (2018). Insider threat detection with deep neural network. In *Computational science–ICCS 2018: 18th international conference, Wuxi, China, June 11–13, 2018, proceedings, Part I 18* (pp. 43–54). Springer.
59. Lin, L., Zhong, S., Jia, C., & Chen, K. (2017). Insider threat detection based on deep belief network feature representation. In *2017 International conference on green informatics (ICGI)* (pp. 54–59). IEEE.
60. Sharma, B., Pokharel, P., & Joshi, B. (2020). User behavior analytics for anomaly detection using lstm autoencoder-insider threat detection. In *Proceedings of the 11th international conference on advances in information technology* (pp. 1–9).
61. Singh, M., Mehtre, B., & Sangeetha, S. (2020). Insider threat detection based on user behaviour analysis. In *Machine learning, image processing, network security and data sciences: Second international conference, MIND 2020, Silchar, India, July 30–31, 2020, proceedings, Part II 2* (pp. 559–574). Springer.

Phavithra Manoharan is currently pursuing a Ph.D. degree with the Institute for Sustainable Industries and Liveable Cities at Victoria University, Melbourne, VIC, Australia. She achieved a Master of Engineering degree in communication Systems from Anna University, India. Her research interests include cybersecurity, data security, privacy and data analysis.

Jiao Yin received her Ph.D. degree in computer science from the Department of Computer Science and Information Technology, La Trobe University, Melbourne, VIC, Australia, in 2022. She is currently a Research Fellow with the Institute for Sustainable Industries and Liveable Cities at Victoria University, Melbourne, VIC, Australia. Her research interests include deep learning applications and cybersecurity, focusing on graph-based learning, insider threat detection, vulnerability assessment and access control strategies.



Hua Wang received his Ph.D. degree from the University of Southern Queensland, Australia. He is now a full-time professor at Victoria University. He was a professor at the University of Southern Queensland before he joined Victoria University. He has more than ten years of teaching and working experience in applied informatics at both enterprises and universities. He has expertise in electronic commerce, business process modeling, and enterprise architecture. He is a senior member of IEEE. As a chief investigator, six Australian Research Council Discovery grants have been awarded since 2006, and 350 peer-reviewed scholarly papers have been published. 16 Ph.D. students have already graduated under his principal supervision.

Yanchun Zhang is currently a distinguished professor at Zhejiang Normal University, China and an Emeritus Professor at Victoria University, Australia. His research interests include databases, data mining, social networking, web services and e-health. He has published over 400 research papers in international journals and conference proceedings, authored/co-authored five monographs, and edited a dozen books in related areas. Dr. Zhang is a founding editor and editor-in-chief of the World Wide Web Journal (Springer) and Health Information Science and Systems Journal (Springer).

Wenjie Ye received his B.Eng. degree in computer engineering from Nanyang Technological University, Singapore, in 2007 and his Ph.D. degree from Swinburne University of Technology, Australia, in 2017. He is currently a lecturer in Information Technology at Victoria University, Australia. His research interests include sliding mode control, robotics, neural networks, machine comprehension, computer vision, nonlinear systems, and vehicle control.