

Received 7 April 2023, accepted 27 April 2023, date of publication 8 May 2023, date of current version 17 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3273895

RESEARCH ARTICLE

Privilege Escalation Attack Detection and Mitigation in Cloud Using Machine Learning

MUHAMMAD MEHMOOD¹, RASHID AMIN^{1,2},
MUHANA MAGBOUL ALI MUSLAM³, (Member, IEEE),
JIANG XIE⁴, (Fellow, IEEE), AND HAMZA ALDABBAS⁵

¹Department of Computer Science, University of Engineering and Technology at Taxila, Taxila 47050, Pakistan

²Department of Computer Science, University of Chakwal, Chakwal 48800, Pakistan

³Department of Information Technology, Imam Mohammad Ibn Saud Islamic University, Riyadh 11432, Saudi Arabia

⁴Department of Electrical and Computer Engineering, The University of North Carolina at Charlotte (UNC-Charlotte), Charlotte, NC 28223, USA

⁵Prince Abdullah bin Ghazi Faculty of Information and Communication Technology, Al-Balqa Applied University, Al-Salt 1705, Jordan

Corresponding author: Rashid Amin (rashid4nw@gmail.com)

The authors extend their appreciation to the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University for funding and supporting this work through Research Partnership Program No. RP-21-07-05.

ABSTRACT Because of the recent exponential rise in attack frequency and sophistication, the proliferation of smart things has created significant cybersecurity challenges. Even though the tremendous changes cloud computing has brought to the business world, its centralization makes it challenging to use distributed services like security systems. Valuable data breaches might occur due to the high volume of data that moves between businesses and cloud service suppliers, both accidental and malicious. The malicious insider becomes a crucial threat to the organization since they have more access and opportunity to produce significant damage. Unlike outsiders, insiders possess privileged and proper access to information and resources. In this work, a machine learning-based system for insider threat detection and classification is proposed and developed a systematic approach to identify various anomalous occurrences that may point to anomalies and security problems associated with privilege escalation. By combining many models, ensemble learning enhances machine learning outcomes and enables greater prediction performance. Multiple studies have been presented regarding detecting irregularities and vulnerabilities in network systems to find security flaws or threats involving privilege escalation. But these studies lack the proper identification of the attacks. This study proposes and evaluates ensembles of Machine learning (ML) techniques in this context. This paper implements machine learning algorithms for the classification of insider attacks. A customized dataset from multiple files of the CERT dataset is used. Four machine learning algorithms, i.e., Random Forest (RF), Adaboost, XGBoost, and LightGBM, are applied to that dataset and analyzed results. Overall, LightGBM performed best. However, some other algorithms, such as RF or AdaBoost, may perform better on some internal attacks (Behavioral Biometrics attacks) or other internal attacks. Therefore, there is room for incorporating more than one machine learning algorithm to obtain a stronger classification in multiple internal attacks. Among the proposed algorithms, the LightGBM algorithm provides the highest accuracy of 97%; the other accuracy values are RF at 86%, AdaBoost at 88%, and XGBoost at 88.27%.

INDEX TERMS Privilege escalation, insider attack, machine learning, random forest, adaboost, XGBoost, LightGBM, classification.

The associate editor coordinating the review of this manuscript and approving it for publication was Wentao Fan.

I. INTRODUCTION

Cloud computing is a new way of thinking about how to facilitate and provide services through the Internet. The current

financial crisis, as well as the expanding computing demands, have necessitated significant changes to the current Cloud Model in terms of data storage, processing, and display [1]. Cloud computing prevents people from spending a lot on equipment maintenance and purchases by utilizing cloud infrastructure. Cloud storage providers adopt fundamental security measures for their systems and the data they handle, including encryption, access control, and authentication. Depending on the accessibility, speed, and frequency of data access, the cloud has an almost infinite capacity for storing any type of data in different cloud data storage structures. Sensitive data breaches might occur due to the volume of data that moves between businesses and cloud service providers, both inadvertent and malicious. The characteristics that make online services easy to use for workers and IT systems also make it harder for businesses to prevent unwanted access [2]. Authentication and open Interfaces are new security vulnerabilities that Cloud services subject enterprises face. Hackers with advanced skills utilize their knowledge to access Cloud systems. Machine learning employs a variety of approaches and algorithms to address the security challenge and better manage data. Many datasets are private and cannot be released owing to privacy concerns, or they may be missing crucial statistical properties [3], [4]. The fast rise of the Cloud industry creates privacy and security risks governed by regulations. Employee access privileges may not necessarily change when they change roles or positions within the Cloud Company. As a result, old privileges are used inconveniently to steal and harm valuable data. Each account that communicates with a computer has some level of authority. Server databases, confidential files, and other services are often restricted to approved users. A malicious attacker can access a sensitive system by gaining control of a higher user account and exploiting or expanding privileges. Based on their objectives, attackers can move horizontally to obtain control of more systems or vertically to obtain admin and root access till they have complete control of the whole environment [1]. When a user gets the access permissions of another user with the same access level, this is known as horizontal privilege escalation. An attacker can use horizontal privilege escalation to access data that does not necessarily relate to him. An attacker may be able to uncover holes in a Web application that provides him entry to certain other people's information in badly designed apps [3], [5]. Because the attacker has completed a horizontal elevation of privileges exploit, they can see, alter, and copy sensitive information. Figure 1 illustrates the scenario of how the horizontal privilege escalation attack happened among the entities of the organizations. This form of assault usually necessitates a thorough knowledge of the weaknesses that impact specific operating systems and the usage of malicious programs. It's also called privilege elevation assault, which entails giving a user, software, or other assets more rights or privileged access than they already have. Moving from a low degree of privileged access to a greater level of special access is the key objective of the attacker [6]. To achieve vertical access

control, the attacker may need to take various actions to overcome or override security restrictions. Vertical privileges controls are finer-grained versions of security models that implement business objectives like separation of roles and least privilege, as shown in Figure 2. An attacker, for example, takes control of an ordinary registered user on a network and tries to acquire administrative or root access. Anomaly activity on organizational systems or user accounts can be detected using behavioral analytics, which might signal intrusion or privilege escalation.

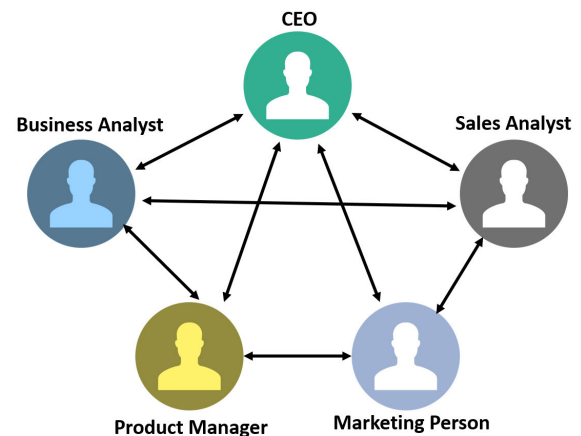


FIGURE 1. Horizontal Privilege Escalation.

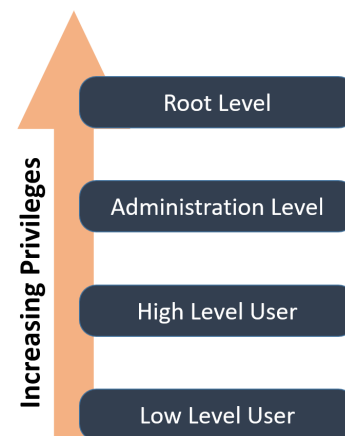


FIGURE 2. Vertical Privilege Escalation.

Attackers target data sources because they have the most valuable and sensitive information. Every cloud user's privacy and security are affected if data is lost. Insider threats are harmful operations carried out by people with authorization. With the fast growth of networks, many companies and organizations have established their internal networks. According to recent estimates, 90% of businesses believe they are vulnerable to insider assaults [7]. Attackers can use privilege elevation to open up additional attack routes on a target system. Insider attackers try to get higher privileges or access to more sensitive systems by attempting privilege

escalation. Insider attacks are difficult to identify and prevent because they exist beneath the enterprise-level security defense measures and frequently have privileged access to the network. Detecting and classifying insider threats has become difficult and time-consuming [8]. In recent studies, researchers worked on detecting and classifying privileged elevation attacks from insider personnel. They proposed different machine learning and deep learning techniques to counter these challenges. Techniques like SVM, Naïve Bayes, CNN, Linear Regression, PCA, Random Forest, and KNN were applied in recent studies. However, the demand for fast and effective machine learning algorithms is highly valued with the diversity of attack types. Therefore an effective and efficient strategy is required to detect, classify and mitigate these insider attacks.

To get better security protection systems, we need intelligent algorithms, such as ML algorithms, to classify and predict insider attacks [17]. In addition, knowing the performance of ML algorithms on classifying insider attacks allows you to choose the most appropriate algorithm for each case, and the ones (ML algorithms) need to be improved. So you can provide a higher level of security protection. This research aims to apply effective and efficient ML algorithms to insider attack scenarios to gain better and faster results. ML algorithms have been applied and evaluated in this regard: Random Forest, AdaBoost, XGBoost, and LightGBM. The principle behind the boosting strategy is to take a weak classifier and train it to become a very good one by raising the prediction of the classification algorithm. Random Forest, AdaBoost, and XGBoost worked accurately and quickly to classify insider threats. These are the contributions that this research intends to make:

- 1) In order to generate findings that represent real-world situations, this work assumes a realistic context for ML model training. After this, the work emphasizes the differences from training under conventional ML conditions.
- 2) Create and analyze a user-centered insider attack detection process, including data collection, pre-processing, and ML model-based data analysis.
- 3) To better understand insider attack situations offer a detailed result reporting procedure where instance and user-based results are presented and malicious incidents are evaluated.

To the best of our knowledge, this is the first paper that deals with measuring the performance of the four machine learning algorithms (Random Forest, AdaBoost, XGBoost, and LightGBM) on classifying insider attacks and using this (algorithm performance) to quickly identify appropriate defense tools that improve the level of security protection. Recent insider threat detection and classification studies used different models and ensemble techniques. Those studies individually implemented the models on different datasets and then gave the classification results. This paper implemented the four ensemble models on a single customized dataset to better detect and classify insider threats. Our study presented

the best results of applied ensemble algorithms. The research structure is organized as follows. Section I describes the Introduction to Cloud and privilege escalation attacks. Section II is the summarization of the previous work related to this study. Section III describes the proposed methodology that implements the machine learning algorithms. Section IV is a detailed representation of the applied dataset, experimental setups, and the results evaluations. The conclusion is drawn in Section V.

II. LITERATURE REVIEW

Le et al. [9] discussed that insider threats are among the most expensive and difficult-to-detect forms of assault since insiders have access to a company's networked systems and are familiar with its structure and security processes. A unique set of challenges face insider malware detection, such as extremely unbalanced data, limited ground truth, and behavioral drifts and shifts. Machine learning is used to analyze data at several levels of detail under realistic situations to identify harmful behaviors, especially malicious insider attacks. Random Forest beats the other ML methods, achieving good detection performance and F1-score with low false positive rates in most situations. The proposed work achieved an accuracy of 85% and a false positive rate of only 0.78%. Janjua et al. [10] discussed that preventing malicious insiders from acting maliciously in an organization's system is a significant cybersecurity challenge. The paper's main goal is to use several Machine Learning approaches to classify email from the TWOS dataset. The following supervised learning techniques that have been used on the dataset are Adaboost, Naïve Bayes (NB), Logistic Regression (LR), KNN, Linear Regression (LR), and Support Vector Machine (SVM). Experiments reveal that AdaBoost has the best classification accuracy for harmful and non-malicious emails, with a 98% accuracy rate. Although the model was trained on the original dataset, the data is limited. The model's results may be improved if the dataset is bigger.

Kumar et al. [11] discussed that due to the large number of diverse apps operating on shared resources, implementing security and resilience on a Cloud platform is necessary but difficult. Inside the Cloud infrastructure. Based on the idea of clustering, a novel malware detection technique was suggested: trend micro locality sensitive hashing (TLSH). They utilized Cuckoo sandbox, which generates dynamic file analysis results by running them in a separate environment. Principal component analysis (PCA), random forest, and Chi-square feature selection approaches are also used to choose the essential features. Experimental outcomes for clustering and non-clustering methods are obtained for three proposed classifiers. According to the outcomes of the experiments, the Random Forest obtains the best accuracy among the other classifiers. Cloud security has long been a serious concern. Attackers target data sources because they want the most valuable and sensitive information. If data is lost, every Cloud user's privacy and security is seriously threatened. Internal attackers get access to a system by compromising

a susceptible user node. Internally linked to the cloud network, they conduct assaults while posing as trusted users. The use of Improved LSTM to identify internal attackers in a cloud network is offered as a security technique. Not only does the proposed ILSTM identify internal attackers, but it also minimizes false alert rates by distinguishing broken and new user nodes from malfunctioning nodes.

Le and Zincir-Heywood [2] discussed that insider threat actions could be taken intentionally or accidentally, like information system sabotage or irresponsible working with cloud resources. One of the difficulties in researching insider threats is that a malicious insider has access to the organization's network systems and is familiar with its security processes. To assist cybersecurity experts in detecting harmful insider activity on unseen data, ANN, RF, and LR machine learning techniques are taught on finite ground truth. User-session data looks to be the greatest choice for data granularity since it enables a system with a significant malicious insider detection rate and quick response times. They used machine learning techniques such as RF and ANN, which performed well in this work. Because RF provides excellent precision, it can be used when manpower for examining alerts is restricted. Tripathy et al. [12] discussed that conventional web-based and cloud apps are vulnerable to the most popular online threats. One of the greatest threats to a SaaS application is the SQL injection attack. They construct and test the classification for SQL attack detection using machine learning methods. They explore the ability of machine learning models to identify SQL injection attacks, including the AdaBoost Classifier, Random Forest, and Deep Learning utilizing ANN, TensorFlow's Linear Classifier, and Boosted Trees Classifier. More important than malicious reading activities are malicious writing operations. The random forest classifier surpasses all others on the dataset and obtains better accuracy.

Sun et al. [13] discussed that the network is becoming increasingly integral to businesses and organizations. So there is an increase in network security threats. Data leakage incidents from 15 nations and 17 industry groups were examined for Ponemon's 2018 Cost of a Data Breach Study, with 48% being malicious operations. While insiders' faulty actions were the cause of 27% of the incidents. They used the tree structure technique to study user behavior and create the feature sequence in this article. To distinguish between the feature patterns and detect unusual users, the COPOD approach is adopted. Additionally, the detection effect outperforms the standard unsupervised learning approach. Processing vast amounts of complicated and diverse data using this way provides benefits. Kim et al. [14] discussed that the authorized user's malicious acts, such as stealing intellectual property or sensitive information, fraud, and sabotage, are examples of insider risks. Although insider threats are far less common than external network assaults, they can still do significant harm. There are three widely used research methodologies for detecting insider threats. Making a rule-based detection system is the first method. The second technique is to create a network graph and monitor modifications in the graph's

structure to spot suspicious people or bad behavior. The third technique uses historical data to create a statistical or machine-learning model that can predict potentially dangerous activity. They utilized the "CERT Insider Threat Tools" dataset since obtaining genuine business system logs is extremely challenging. Employee computer actions logs are included in the CERT dataset and certain organizational data such as employee's departments and responsibilities. They built insider-threat detection models to emulate real-world companies using machine learning-based methods. Experiments indicate that the suggested system can detect harmful insider activities relatively effectively.

Liu et al. [15] discussed that information communication technology systems are increasingly vulnerable to cyber security attacks, most of which come from within the organization. Detecting and mitigating insider threats is a complicated challenge because insiders are hidden behind enterprise-level security defense measures and frequently have privileged network access. By gathering and reassembling information from the literature, they present the many types of insiders and the threats they bring. Insider threats are of three types: Masquerader, Traitor, and Unintentional perpetrator. Prevention may be viewed as a set of defensive procedures that can help prevent or enhance the identification of various internal threats. They examine the suggested efforts from a data analytics viewpoint, presenting them in terms of host, network, and contextual data analytics. Meanwhile, relevant studies are analyzed and compared, with a brief overview to show the benefits and drawbacks. Wang et al. [16], [17] discussed that the insiders are an organization's trusted partners who have access to the organization's assets, information, and network. Over 60% of all security breaches or assaults documented worldwide in 2015 were committed entirely by insiders. As a result, preventing insider threats is a severe problem. The major goal of this work is to create a reliable insider threat detection system that can distinguish between malicious and non-malicious insider activity. The examination of human behavioral activities will be the main emphasis of this paper. They examine three scenarios related to the behavioral activity of the insider user. These scenarios are as follows:

- A user performs activities after working hours using a removable device to access and steal data.
- Before leaving the current job, the user's frequency of using the thumb drive increases and then is used to steal important company data.
- Users download some spyware software to get the passwords of the employees of the organization, and after getting the passwords, they try to steal the supervisor's credentials. After that, they generate fake alarming emails to create panic in the organization.

Tariq et al. [18] discussed that Deep learning, also known as multilevel and deep-structured learning, is a subset of machine learning methods that can be supervised or unsupervised. The DL's encrypted data comes from learning and interface modules and is its main issue. Due to the widespread adoption of DL models in numerous applications, security

and privacy concerns are of utmost importance. Due to numerous Deep Neural Network properties, which rely on a significant quantity of input training data, privacy issues constantly exist. The Industries and researchers have focused on many Deep Learning security threats and associated defenses.

Berman et al. [19] discussed that the set of procedures, methods, tools and technologies collectively called “cyber security” are used to safeguard computing resources’ availability, confidentiality, and integrity. There is evidence of compromise throughout an attack’s life cycle, and there may even be important warning signs of an upcoming attack. Finding these indications, which could be scattered across the environment, is difficult. Many data are produced by machine-to-machine and human-to-machine exchanges from apps, websites, electronic objects, and other cyber-enabled resources. Malware threats are becoming more prevalent and diverse, making it more challenging to protect against them using conventional techniques. DL offers the chance to create generalized models for malware detection and classification. Network behavior-based approaches are required to identify complex malware since they focus on the synchronized command and control traffic from the malware. Pang et al. [20], [21] discussed that insider threat, which may lead to data theft or system sabotage, is one of today’s main cyber security issues. Although insider threats can substantially harm, their objectives and activities might differ greatly. The use of anomaly-based intrusion detection methods is a useful way of identifying both known and undiscovered/unknown threats. The type of anomaly is a key element in anomaly identification. There are three subcategories of anomalies: point anomalies, contextual anomalies, and collective anomalies. In anomaly-based intrusion detection systems, a model is created by training the system using “normal” network data. When the system’s model is ready, it is then utilized to determine whether or not new events, objects, or traffic are abnormal. Deep learning is a subset of machine learning algorithms that uses several levels of information processing steps in hierarchical structures to learn features unsupervised and evaluate or classify patterns. On the KDDCup99 dataset, the system is developed using two techniques: RBM and Autoencoder. Based on the data currently available from the KDD99 dataset, a statistical analysis is done on the values of each characteristic. Tests using connections to the KDD-Cup99 network traffic have demonstrated that Deep Learning algorithms efficiently detect intrusions with minimal error rates. Coppolino et al. [22], [23] discussed that security is a major issue since cloud services handle sensitive data that may be accessible from anywhere over the Internet. Malevolent insiders frequently wreak significantly more harm than is anticipated. Such attackers inject insecure code into the cloud and use their equipment as a channel. When correctly injected, this code performs maliciously, and the user running it has power over it. The significance of this code’s capacity to give the malicious user access to information depends on the strength of the developed code and the degree of security

measures implemented by the cloud. Cloud Ecosystem that aims to offer security controls across all Clouds. The system aims to guarantee data privacy and security from the user authentication procedure through cloud storage. One Time Password (OTP) was made possible by the system’s design’s consideration of verified authentication. The CloudSim simulator is used to model both the proposed system and algorithm.

Abdelsalam et al. [24] discussed a Deep Learning-based malware detection technique (DL). Employing raw, process behavior (performance metrics) data, the study demonstrated the usefulness of using a 2D Convolutional Neural Network (CNN) for malware detection. The study illustrates the effectiveness of the proposed method by first developing a standard 2D CNN model which does not include the time window, and then making comparisons it to a newly developed 3D CNN model that greatly enhances detection accuracy, because of the use of a time window as the third dimension, thereby minimizing the problem of mislabeling. Results revealed a reasonable accuracy of 79% on the testing dataset by using 2D CNN.

Jaafar et al. [25] illustrated that information systems are created to provide services and functions to a large number of people. As a result, it is common to have multiple levels of privilege for different users on the same information system. By identifying irregularities and flaws in information systems, several studies have been published to find security issues or attacks associated with privilege escalation. In the article, the study first introduces a new distance-based outlier detection technique for detecting unexpected situations of privilege escalation assaults without making any assumptions about the dataset or distribution. Second, based on known privilege escalation scenarios, the study identifies four kinds of privilege escalation assaults and the justification for their specifications.

Alhebaishi et al. [26] discussed that the growing use of cloud computing brings with it plenty of new security and privacy issues. In order to execute their assigned maintenance responsibilities, remote administrators must be given the proper privileges, which may include direct access to the underlying cloud infrastructure. A dishonest remote administrator, or an attacker who has stolen an administrator’s credentials, might pose serious internal risks to the cloud. The study starts by modeling the maintenance jobs and their associated rights. The study was next uses the current k-zero day safety metric to represent the insider threats caused by remote administrators allocated to maintenance tasks.

Yuan et al. [27] demonstrated that the approach for detecting insider threats is to model a user’s usual behaviour and look for anomalies. To determine if user behaviour is normal or abnormal, it is to present the unique insider threat detection approach. Specifically, using the LSTM to categorize the user action sequence directly is inefficient, because each sequence is represented by a single bit of information in the LSTM output. The proposed model works in two stages: In the first

stage, the LSTM is used to extract the temporal features about the behaviour of the user, then these features are converted to fixed-size feature matrices. In the second stage CNN is used to classify fixed-size matrices as normal or anomaly.

Mohammed [28] demonstrated that Cloud computing has become more vital for today's businesses to satisfy their demands. The present popularity of cloud web services is a result of their affordability and accessibility. Several adaptable service models, including IaaS, SaaS, PaaS, and multi-tenancy, are used to achieve this. Security and privacy risks associated with these cloud services are serious. By limiting illegal access, the combination of verification and attribute-based access control enhances the performance of the cloud web application. Relying on identification and access control systems to prevent unauthorized use of the systems, which is among the most typical circumstances, is one of the biggest risks. Despite the risks and challenges, there are more benefits to having an IAM system than problems. Future cost savings and use cases will emerge, but they will probably only be available to businesses with strong cloud identity standards. The majority of firms looking to establish themselves for long-term success have found that identity as a service provides the best route forward.

III. RESEARCH METHODOLOGY

The malicious insider becomes a crucial threat to the organization since they have more access and opportunity to produce significant damage. Unlike outsiders, insiders possess privileged and proper access to information and resources. Furthermore, insiders are well-versed in the organization's vital assets. As a result, identifying and understanding insider attackers and their objectives requires good internal threat classification. Insider risks may be defined and addressed using criteria including insider indications, detection approaches, and insider kinds. There are two sorts of analysis intervals: real-time, which may identify malicious activity in real-time, and offline anomaly detection, which gathers log data and looks for certain patterns. Both purposeful and accidental cyber-attacks on the information and the use of unauthorized activities to affect the information's availability, integrity, or secrecy are examples of authorized misuse actions. The threat approach determines the method for detecting malicious agents. Attackers might easily introduce random data into the distributed algorithm to prevent it from convergence [29], [30]. Figure 3 shows the attacker's approaches toward the user's system of an organization for stealing sensitive information or performing some serious damage to the data. Attackers can also attack through email by sending malicious code or URLs to the desired user accounts. He got control or credentials of that user and later use these details for further attacks.

The limitations of traditional machine learning for attack detection include their inability to automatically design features, poor detection rate, and inability to identify tiny mutants of known attacks and insider attacks. In the majority of circumstances, the ensemble of models will perform

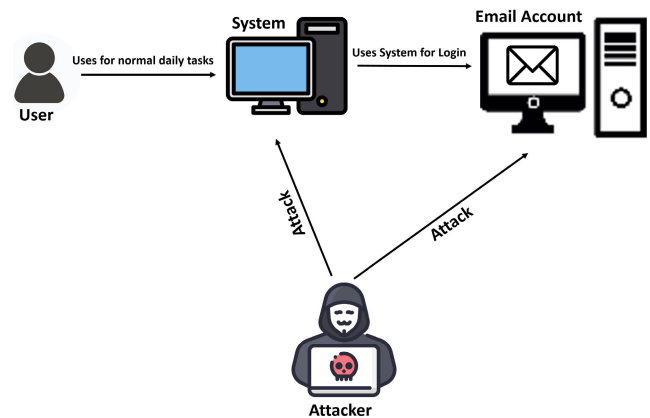


FIGURE 3. Privilege Escalation Attack Process.

better than the individual models on insider threat detection and classification. The combined output of several models is almost always lesser noisy than the sum of the individual models. Model consistency and robustness result from this approach. Both linear and non-linear connections in the data are captured by ensemble models. It is achieved by combining two separate models into one ensemble. The proposed methodology consists of well-known supervised machine learning algorithms, i.e., Random Forest, AdaBoost, XGBoost, and LightGBM. These models utilized datasets for detection and classification. Implementing these models contains a series of steps, including dataset preprocessing, model training, model testing, detection, and classification [31], [32]. Several challenges were faced when implementing the proposed ML algorithms. A major concern was the dataset collection for insider attacks. After evaluating the datasets from multiple sources like dataset repositories and websites, the final dataset has been selected for implementation. Before training the models, the dataset was analyzed completely for its quality. Some features contain missing values and outliers. The 'size' feature of the dataset contained some outliers, that were removed by averaging the neighboring values. The 'File Copy' feature had some values missing. The dataset pattern of that feature then fills those missing values. High-quality data is very important for better results. Irrelevant features in the dataset also impacted the training of models. So in this context, irrelevant features are removed i.e., 'employee', and 'file tree', and train the models on specifically selected features. The initial results were not promising during the evaluation of the proposed models. We tuned these parameters, i.e., learning rate, maximum depth, and K-fold, to get efficient results.

A. RANDOM FOREST

Among some of the machine learning techniques used in supervised learning is Random Forest, which is widely recognized. It may be used in machine learning to address various regression and classification issues. It's an ensemble learning method that combines many classifiers to

tackle complex problems and enhance the model's efficiency [33], [34]. The advantages of Random Forest are that classification and regression issues can be handled with Random Forest. Large datasets with several dimensions may be handled by it. It increases model accuracy and solves the overfitting issue. Obtain the relative feature significance, which helps choose the classifier's most beneficial features. Implementation Steps are as follows:

- Preprocessing of Dataset
- Random Forest algorithm Training
- Random Forest algorithm Testing
- Model Accuracy
- Visualize Results

The relevance of each feature in the random forest is determined using Gini importance or mean decrease in impurity (MDI). The total decline in node impurity is another name for the Gini importance. This is the reduction to fit the model or accuracy caused by removing a variable. The significance of the variable increases with the size of the decline. Here, the mean decline is key in determining which variables to use. The entire explanatory power of the variables may be expressed using the Gini index. Random forest works based on the decision tree. Multiple decision trees were created based on randomly selected features from the dataset. For the insider attack dataset, the random forest classifier selects the features and makes a decision tree on that dataset. The classification yields 0 for no attack and 1 for an attack. All the generated decision trees yield 0 or 1. The combination of bootstrapping and aggregation works in a random forest. Then the outcomes of each decision are checked, and then the outcome of random forest is the majority of that decision tree's outcomes. For example, if the majority outcomes are 1 from decision trees, the final prediction will be 1 and vice versa. Figure 4 demonstrate the working of random forest to classify insider attack. Starting from the main dataset, the random subsets of the datasets for generating decision trees. Decision trees yield in 0 or 1, and then random forest algorithm outcomes 0 or 1 basis on the majority existence. Some important feature of random forest are as follow:

- Diversity
- Immune to the curse of dimensionality
- Parallelization
- Train-Test Split
- Stability

1) DETECTING ATTACKS USING RANDOM FOREST

Figure 4 shows the Random Forest's stepwise working for classifying insider threats. The dataset is subdivided into sets of the dataset and then a random classifier builds a decision tree on each subset of the dataset. Each decision tree predicted an outcome. All the outcomes of decision trees were then evaluated on the basis of majority voting. The final prediction is the most frequent outcome of decision trees.

B. AdaBoost

AdaBoost often referred to as Adaptive Boosting, is a machine learning strategy used as a component of the

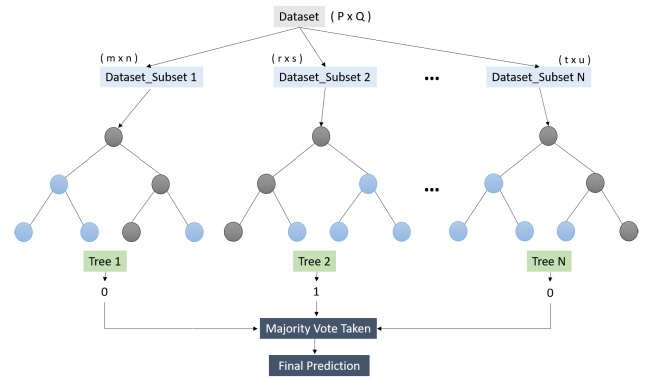


FIGURE 4. Random Forest Classifier for Insider threat classification.

ensemble method. It builds a model that gives each piece of data equal weight. The improperly classified points are subsequently given more weight. All points with higher weights are prioritized more in the next model. Models will keep being trained up till a low error is detected. The weight-assigning approach used after each iteration distinguishes the AdaBoost algorithm from all those other boosting algorithms, which is its strongest attribute. Unlike other algorithms, it is easy to use and requires less changing of parameters. Overfitting is not a problem with AdaBoost. AdaBoost can help poor classifiers increase their accuracy.

1) AdaBoost OPERATION

AdaBoost is an iterative ensemble algorithm. AdaBoost classifier combines several weak classifiers to create a powerful classifier that has a high degree of accuracy. The fundamental idea underlying Adaboost is to train the data sample and adjust the classifier weights in each iteration to provide accurate predictions of uncommon observations. It strives to minimize training errors to offer the best fit possible for these instances in each iteration. The AdaBoost algorithm selects the random training subset for the insider attack dataset. The subset of the dataset is utilized for the training of the Adaboost algorithm. AdaBoost gives incorrectly classified observations a higher weight so that they will have a higher chance of being correctly classified in the next iteration. Additionally, based on the trained classifier's accuracy, weight is assigned to it in each iteration. The more precise classifier will be given more weight. This method iterates until the entire training set fits perfectly or until the largest number of estimators has been reached. When choosing a base learner, Gini and Entropy are considered. The base learner will be the stump with the lowest Gini or Entropy. The output it may create while traveling through the first stump is 1. Once through the second stump, the output may once more be created as 1. It may produce 0 when going through the third stump. Similar to random trees, the majority of votes in the AdaBoost method also occur between the stumps, and then the final prediction is achieved by voting among all stumps. The mathematical approach of the random

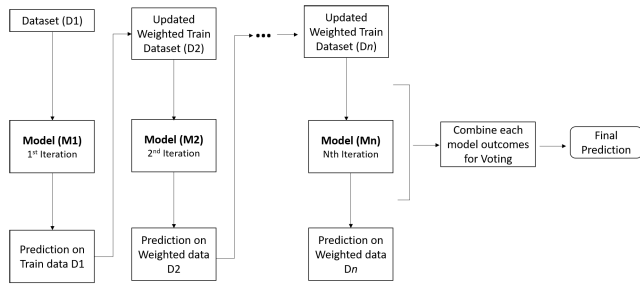


FIGURE 5. Adaboost Classifier for Insider threat classification.

forest classifier is given below: Eq 1 is used to assign the sample weights to the target class of the dataset. The Gini Impurity is then calculated using Eq 2. The Alpha value will be calculated by using Eq 3 to measure the correctness of sample classification. After 1st iteration, the weights for the next iteration will be calculated for both correctly and incorrectly classified by using Eq 4 and Eq 5.

Step 1: Assigning Sample weights.

$$SampleWeight = \left[\frac{1}{(NumberOfSample)} \right] \quad (1)$$

Step 2: Calculating Gini Impurity for each feature.

$$GiniImpurity = 1 - (TrueProbability)^2 - (FalseProbability)^2 \quad (2)$$

Step 3: Calculating Amount of Say for created Stump.

$$AmountofSay = \frac{1 \log(1 - totalerror)}{2(totalerror)} \quad (3)$$

Step 4: Calculation for new weights for next stumps.

$$\begin{aligned} &Newsampleweightforincorrectsamples \\ &= sampleweight * eamountofsay \end{aligned} \quad (4)$$

$$\begin{aligned} &Newsampleweightforcorrectsamples \\ &= sampleweight * e-amountofsay \end{aligned} \quad (5)$$

Step 5: Randomly selected a new sample of a dataset based on the new sample weight.

Step 6: Process repetition by N numbers of times.

In figure 5, the implementation of the Adaboost classifier is demonstrated. The subset of the dataset is utilized for the training of the Adaboost algorithm. AdaBoost gives incorrectly classified observations a higher weight so that they will have a higher chance of being correctly classified in the next iteration. All the models on a subset of the dataset with higher weightage scenarios were then analyzed on the basis of majority voting. The final prediction is based on the majority outcome of the models.

C. XGBoost

XGBoost is a flexible and extremely accurate gradient-boosting system that pushes the boundaries of computing capabilities for boosted tree methods. It has the advantages of improving the algorithm and modifying the model and

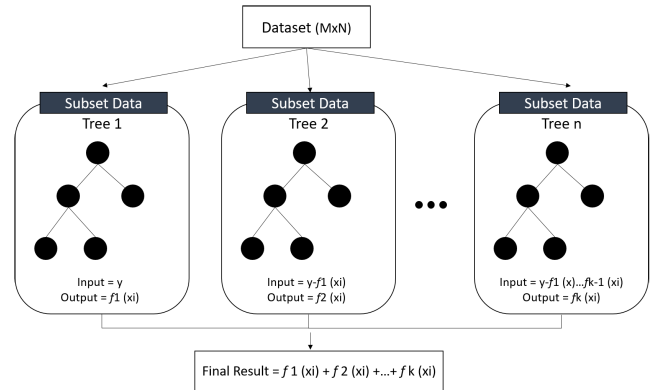


FIGURE 6. XGBoost Classifier for Insider threat classification.

can also be used in computing infrastructure [35]. Regression and classification problems are among those it is used to address. This method involves building decision trees one at a time. Weights are highly important in XGBoost. Weights are assigned to each independent variable, which is then put into the decision tree and predicts outcomes. It is less time-consuming than Gradient Boosting and is designed to deal with incomplete data with the help of its built-in abilities. The user may perform cross-validation after each loop. It is good for small to larger datasets. Two equations are used to calculate the similarity scores and the new residuals. Eq 6 is used for calculating the similarity score and then Eq 7 is used to calculate the new residuals for the next iteration of the algorithm.

$$SimilarityScore = \frac{(Gradient)^2}{(Hessian + \lambda)} \quad (6)$$

$$Newresiduals = Oldresiduals + pPredictedresiduals \quad (7)$$

XGBoost is a distributed gradient boosting library developed to be very effective, adaptable, and portable. It uses the Gradient Boosting framework to construct machine learning algorithms. Figure 6 demonstrated the XGBoost classifier works on the dataset for classification. By consistently separating features, XGBoost grows new trees. In reality, each time it adds a tree, it learns a new function to reflect the previous prediction's residual. The characteristics of the prediction data will have a matching leaf node for each tree when K trees are created after training, so each leaf node will correlate to a score. To determine the sample's recognition prediction value, the matching scores from each tree are finally added up.

D. LightGBM

It's a boosting strategy that applies techniques for tree-based learning, which are regarded as a very effective processing method. It is thought to be an efficient processing method. Unlike other algorithms, which construct their trees horizontally, the LightGBM method grows vertically, which means it grows leaf-wise while other methods grow level-wise.

With its processing speed and speedy delivery of results, LightGBM is termed “Light.”

1) LightGBM OPERATION

In contrast to previous boosting algorithms that develop trees level-by-level, LightGBM divides the tree leaf-wise. It selects the leaf with the greatest delta loss for growth. Figure 7 shows the implementation architecture of the LightGBM. The leaf-wise growth technique is more effective since it only divides the leaves with the greatest information gain across the same layer. The learning rate (Lr), number of leaves, and maximum depth are a few key parameters that we altered during the construction of Lightgbm. Lr is a super parameter that regulates how quickly the model’s internal parameters are updated. LightGBM is a type of robust machine learning model built on the decision tree that is fast, stable, and has good accuracy and predictive power.

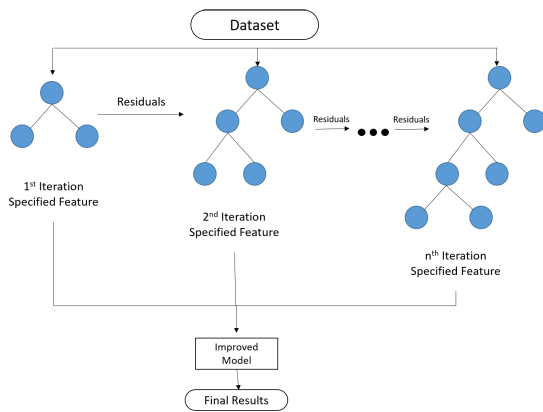


FIGURE 7. LightGBM Classifier for Insider threat classification.

E. OVERVIEW OF PROPOSED METHODOLOGY

Figure 8 is the overview of proposed models for the classification of privilege escalation attacks. A customized dataset from multiple files of the CERT dataset is used in this work. Machine learning algorithms, Random Forest, AdaBoost, XGBoost, and LightGBM are applied to that dataset and give better results. the main goal is to provide an overview of the applied framework for the detection and classification of an insider attack. The proposed model consists of four algorithms that were applied to CERT dataset. The mathematical and technical analysis is predefined in the earlier sections, where implemented models contained both mathematical equations and technical details.

The main goal of ensemble learning is to enhance a model’s performance. Ensemble learning includes bagging, boosting, and stacking. Bagging and boosting techniques are used in our study to detect and classify insider threats. Data aggregation during the data pre-processing phase enables us to extract data that offers insightful information about how well the models work. Data normalization is a very helpful method of transforming characteristics to be on a comparable scale

during the data preparation stage. The model performs better and maintains training stability as a result. The feature extraction process became essential in lowering the volume of redundant data in the data collection. In the end, the data reduction speeds up the learning and generalizations phases of the machine learning process while also enabling the model to be built with less machine effort. In order to reduce training mistakes, boosting is an ensemble learning technique that combines a number of base learners into strong learners. So in this perspective, the boosting technique is utilized, which lies in ensemble learning.

The best algorithm among these four is LightGBM which shows the highest accuracy. Table 1 demonstrated the factors which play their roles in the high performance of these algorithms.

TABLE 1. Overview of Algorithms with performance factors.

Algorithms	Different Performance Factors
Random Forest	Hyperparameter ($n_{estimators}$)
AdaBoost	Multiple decision stumps.
XGBoost	Increased attributes.
LightGBM	learning_rate, feature_fraction, bagging_freq, max_depth, bagging_fraction and min_data_in_leaf.

One of the most often used computer languages is Python, which has displaced many other languages in the field primarily due to its enormous library set. We have implemented the following best Python libraries as shown in table 2 in the proposed study.

TABLE 2. ML Libraries with different operations.

ML Libraries	Mathematical and Working Functions
Numpy	Linear Algebra
Scikit-learn	Dataset Exploring, Features Selection, Dataset preparation, and Training / Testing of Models.
Keras	Comparative Analysis
Pandas	Data Cleaning and Analysis
Matplotlib	Data visualization and graphs

Figure 9 explains the flow of the proposed technique, in which data is gathered from the dataset and is preprocessed. Machine learning models are trained using the data, and testing is performed based on the ratio of the dataset.

F. PERFORMANCE EVALUATION

The experiments are done on a Linux operating system, Ubuntu 18.04 on an ACER Aspire 5349 machine with a 3rd generation Intel Core i5 processor and 6 Gb of RAM for the multichain network. In this work, the CERT dataset is utilized. This dataset includes many features for the detection and classification of insider threats. This dataset includes multiple files for different activities performed by the user or attacker. From this newly updated dataset, this paper selects multiple features from multiple files and merges them to make a single dataset file for effective experimental setup

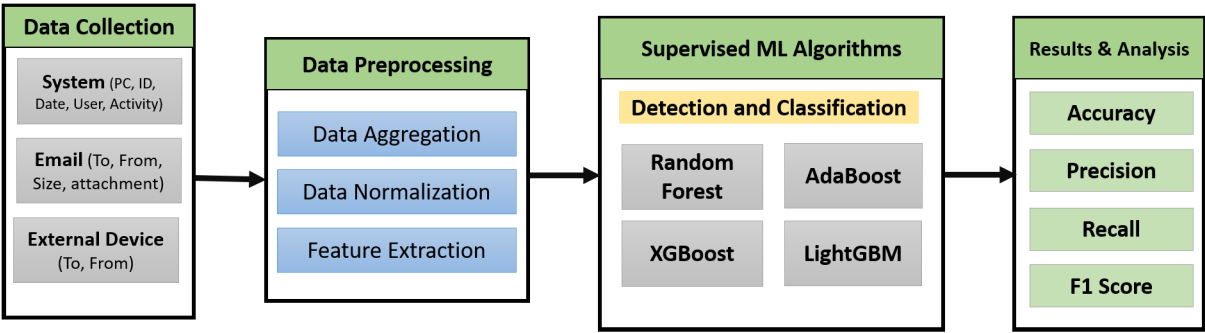


FIGURE 8. Overview of Privilege Escalation Attack Proposed Models.

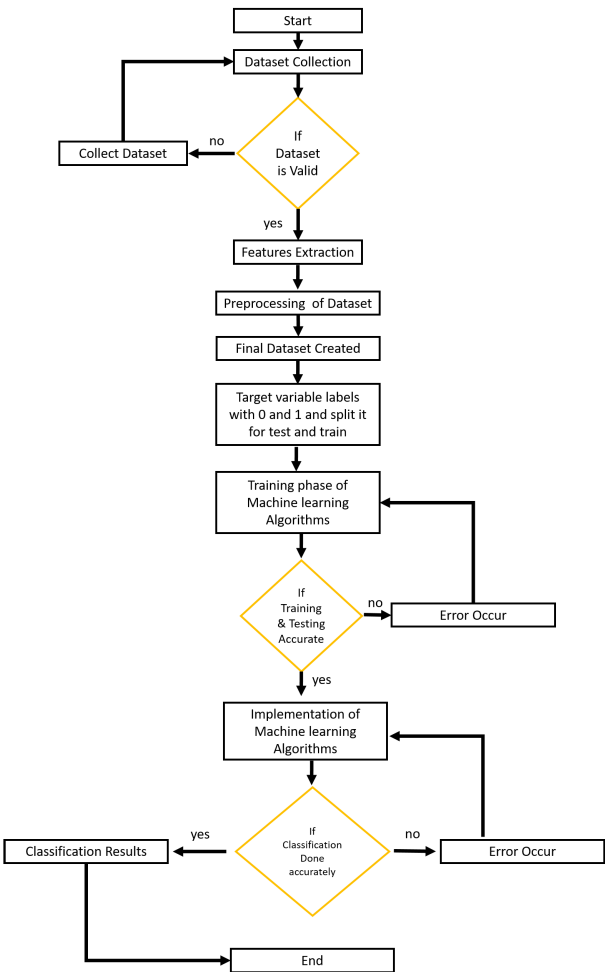


FIGURE 9. Flow chart of the proposed technique.

and results evaluation. All the files in that dataset are CSV format, so it is easily analyzed, preprocessed, and applied by proposed models for better results generation.

Figure 10 represents the features and the value range of the given features. These features are specifically gathered from multiple dataset files. These relevant features show the most important ways the attacker performs insider attacks.

Figure 11 shows the distribution of user actions along with the pc number they contain. It is shown that most of the action

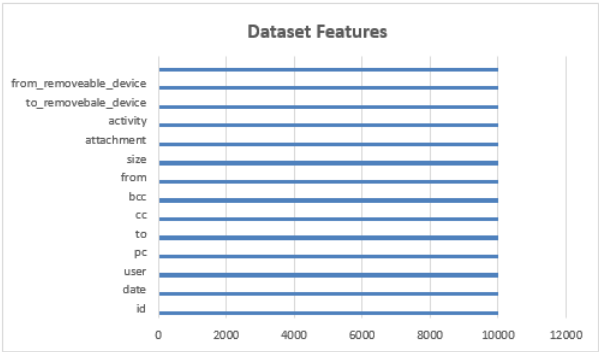


FIGURE 10. Features of Dataset.

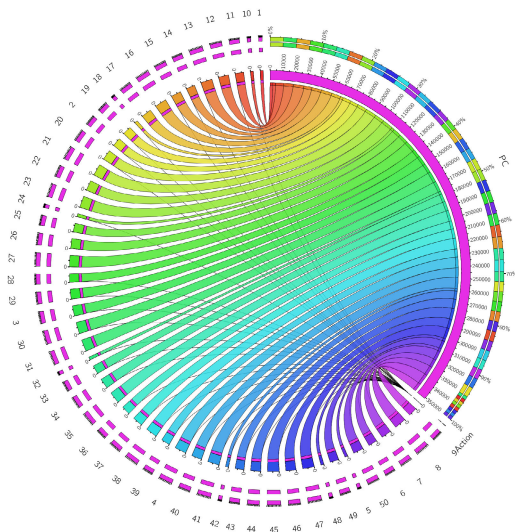


FIGURE 11. Demonstration of User Action from respective PC.

is filed open and then file copy, while the few actions are about to delete files.

1) EXPERIMENTAL PARAMETERS

Table 3 shows the experimental parameters considered during the classification using machine learning algorithms. The core parameters that helped in increasing the accuracy of LightGBM are learning_rate, num_leavesa, and bagging

TABLE 3. Experimental Parameters of Models in Test and Train.

Model	Parameters
LightGBM	objective: "binary" metric: "auc" num_leaves : 40 learning_rate : 0.004 bagging_fraction : 0.6 feature_fraction : 0.6 bagging_frequency : 6 bagging_seed : 42 verbosity: -1 seed: 42
XGBoost	learning_rate =0.1 n_estimators=20 max_depth=3 min_child_weight=2 gamma=5 subsample=0.7 colsample_bytree=0.5 objective= 'binary: logistic' nthread=2 scale_pos_weight=2 seed=20 reg_alpha=3 num_parallel_tree=3 max_cat_to_onehot=2
AdaBoost	n_estimators=10 learning_rate=1.0 random_state=0
Random Forest	n_estimators=100 random_state=0

TABLE 4. Performance comparison of Proposed Models.

Model	Precision	Recall	F1-Score
LightGBM	0.97	0.95	0.95
XGBoost	0.8827	0.87	0.87
AdaBoost	0.88	0.86	0.86
Random Forest	0.86	0.85	0.85

frequency. For increasing the performance of the XGBoost classifier the core parameters are max_depth, learning_rate, min_child_weight, and gamma.

Table 4 shows the comparative analysis of the performance of the classification algorithms used in this paper. Lightgbm performance is higher with the highest results produced in terms of accuracy. The other applied algorithms which are XGBoost, AdaBoost, and Random Forest also performed with better accuracy than the others in the previous studies.

2) EXPERIMENTAL RESULTS

One technique to enhance the precision of a decision tree is to boost it. Each of the training datasets is given weight at first. Following the learning of the classifiers, the weights are modified such that the next classifier pays greater attention to the previously overlooked datasets. It has been seen from the results that the proposed models achieve the best accuracy on the given dataset as RF has 86%, Adaboost has 88%, XGBoost has 88.27%, and the best of all those LightGBM gives the best higher accuracy of 97%. The measurement utilized for the assessment of the proposed models is

given below.

$$Accuracy : \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (8)$$

$$Precision : \frac{(TP)}{(TP + FP)} \quad (9)$$

$$Recall : \frac{(TP)}{(TP + FN)} \quad (10)$$

$$F1 - Score : \frac{(2(Precision)(Recall))}{(Precision + Recall)} \quad (11)$$

True Positive: It shows that if predicted positively, it is a true prediction.

False Positive: It shows that if predicted positive but it is a false prediction.

True Negative: It shows that if predicted negatively and it is a true prediction.

False Negative: It shows that if predicted negatively and it is a false prediction.

Figure 12 illustrates the confusion matrix built on the random forest algorithm classification. It demonstrated the predicted values against the actual values. Random Forest classifier predicted most samples of the dataset correctly and hence it helps in improving the accuracy of the classifier.

		Actual Values	
		Positive	Negative
Predicted Values	Positive	TP 1352	FP 235
	Negative	FN 228	TN 1485

FIGURE 12. Confusion Matrix of Random Forest.

Figure 13 illustrates the confusion matrix built on the classification of the AdaBoost algorithm. It demonstrated the predicted values against the actual values. AdaBoost classifier predicted most samples of the dataset correctly and hence it helps in improving the accuracy of the classifier. The prediction results of Adaboost are better than Random Forest on the same dataset.

Figure 14 illustrates the confusion matrix built on the classification of the XGBoost algorithm. It demonstrated the predicted values against the actual values. XGBoost classifier predicted most samples of the dataset correctly and hence it helps in improving the accuracy of the classifier. The prediction results XGBoost is better than AdaBoost on the same dataset.

Figure 15 illustrates the confusion matrix built on the classification of the LightGBM algorithm. It demonstrated the predicted values against the actual values. LightGBM classifier predicted most samples of the dataset correctly and hence it helps in improving the accuracy of the classifier. The prediction results in LightGBM are better than XGBoost on the same dataset.

		Actual Values	
		Positive	Negative
Predicted Values	Positive	TP 3080	FP 270
	Negative	FN 330	TN 1320

FIGURE 13. Confusion Matrix of AdaBoost.

		Actual Values	
		Positive	Negative
Predicted Values	Positive	TP 2930	FP 390
	Negative	FN 410	TN 1270

FIGURE 14. Confusion Matrix of XGBoost.

		Actual Values	
		Positive	Negative
Predicted Values	Positive	TP 4050	FP 290
	Negative	FN 210	TN 450

FIGURE 15. Confusion Matrix of LightGBM.

In figure 16 the heatmap of the specified features of the dataset is illustrated. It is shown in Figure 16 that the feature values with dark color code represent the high value. This helps get an idea of the dataset's most important features for better classification.

3) DISCUSSION

This work applies four machine learning algorithms to classify insider attacks. Figure 17 is the graphical representation of the applied algorithms for classifying the insider attack. The best algorithm among these is LightGBM which shows the highest accuracy. These algorithms were applied to the same dataset, and their comparative classification results are shown in Figure 17.

Machine learning algorithms are vast, and all algorithms have their benefits and limitations. Different machine learning algorithms have been applied to various datasets to perform classification. In this work, one bagging and three boosting algorithms have been utilized on the same dataset to perform classification. The results show that the boosting algorithms get higher accuracy than random forest. All other algorithms are also used for the classification of different

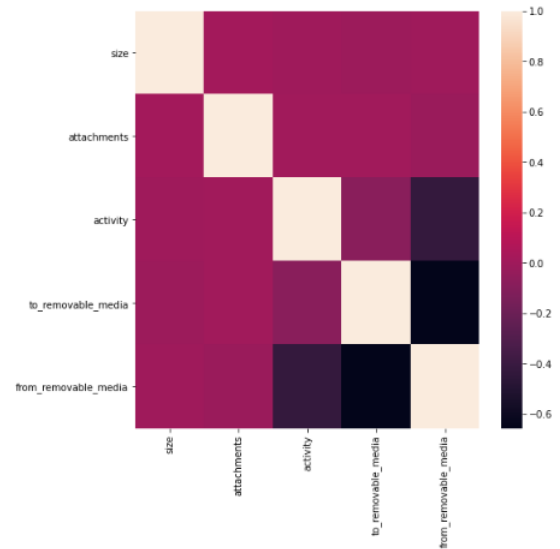


FIGURE 16. Heatmap of the specified features of dataset.

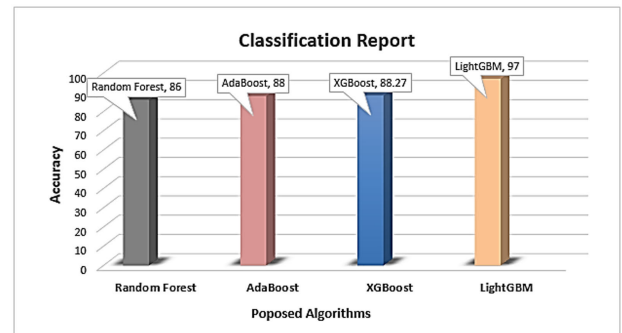


FIGURE 17. Classification Report.

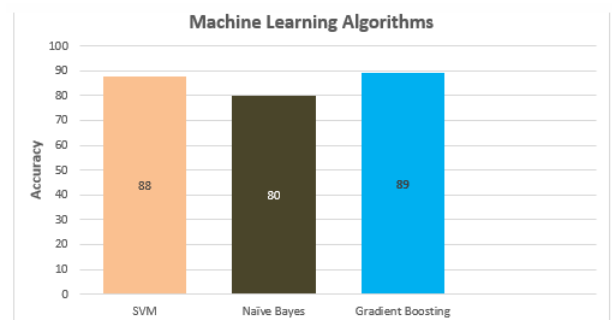


FIGURE 18. Different Algorithms for classification.

datasets. Figure 18 demonstrates multiple algorithms applied to the CERT customized dataset for classification. SVM, Naïve Bayes, and Gradient Boosting algorithms were analyzed on the same set of datasets. Naïve Bayes algorithm did not perform well while the other two algorithms give better results in terms of classifying threats.

Figure 19 is the demonstration of recall of the proposed techniques. LightGBM has the highest value among other

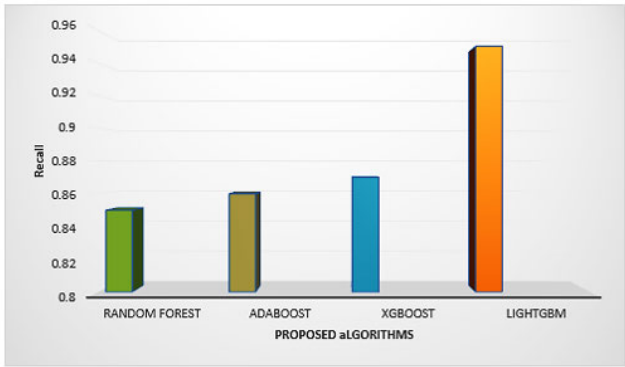


FIGURE 19. Recall score of proposed algorithms.

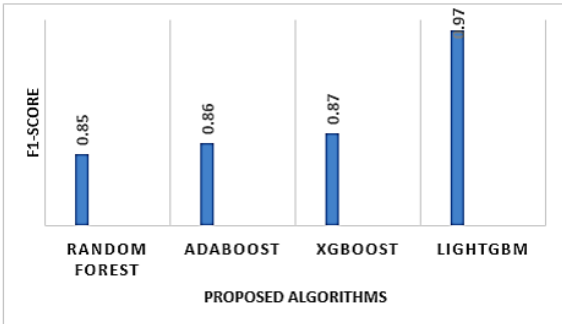


FIGURE 20. F1-Score of proposed algorithms.

techniques. The LightGBM returns the most relevant results and gets the better recall value.

The F-measure measures the accuracy of a test. It is determined using the test's accuracy and recall. Figure 20 shows the F-1 measure of the proposed techniques. It can be seen in the figure that the LightGBM algorithm got the highest value of 0.97.

4) COMPARATIVE ANALYSIS

Figure 21 shows the comparative analysis of different algorithms applied to the customized dataset. The results are compared on the basis of Recall, Precision, and F1 Score. Ensemble learning approaches came up with great results because of their best learning and classification approach. Figure 18 shows the highest accuracy value 97%, of the LightGBM algorithm.

With various hyperparameters that can be tweaked for optimal efficiencies, such as the number of leaves per tree, the learning rate, and the regularization parameters, LightGBM enables greater control over the training process. Due to its ability to efficiently understand complicated associations between features and targets in huge, high-dimensional datasets, LightGBM performs better than other algorithms at classification tasks due to all these aspects. Due to its histogram-based approach, LightGBM provides several benefits for classification problems, including a faster training speed and more efficiency. Moreover, it employs two

TABLE 5. Comparison of proposed models performance with other Algorithms.

Algorithms	Accuracy
LightGBM	97%
CNN	90%
Gradient Boosting	90%
XGBoost	89%
Adaboost	88.27%
Random Forest	86%
Logistic Regression	85%
Hidden Markov Model	83%
Adaboost + PCA	81.83%

cutting-edge methods that provide high levels of accuracy. LightGBM can train datasets as quickly as or more quickly than existing machine learning models.

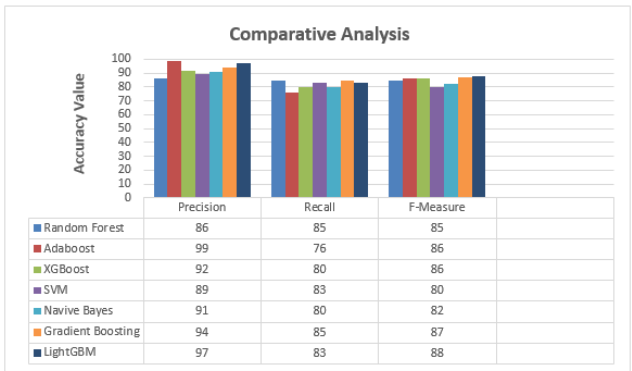


FIGURE 21. Comparative Analysis.

The proportion of false alarms or false positives produced by a model is measured by the false alarm rate, making it a crucial parameter in machine learning. False alarms can lead to unneeded interruptions or treatments, which can be dangerous in some situations. The accuracy and efficiency of machine learning models must thus be increased while simultaneously lowering the false alarm rate. Figure 22 is the graphical representation of the False Alarm Rate.

Table 5 compares literature work algorithms with the proposed methodology algorithms. The LightGBM achieved the highest accuracy among all the applied algorithms in this work and also the previous algorithms in recent studies.

IV. MITIGATION STRATEGIES FOR INSIDER ATTACKS

A crucial step in the cyber-attack chain is privilege escalation, which often includes the execution of a privilege escalation vulnerability caused by a flaw in the system, a configuration error, or insufficient access controls. The following are the countermeasures against the privilege escalation attacks:

A. SECURITY POLICY

An effective security policy should, at the very least, outline the mitigation of security threats. Including measures in your security strategy to avoid and identify misuse is one of the greatest strategies to stop insider threats. Rules for handling

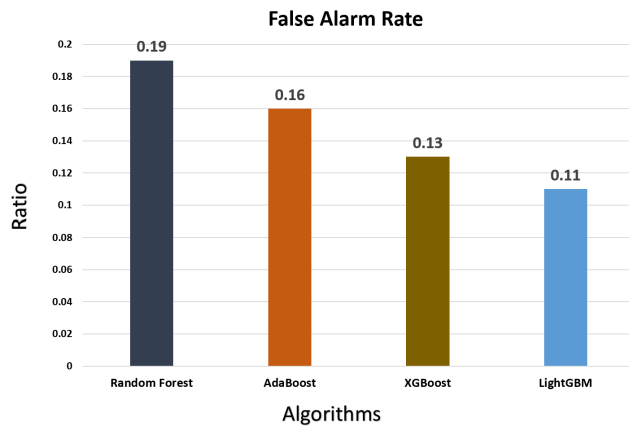


FIGURE 22. False Alarm Rate.

insider misuse investigations should also be included in the policy.

B. MULTIFACTOR AUTHENTICATION

Password-cracking technology is evolving, making hacking into an employee's computer and critical access data simpler than ever because several workers create weak credentials to access data. Must add robust multifactor authentication to the company's most critical apps. Unauthorized users will find it considerably more challenging to access sensitive data.

C. SECURE DESKTOPS

Certain services can lock down PCs throughout Corporation. Because businesses can't rely on their staff to handle all their setups with the appropriate level of responsibility, these services are quite helpful. To further assist companies in preventing dangers, these services also let companies lock off certain areas of a user's computer programs.

D. SEAL INFORMATION LEAKS

Utilize software that scans the company's policies and notifies authorities when employees break them on the network. To be sure that company staff is not revealing business secrets, there is software available that will examine the content of outgoing emails.

E. INVESTIGATE UNUSUAL ACTIVITIES

Because most businesses are too focused on looking for external dangers, it frequently happens that an employee abuses a company's trust without expecting to be held accountable. As a result, it is best to look into any suspicious behavior that occurs on the LAN of your business. Consider that there are rules governing monitoring, so inform yourself before breaking any of them.

F. BEHAVIORAL BIOMETRICS

To strengthen the defense against insider attacks, many biometrics have been deployed. Some studied methods included behavioral biometrics (e.g., typing patterns, eye and

head motions). Keystroke dynamics is a type of biometrics where insiders are continuously verified depending on their typing style. Insider keystroke patterns' variances among presses and releases were computed. The tasks being executed will be promptly prevented as soon as an abnormal typing pattern is discovered, which is regarded as a masquerader attack.

G. PHYSIOLOGICAL BIOMETRICS

Access control models' primary objective is to control access to digital assets using various authentication techniques, such as passwords, tokens, fingerprints, etc., so access may only be allowed to people who have the appropriate permissions and are approved [36], [37]. One of the main issues with access control models is generally that if a user is trusted for the duration of a session, they can abuse the capabilities they have been given without being noticed. The Intent-Based Access Control Model (IBAC) was developed to solve this issue. IBAC confirms the integrity of insiders' purpose rather than their identity, in contrast to conventional access control schemes. IBAC is based on the theory that physiological traits, including brain signals, may be used to assess the sincerity of intents and prevent insider threats.

1) FUTURE WORK

Future research will examine how temporal information is used in user behavior. Models may be able to make non-Markovian judgments if they can view numerous exemplars or remember their state.

2) MITIGATION STRATEGIES

The threat of sensitive information being accessed will be reduced by using the best practices to prevent insider attacks. These practices are as follows:

- Policies and controls must be well-documented and consistently followed.
- Install security tools and applications.
- All connections, including mobile ones, are monitored and under remote access control.
- Enforce the least privilege and task separation.

V. CONCLUSION

The malicious insider becomes a crucial threat to the organization since they have more access and opportunity to produce significant damage. Unlike outsiders, insiders possess privileged and proper access to information and resources. This paper proposed machine learning algorithms for detecting and classifying an insider attack. A customized dataset from multiple files of the CERT dataset is used in this work. Four machine learning algorithms were applied to that dataset and gave better results. These algorithms are Random Forest, AdaBoost, XGBoost, and LightGBM. Using these supervised machine learning algorithms, this paper demonstrated the effective experimental results having higher accuracy in the classification report. Among the proposed algorithms, the

LightGBM algorithm provides the highest accuracy of 97%; the other accuracy values are RF with 86%, AdaBoost with 88%, and XGBoost with 88.27%. In the future, the proposed models may increase their performance by expanding the dataset in size and diversity in terms of its features and the new trends of insider attackers to perform the attack. This may open up new research trends toward detecting and classifying insider attacks related to many fields of organization. Machine learning models are used by businesses to make credible business decisions, and improved model results lead to better judgments. The cost of mistakes can be quite high, however, this cost is reduced by improving model accuracy. ML-based research enables users to provide massive amounts of data to computer algorithms, which then evaluate, recommend, and decide using the supplied data.

APPENDIX

Sr.No.	Word	Abbreviations
1	ML	Machine Learning
2	DL	Deep Learning
3	IT	Information Technology
4	DDoS	Distributed Denial of Services
5	URL	Uniform Resource Locator
6	RF	Random Forest
7	AdaBoost	Adaptive Boosting
8	XGBoost	Extreme Gradient Boosting
9	LightGBM	Light Gradient Boosting Machine
10	LSTM	Long Short Term Memory
11	ILSTM	Improved Long Short Term Memory
12	SVM	Support Vector Machine
13	KNN	K-Nearest Neighbour
14	LR	Logistic Regression
15	CNN	Convolutional Neural Network
16	PCA	Principal Component Analysis
17	ANN	Artificial Neuron Network
18	Ms	Model Selection
19	Mt	Model Train
20	Ts	Train ,Test
21	Csv	Comma-Separated Value

REFERENCES

- U. A. Butt, R. Amin, H. Aldabbas, S. Mohan, B. Alouffi, and A. Ahmadian, "Cloud-based email phishing attack using machine and deep learning algorithm," *Complex Intell. Syst.*, pp. 1–28, Jun. 2022.
- D. C. Le and A. N. Zincir-Heywood, "Machine learning based insider threat modelling and detection," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manag. (IM)*, Apr. 2019, pp. 1–6.
- P. Oberoi, "Survey of various security attacks in clouds based environments," *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 9, pp. 405–410, Sep. 2017.
- A. Ajmal, S. Ibrar, and R. Amin, "Cloud computing platform: Performance analysis of prominent cryptographic algorithms," *Concurrency Comput., Pract. Exper.*, vol. 34, no. 15, p. e6938, Jul. 2022.
- U. A. Butt, R. Amin, M. Mehmood, H. Aldabbas, M. T. Alharbi, and N. Albaqami, "Cloud security threats and solutions: A survey," *Wireless Pers. Commun.*, vol. 128, no. 1, pp. 387–413, Jan. 2023.
- H. Touqeer, S. Zaman, R. Amin, M. Hussain, F. Al-Turjman, and M. Bilal, "Smart home security: Challenges, issues and solutions at different IoT layers," *J. Supercomput.*, vol. 77, no. 12, pp. 14053–14089, Dec. 2021.
- S. Zou, H. Sun, G. Xu, and R. Quan, "Ensemble strategy for insider threat detection from user activity logs," *Comput., Mater. Continua*, vol. 65, no. 2, pp. 1321–1334, 2020.
- G. Apruzzese, M. Colajanni, L. Ferretti, A. Guido, and M. Marchetti, "On the effectiveness of machine and deep learning for cyber security," in *Proc. 10th Int. Conf. Cyber Conflict (CyCon)*, May 2018, pp. 371–390.
- D. C. Le, N. Zincir-Heywood, and M. I. Heywood, "Analyzing data granularity levels for insider threat detection using machine learning," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 1, pp. 30–44, Mar. 2020.
- F. Janjua, A. Masood, H. Abbas, and I. Rashid, "Handling insider threat through supervised machine learning techniques," *Proc. Comput. Sci.*, vol. 177, pp. 64–71, Jan. 2020.
- R. Kumar, K. Sethi, N. Prajapati, R. R. Rout, and P. Bera, "Machine learning based malware detection in cloud environment using clustering approach," in *Proc. 11th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2020, pp. 1–7.
- D. Tripathy, R. Gohil, and T. Halabi, "Detecting SQL injection attacks in cloud SaaS using machine learning," in *Proc. IEEE 6th Int. Conf. Big Data Secur. Cloud (BigDataSecurity), Int. Conf. High Perform. Smart Comput., (HPSC), IEEE Int. Conf. Intell. Data Secur. (IDS)*, May 2020, pp. 145–150.
- X. Sun, Y. Wang, and Z. Shi, "Insider threat detection using an unsupervised learning method: COPOD," in *Proc. Int. Conf. Commun., Inf. Syst. Comput. Eng. (CISCE)*, May 2021, pp. 749–754.
- J. Kim, M. Park, H. Kim, S. Cho, and P. Kang, "Insider threat detection based on user behavior modeling and anomaly detection algorithms," *Appl. Sci.*, vol. 9, no. 19, p. 4018, Sep. 2019.
- L. Liu, O. de Vel, Q.-L. Han, J. Zhang, and Y. Xiang, "Detecting and preventing cyber insider threats: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 1397–1417, 2nd Quart., 2018.
- P. Chattopadhyay, L. Wang, and Y.-P. Tan, "Scenario-based insider threat detection from cyber activities," *IEEE Trans. Computat. Social Syst.*, vol. 5, no. 3, pp. 660–675, Sep. 2018.
- G. Ravikumar and M. Govindarasu, "Anomaly detection and mitigation for wide-area damping control using machine learning," *IEEE Trans. Smart Grid*, early access, May 18, 2020, doi: 10.1109/TSG.2020.2995313.
- M. I. Tariq, N. A. Memon, S. Ahmed, S. Tayyaba, M. T. Mushtaq, N. A. Mian, M. Imran, and M. W. Ashraf, "A review of deep learning security and privacy defensive techniques," *Mobile Inf. Syst.*, vol. 2020, pp. 1–18, Apr. 2020.
- D. S. Berman, A. L. Buczak, J. S. Chavis, and C. L. Corbett, "A survey of deep learning methods for cyber security," *Information*, vol. 10, no. 4, p. 122, 2019.
- N. T. Van and T. N. Thinh, "An anomaly-based network intrusion detection system using deep learning," in *Proc. Int. Conf. Syst. Sci. Eng. (ICSSE)*, 2017, pp. 210–214.
- G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, Mar. 2021.
- A. Arora, A. Khanna, A. Rastogi, and A. Agarwal, "Cloud security ecosystem for data security and privacy," in *Proc. 7th Int. Conf. Cloud Comput., Data Sci. Eng.*, Jan. 2017, pp. 288–292.
- L. Coppolino, S. D'Antonio, G. Mazzeo, and L. Romano, "Cloud security: Emerging threats and current solutions," *Comput. Electr. Eng.*, vol. 59, pp. 126–140, Apr. 2017.
- M. Abdelsalam, R. Krishnan, Y. Huang, and R. Sandhu, "Malware detection in cloud infrastructures using convolutional neural networks," in *Proc. IEEE 11th Int. Conf. Cloud Comput. (CLOUD)*, Jul. 2018, pp. 162–169.
- F. Jaafar, G. Nicolescu, and C. Richard, "A systematic approach for privilege escalation prevention," in *Proc. IEEE Int. Conf. Softw. Quality, Rel. Secur. Companion (QRS-C)*, Aug. 2016, pp. 101–108.
- N. Alhebaishi, L. Wang, S. Jajodia, and A. Singhal, "Modeling and mitigating the insider threat of remote administrators in clouds," in *Proc. IFIP Annu. Conf. Data Appl. Secur. Privacy*, Bergamo, Italy: Springer, 2018, pp. 3–20.
- F. Yuan, Y. Cao, Y. Shang, Y. Liu, J. Tan, and B. Fang, "Insider threat detection with deep neural network," in *Proc. Int. Conf. Comput. Sci.* Wuxi, China: Springer, 2018, pp. 43–54.
- I. A. Mohammed, "Cloud identity and access management—A model proposal," *Int. J. Innov. Eng. Res. Technol.*, vol. 6, no. 10, pp. 1–8, 2019.
- F. M. Okikiola, A. M. Mustapha, A. F. Akinsola, and M. A. Sokunbi, "A new framework for detecting insider attacks in cloud-based e-health care system," in *Proc. Int. Conf. Math., Comput. Eng. Comput. Sci. (ICMCECS)*, Mar. 2020, pp. 1–6.
- G. Li, S. X. Wu, S. Zhang, and Q. Li, "Neural networks-aided insider attack detection for the average consensus algorithm," *IEEE Access*, vol. 8, pp. 51871–51883, 2020.
- A. R. Wani, Q. P. Rana, U. Saxena, and N. Pandey, "Analysis and detection of DDoS attacks on cloud computing environment using machine learning techniques," in *Proc. Amity Int. Conf. Artif. Intell. (AICAI)*, Feb. 2019, pp. 870–875.

- [32] N. M. Sheykhanloo and A. Hall, "Insider threat detection using supervised machine learning algorithms on an extremely imbalanced dataset," *Int. J. Cyber Warfare Terrorism*, vol. 10, no. 2, pp. 1–26, Apr. 2020.
- [33] M. Idhammad, K. Afdel, and M. Belouch, "Distributed intrusion detection system for cloud environments based on data mining techniques," *Proc. Comput. Sci.*, vol. 127, pp. 35–41, Jan. 2018.
- [34] P. Kaur, R. Kumar, and M. Kumar, "A healthcare monitoring system using random forest and Internet of Things (IoT)," *Multimedia Tools Appl.*, vol. 78, no. 14, pp. 19905–19916, 2019.
- [35] J. L. Leevy, J. Hancock, R. Zuech, and T. M. Khoshgoftaar, "Detecting cybersecurity attacks using different network features with LightGBM and XGBoost learners," in *Proc. IEEE 2nd Int. Conf. Cognit. Mach. Intell. (CogMI)*, Oct. 2020, pp. 190–197.
- [36] R. A. Alsowail and T. Al-Shehari, "Techniques and countermeasures for preventing insider threats," *PeerJ Comput. Sci.*, vol. 8, p. e938, Apr. 2022.
- [37] B. Alouffi, M. Hasnain, A. Alharbi, W. Alosaimi, H. Alyami, and M. Ayaz, "A systematic literature review on cloud computing security: Threats and mitigation strategies," *IEEE Access*, vol. 9, pp. 57792–57807, 2021.



MUHAMMAD MEHMOOD received the B.S.C.S. degree in computer science from the COMSATS University Islamabad, Wah Campus, Pakistan. His research interests include ML and security.



RASHID AMIN received the M.S.C.S. and M.C.S. degrees from International Islamic University, Islamabad, and the Ph.D. degree in computer science from COMSATS University Islamabad, Wah Campus, Pakistan. He is currently an Assistant Professor with the Department of Computer Science, University of Chakwal, Pakistan. Before this, he was a Lecturer with the Department of Computer Science, University of Engineering and Technology, Taxila, Pakistan, for seven years, and

the University of Wah, Wah Cantt, Pakistan, for four years. He supervised many M.S. degree level student's thesis, and five Ph.D. degree students are working under his supervision. He has published several research papers on ML, DL, and SDN in well-reputed venues, like IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, IEEE ACCESS, *Electronics* (MDPI), and *IJACSA*. His current research interests include machine learning, deep learning, the IoMT, distributed systems, and cyber security. He is co-editing some Special Issues in some renowned journals. He is a Reviewer of international journals, such as NetSoft, LCN, IEEE GLOBECOM, FiT, IEEE WIRELESS COMMUNICATION, IEEE INTERNET OF THINGS JOURNAL, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE ACCESS, and IEEE SYSTEM JOURNAL.



MUHANA MAGBOUL ALI MUSLAM (Member, IEEE) received the B.Sc. degree in computer science from The Future University, Khartoum, Sudan, in 2003, the M.Sc. degree in computer science from the University of Khartoum, in 2006, and the Ph.D. degree in electrical engineering from the University of Cape Town, Cape Town, South Africa, in 2012. He is currently an Assistant Professor with the Department of Information Technology, Imam Mohammad Ibn Saud Islamic University.



JIANG (LINDA) XIE (Fellow, IEEE) received the B.E. degree in electrical and computer engineering from Tsinghua University, Beijing, China, the M.Phil. degree in electrical and computer engineering from The Hong Kong University of Science and Technology, and the M.S. and Ph.D. degrees in electrical and computer engineering from the Georgia Institute of Technology. She joined the Department of Electrical and Computer Engineering, The University of North Carolina at Charlotte (UNC-Charlotte), as an Assistant Professor, in August 2004, where she is currently a Full Professor. Her current research interests include resource and mobility management in wireless networks, mobile computing, the Internet of Things, cloud/edge computing, and virtual/augmented reality. She is a Senior Member of ACM. She received the U.S. National Science Foundation NSF Faculty Early Career Development (CAREER) Award, in 2010, the Best Paper Award from IEEE Global Communications Conference, in 2017, the Best Paper Award from IEEE/WIC/ACM International Conference on Intelligent Agent Technology, in 2010, and the Graduate Teaching Excellence Award from the College of Engineering, UNC-Charlotte, in 2007. She is on the editorial boards of the IEEE TRANSACTIONS ON SUSTAINABLE COMPUTING and *Journal of Network and Computer Applications* (Elsevier).



HAMZA ALDABBAS received the B.Sc. degree in computer information systems and the M.Sc. degree in computer science from Al-Balqa Applied University, Al-Salt, Jordan, in 2006 and 2009, respectively, and the Ph.D. degree in computer science and software engineering from De Montfort University, Leicester, U.K., in 2012. Since 2020, he has been an Associate Professor with the Prince Abdullah bin Ghazi Faculty of Information and Communication Technology, Al-Balqa Applied University, Jordan. Previously a Lecturer with De Montfort University, U.K., with responsibility for teaching and project supervision for the B.Sc. and M.Sc. degrees levels, from 2010 to 2012. His research interests include machine learning, security, the Internet of Things, networking, and natural language processing.

• • •