Insider Detection Using Combination of Machine Learning and Expert Policies

Buraq Almusawy and Ali A. H. Alrammahi*
Computer Science Department, Faculty of Computer Science and Mathematics, University of Kufa, Najaf, Iraq
Email: buraqn.almusawy@uokufa.edu.iq (B.A.), alia.alramahi@uokufa.edu.iq (A.A.H.A.)

***Abstract*—Today, organizations of all sizes face many difficulties in protecting their data, systems, and tools. One issue of particular concern is the insider threat. Insiders seek to use their privileges to undermine data confidentiality, validity, and availability. Any sabotage committed by someone within a company significantly harms the company's integrity, credibility, and financial profits. Automated feature extraction methods face challenges when used to classify data due to their tendency sometimes to return inaccurate results, leading to overfitting. Furthermore, analyzing irregular data requires extensive manual feature detection. We propose an algorithm that represents an expert system that detects insiders and determines their risk level as well. After that, the decisive step will be to intersect the results obtained from a classification using multiple algorithms with those obtained from the internal detection algorithm using expert rules. This research uses several classification methods that can deal with this type of data to predict the status of insiders within a computer network. The main goal of this study is to improve the accuracy and efficiency of identifying insiders within a computer network. Model performance evaluation includes important parameters such as precision, recall, and F1 score. The highest classification accuracy is obtained at 0.99, and after combining these results with the results of the proposed algorithm, the accuracy is 100%. These results highlight the remarkable ability of these models to detect internal states accurately, providing encouraging possibilities for improving cyber security within a computer network.**

***Index Terms*—machine learning, classification algorithms, synthetic minority oversampling technique, Community Emergency Response Team (CERT) dataset**

I. Introduction

Cybersecurity [1–3] is threatened by insider threats, causing security breaches, data leaks, and other crimes. These risks come from employees with valid access to the network, making them difficult to identify and prevent. In this paper, we aspire to propose a suitable method for detecting insider risks using expert rules, machine learning, and data analysis [4]. Expert rules may reveal potential risks based on human competence. However, machine learning may reveal insider threat trends in large data sets.

The motivation to detect insider threats in computer networks is based on the combination of expert rules, machine learning and data analysis from several key factors:

- *The need for effective insider threat detection*: Traditional methods are limited in their effectiveness. They may not be able to detect subtle or complex insider threat patterns. However, machine learning algorithms may learn from data and make predictions or judgments without being programmed. They can see patterns in data that may not be obvious to a human observer.
- *The importance of expert rules*: Predefined sets of conditions used to identify potential threats, can be valuable in detecting insider threats. They can provide the foundation for machine learning models and help guide their learning process.
- *Efficient data analysis*: Data analysis may reveal network activity and user behaviour, revealing insider threats. NoSQL (Not Only Structured Query Language) is a powerful modern query language that can quickly manage and consume massive amounts of data, making it ideal for real-time insider threat detection.
- *Need for a comprehensive approach*: Detecting insider risks using expert rules, machine learning, and data analysis is robust. This technology may combine the capabilities of each method to detect better and mitigate risks

In this paper we are confronted with many challenges:

- in analyzing this type of data, which consists of five groups of different sizes and types, to gain new characteristics.
- Getting high levels of accuracy from the classification algorithms that we will be using.
- The creation of a data set that is uniform and consistent.
- How to make an intersection or union of results of the proposed algorithm with results of the classification algorithm.

Data analysis may improve this method by giving context and information about network processes. The strategy in the study addresses the difficulties of detecting internal threats. These include high false positive rates, difficulty separating normal activity from malicious activity, and requirements for real-time detection [5].

The main problem of the research is that there is an increase in the number of people working within the company's computer networks. Insiders can be identified when they engage in actions normally reserved for outsiders, such as mailing a file or accessing websites that

Manuscript received February 18, 2024; revised April 9, 2024; accepted April 17, 2024.
*Corresponding author

doi: 10.18178/ijeetc.13.5.389-396

are normally restricted to access on the corporate network. Another type of insider is an employee of the organization. However, it may be difficult to identify, exactly where the issue lies.

The proposed solution uses expert rules, machine learning, and data analysis to enhance insider threat detection and reduce false positives. The classification of insider threats and their problems are also discussed in this article. It covers the latest research in this field, including machine learning algorithms, evaluation, and datasets. In conclusion, the work develops the process of detecting insider threats [6]. A revolutionary method that detects internal risks to a computer network using expert rules, machine learning, and data analysis. This method may lead to increased identification of insider threats and organization security [7].

The contributions of our paper can be summarized as follows:

- The data analysis model using NoSQL differs from SQL (Structured Query Language), as the former can analyze large amounts of data in less time.
- The model and algorithms for the combined application of expert rules and machine learning methods in the interests of detecting insider attacks differ from the existing ones by using an integrated approach to solving the problem of detecting insiders, taking into account the characteristics and properties of users, devices, applications, services, including the time parameter.
- The method of detecting insiders differs from existing ones by using the proposed model for representing huge data on insider attacks and the proposed model and algorithms for the combined use of expert rules, machine learning methods and data analysis.

The remaining sections of the paper are structured as follows: In Section II, we will review previous related work on insider detection systems. Section III will explain our methodology with all proposed algorithms. In Section IV, we will present and analyze the results obtained from the proposed algorithm and classification algorithms and finally, in Section V, we will list the most important outlines, conclusions and future works of the paper.

II. Literature Review

The identification of insider threats has consistently garnered attention throughout the years. Previous works focused on the profile of insider threats. The Community Emergency Response Team (CERT) Insider Threat Centre has produced standard guidelines to mitigate and minimize the risks of insiders within organizations.

Narayana *et al*. [8] used multiple models to classify insider risks in this study. The company aggregated employee access patterns into numerical attributes. They trained and tested on CERT. The technique was evaluated using logistic regression, decision tree, random forest, and XGBoost (eXtreme Gradient Boosting). The model performed better than pre-trained Convolutional Neural Networks (CNN) models in precision, recall, precision, and F1-Score.

Sashi *et al*. [9] presented a machine learning model using XGBoost as its foundation. The model generated the exact outcomes that were intended. Flask Micro web framework was used to develop a web form that gathers user attributes and identifies internal activity via the implementation of XGBoost.

Fisal *et al*. [10] created prediction models utilizing language analysis to assess employee risk in computer-mediated communication, notably emails. In this study, emails were analyzed by supervised machine learning. The collection comprised 24 spammers' behavioural traces over five days. Normalization and overriding with current axial models were restricted. The algorithms that were used in this work are Adaptive Boosting (AdaBoost). Naive Bayes (NB), Logistic Regression (LR), K-Nearest Neighbours (KNN), and Support Vector Machine (SVM).

Table I: Summarizing the Key Information from Each Study

No	Paper info.	Year	The approach proposed	Dataset and Accuracy
1	Narayana *et al*. [8]	2023	logistic regression, decision tree, random forest, and XGboost	CERT r4.2 1.00
2	Sashi *et a*l. [9]	2023	DT, RF, XGboost, XGBoost with Hyperparameter Tuning On Accuracy, XGBoost with Hyperparameter Tuning On Recall	CERT r4.2 98.63
3	Fisal *et al*. [10]	2020	Adaboost. NB, Logistic Regression, KNN, and SVM)), Linear Regression	TWOS 98.3
4	Ghofran *et al*. [11]	2023	RF, KNN, NB, DT, MLP, Adaboost, Adaboost Gradient Boosting, SVM, DT, Linear SVC, XGBoost, Voting Classifier	Banking Sector, E-Banking 99.99
5	Le Duc [12]	2021	LR, NN, RF, XGBoost	CERT(r4.2, r5.2, r6.2) and TWOS 98.6
6	Hall *et al*. [13]	2018	NN, NBN, SVM, RF, DT and LR	CERT r4.2 96.2

Ghofran *et al*. [11] addressed the pressing issue of banking cybercrime detection to protect customer data and avoid financial losses. This study predicted banking crimes using different classification systems. K-Nearest Neighbors (KNN), Random Forest (RF), Naive Bayes, Gradient Boosting, Multilayer Perceptron (MLP), Decision Tree (DT), AdaBoost, SVM, Linear Support Vector Machine (LinearSVC), Voting Classifier, and XGBoost algorithms were tested for binary classification in banking. It sought to increase financial cybercrime detection accuracy and effectiveness. A big dataset of

login attempts, transaction amounts, device details, and geolocation was used.

Duc [12] proposed a machine learning-based framework for insider threat detection, from data pre-processing, a combination of supervised and unsupervised learning. The framework introduces a data extraction approach allowing extraction of numerical feature vectors representing user activities from heterogeneous data, with different data granularity levels and temporal data representations, and enabling applications of machine learning. Unsupervised and supervised learning methods are used for anomaly detection to identify anomalous user behaviours that may indicate insider threats.

Hall *et al.* [13] used the CERT r4.2 dataset and machine learning classifiers to predict a malicious insider threat scenario: publishing sensitive material to Wiki leaks before leaving the organization. Combining these algorithms creates a meta-classifier with better predictive performance than individual models. The accuracy of the classifier is enhanced through boosting. The models were evaluated using a confusion matrix and Receiver Operating Characteristic (ROC) curve analysis.

It is worth noting that most previous works did not use expert rules with machine learning techniques, and this leads us to the ultimate goal of the proposed research. The literary overview and description of each work can be summarized and the classification accuracy obtained can be presented, as in Table I.

III. Methodology

The general idea of the proposed model is to detect insiders using expert policies and again using machine learning algorithms. Finally, we will perform one of the following actions: intersect the results, perform the union operation between them, or leave each algorithm to its classification, as shown in Fig. 1.

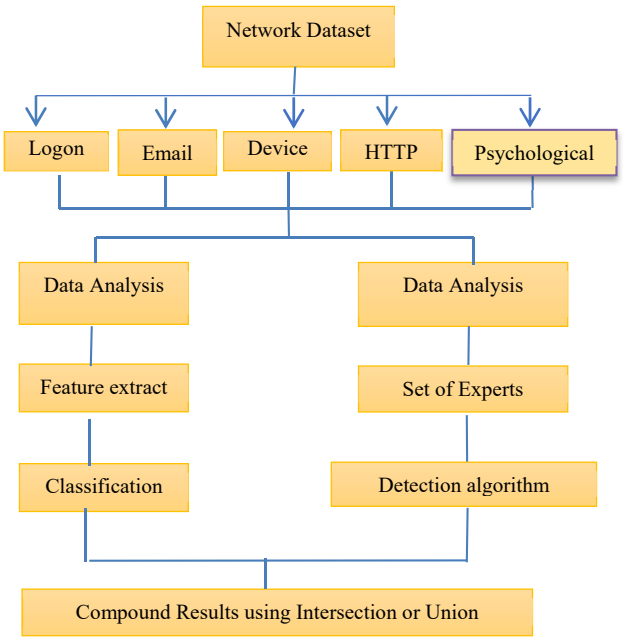

Fig. 1. Block diagram of insider detection using machine learning and expert's policies.

A. Employees Dataset

Our paper uses the CERT (Carnegie Mellon University) insider threat data collection. The main reason for choosing part r4.2 of the dataset is that 70 users represent insiders on whom some experiments can be conducted, while the rest of the sections have small numbers of insiders. Analysis was conducted using the CERT r4.2 dataset, which includes logon/logoff, email transmission, device, and Hypertext Transfer Protocol (HTTP) events, as shown in Table II. These events track the actions of 1000 employees in a company throughout a year [13].

Table II: Dataset Collections and Attributes

Collection with details	Attributes
Logon.csv (Events of logon/logoff)	id, date, user, pc, activity
Device.csv (External storage device Usage)	id, date, user, pc, activity
Email.csv (sent and received Emails)	id, date, user, pc, to, cc, bcc, form, size, attachment count, content
HTTP.csv (Visited websites)	id, date, user, PC, URL, content
Psychometric.csv (psychometric scores)	ID, user, openness, conscientiousness, extraversion, agreeableness, neuroticism

We collected 32,770,222 events from typical and deviant users. The CERT dataset also includes employee's psychometric scores, popularly known as the "Big Five personality characteristics". The psychometric.csv file contains these traits [14].

B. Insiders Threats Classification

1) Features extraction of CERT dataset

Extracting features from the data set is challenging, as five collections have unequal record numbers. So that is why we will use manual excreting using aggregations tools in NoSQL code. The main distinction between the two notions is the relational structure of SQL databases and the inclusion of foreign keys. NoSQL is immutable and does not indicate relationships. Table III displays the properties of both databases [15].

Tabel III: Main Differences between SQL and NoSQL

Property	SQL	NoSQL
The Method of Data Storage	Tabular Documents	Major Value
Data organization	Schema is predefined in SQL	Schema is dynamic in NoSQL.
Scalable	Vertically (Huge RAM, Strong Processor)	Horizontally (Extra Servers, Instances)
Language	Standard Query Language	Customized Query Language
Data Interaction	Relation Key	Embed Document
Safety	Isolated, Consistent, Transactions,	Non-existent

We will extract the features individually from each collection, and finally, we will group them depending on the file (Psychometric) [16, 17] because this file doesn't contain repeated records. The result of the data analysis will be the data set that will be used in the classification

algorithms, which consists of 1000 records with ten extracted features, these features are four new extracted as shown in Table IV, in addition to attributes of psychometric scores.

TABLE IV: NEW EXTRACTED FEATURES FROM THE COLLECTION

Features	Collection	Description
AfterEnd	Logon	How many times did the user access the network outside official working hours?
BlockedSite	HTTP	The number of blocked websites visited by the employee
CountActivity	Device	The number of times the employee used external devices during official work
No_of_Email	Email	The number of blocked Emails sent and received by the employee

2) CERT dataset augmentation

The data analysis conducted in the previous section leads to reducing the number of entries to 1000 [13], and this number sometimes leads to overfitting after the training process because this data will be divided into 80 percent for training and 20% for testing. This percentage is considered variable and does not lead to accurate results.

There are many techniques for data balancing, but the most common are Adaptive Synthetic Sampling (ADASYN) and Synthetic Minority Oversampling Technique (SMOTE). In the proposed method we will use the SMOTE approach, where synthetic samples are generated evenly throughout the feature space, while ADASYN focuses more on creating synthetic samples in regions where classification is challenging. We will use SVM classification, which works better with the SMOTE method. The SMOTE approach is used to augment the number of underrepresented instances in a machine learning dataset [18]. Employing this approach is more effective in augmenting the number of instances compared to just replicating preexisting examples. It is a statistical method used to augment the number of observations in a data collection balanced manner. The process involves creating new instances based on the current minority cases you must submit. It is important to note that this approach does not significantly alter most scenarios. The newly created instances are not just replicas of the preexisting minority class. The technique incorporates samples from each target class's attributes and its closest neighbours. This strategy will enhance the number of accessible characteristics for each class and render the examples seem more universal. In conclusion, SMOTE accepts the dataset as an input, just augmenting the proportion of the minority class in the data. The augmentation process can be explained in (1):

$$\text{New minority class sample} = \text{Minority class sample 1} + (\text{Minority class sample 2} - \text{Minority class sample1}) \times \text{rand}() \quad (1)$$

where rand () is a random number generator that generates random numbers between 0 and 1.

3) Classification algorithms

There are many classification algorithms, so we focused on specific algorithms in the proposed research because the total data analyzed now consists of 1000 entries. Moreover, the training techniques used are compatible with the small-quantity data set to give somewhat acceptable classification accuracy.

- *XGBoost*: The machine learning method belongs to the ensemble learning category. It is based on decision trees and utilizes the gradient-boosting approach to categorize various objects. This method is widely regarded as one of the most potent machine learning algorithms and demonstrates exceptional performance when applied to (small to medium-sized) structured or tabular data. The technique uses parallelization to enhance its performance by constructing decision trees. It employs regularization, a method used to mitigate the overfitting of data [19].
- *AdaBoost*: Its primary use is categorization, and the basic learner, often a decision tree with a single level, is sometimes referred to as a stump. The method uses weighted errors to construct a robust classifier by combining many weak classifiers [20].
- *SVM*: It is better for classification but may help with regression. SVM finds a hyper-plane to divide data kinds. This hyper-plane is a line in 2D. SVM plots each dataset item in an *N*-dimensional space, where *N* is the number of features/attributes. Next, identify the best data separation hyperplane. You must have gathered that SVM can only classify binary data [21].
- *NB*: This classifier uses a probabilistic approach. It applies Bayes' theorem under the premise that the presence of one characteristic in a particular category is independent of the presence of another characteristic in the same category. The joint probability of categories and phrases estimates the likelihood of specific categories. By assuming independence, it becomes possible to study the parameters for each term separately, which speeds up calculation activities. A Bayesian network comprises a structural model and a collection of conditional probabilities [22].

C. Insider Detection Using Expert Policies Algorithm

As a proactive step to detect insiders, we proposed a logical algorithm relying on expert policies to identify insiders. This algorithm takes an abnormal sample and a non-abnormal sample and enters its data for testing based on conditions set by the experts. For example, the employee accesses blocked websites, contacts external devices during official working hours, sends and receives blocked e-mails, and logs into the system outside official working hours. Based on these policies, insiders are identified, as shown in Algorithm 1.

Algorithm 1: Insider detection based on Expert's Policies

Step1: Select Sample (USER, LOGON, EMAIL, HTTP, DEVICE, PSYCHOMETRIC)
Step2: Flag=0
Step3: IF TIME OF LOGON IS OUTSIDE OF TIME WORK Then Flag=1 Go To: 7 Else Go To: Step 8
Step4: IF SENT EMAIL (BLOCKED) Then Flag=1
Step5: IF Visited website (BLOCKED) Then Flag=1

Step6: IF EXTERNAL DEVICE (ACTIVE) Then Flage=1
Step7: Insider Go to Step 9
Step8: Normal
Step9: End.

IV. Results and Analysis

We will classify the data set by the features we obtained after analyzing the data using the NoSQL language. The first step in classification using the proposed algorithms is to divide the data into two parts (80% for training and 20% for testing), and then we will use the SMOTE technique for balancing the training data part only [23]. Then, the stage of data training and prediction of results begins.

To determine the accuracy of the results we get from classification algorithms, we use the classification quality metrics as follows:

A. Accuracy

The quotient between the total number of accurate predictions and the total number of predictions produced by the model is given as

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (2)$$

B. Precision

The quotient of the number of true positives and the total number of positive predictions generated by the model is expressed as

$$\text{Precision} = \frac{TP}{(TP+FP)} \quad (3)$$

C. Recall

The proportion of correctly identified positive cases out of the total number of positive cases is represented as

$$\text{Recall} = \frac{TP}{(TP+FN)} \quad (4)$$

D. F1 Score

F1 score can be calculated by taking the reciprocal of the average of the reciprocals of accuracy and recall, given as

$$F_1 - \text{Score} = 2_{,,} \frac{\text{Precision}_{,,}\text{Recall}}{\text{Precision}+\text{Recall}} \quad (5)$$

- True Positive (TP): The number of correctly predicted instances as positive.
- False Positive (FP): The number of instances incorrectly predicted as positive.
- True Negative (TN): The number of instances correctly predicted as negative.
- False Negative (FN): The number of instances incorrectly predicted as negative.

We will get an accuracy report before and after using the SMOTE balancing technique, as shown in Table V and Table VI [24].

Table V: Evolution Report for All Classification Algorithms without the SMOTE Technique [19]

Model name	Acc	Pre	Rec	F1
XGboost	0.97	1.00	0.84	0.91
Adaboost	0.97	0.96	0.84	0.90
SVM	0.94	0.88	0.71	0.79
NB	0.91	0.88	0.48	0.62

Table VI: Evolution Report for All Classification Algorithms with SMOTE Technique

Model name	Acc	Pre	Rec	F1
XGboost	0.99	1.00	0.94	0.97
Adaboost	0.98	0.97	0.94	0.95
SVM	0.95	0.86	0.81	0.83
NB	0.91	0.89	0.52	0.65

We can also clarify the discrepancy between the classification methods used before the balancing process using the SMOTE approach, as in Fig. 3, which shows the somewhat low evaluation rates, especially the classification accuracy, which is 97% [25].

We notice a relative increase in accuracy in the classification results after using the SMOTE balancing approach.

This increase occurs when using the XGBoost algorithm, where the accuracy reached 99%, as shown in Fig. 4.

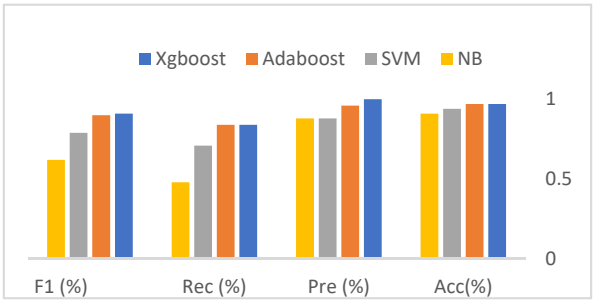

Fig. 3. Chart of the variance of evaluation accuracy of classification algorithms without SMOTE.

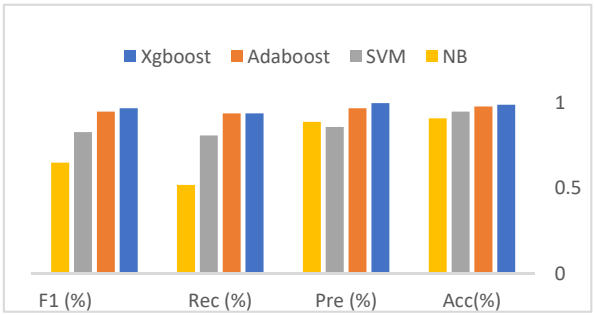

Fig. 4. Chart of the variance of evaluation accuracy of classification algorithms with SMOTE.

The confusion matrix can also confirm the accuracy of the results obtained from the classification algorithms, as shown in Table VI [26].

Based on the stated functional requirements, the developed system must combine several successful approaches to detecting insiders. This can be achieved by combining different insider detection algorithms as follows.

Table VI: Confusion Matrix of Classification Algorithms

Model name	Confusion matrix	
XGboost	169(TP) 2 (FP)	0(FN) 29(TN)
Adaboost	168(TP) 2 (FP)	1 (FN) 29 (TN)
SVM	165(TP) 6 (FP)	4 (FN) 25(TN)
NB	167(TP) 15(FP)	2 (FN) 16(TN)

Since each algorithm has the same input data - attributes of user behaviour, and the output - identifiers of detected insiders, it is possible to execute them in parallel by combining the results in one of the following ways. There are four most common ways of combining results from the perspective of working with sets (the last two of which can be considered degenerate but necessary for consideration):

- Association—the result of the complex includes insiders detected by any of the algorithms.
- Intersection—the result of the complex includes insiders detected by both algorithms simultaneously.
- Only the first—the result of the complex includes insiders detected only by the first of the algorithms.
- Only the second—the result of the complex includes insiders detected only by the second of the algorithms.

A graphical interpretation of the combination methods is shown in Fig. 5 (the dotted red line indicates the result of the combination).

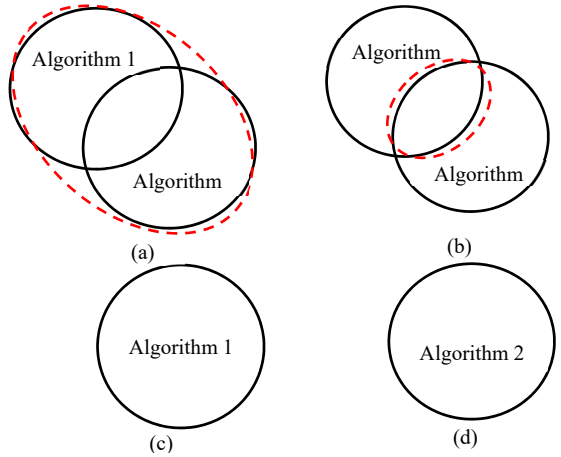

Fig. 5. (a) Union, (b) Intersection, (c) Algorithm 1, and (d) Algorithm 2.

The choice of one of the calculation formulas associated with the appropriate method should show the best values for measures of the quality of system operation. Such an informed choice will be made by using appropriate experimental evaluation.

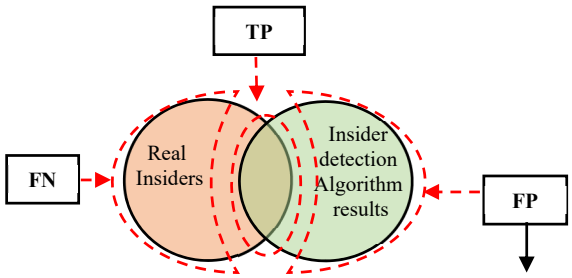

Fig. 6. Graphic interpretation of performance metrics of the insider detection algorithm.

Schematically, quality measures TP, TN, FP, and TN of the work of any insider detection algorithm can be presented graphically as follows in Fig. 6.

In Fig. 6, we exhibit two areas of insiders: those operating in the Computer Network (CN) (red circle) and those detected by the algorithm (green circle); the entire white area around the circles corresponds to legitimate users. Thus, the intersection of red and green circles means the correct operation of the algorithm (measure TP), and the absence of circles (white area) means the correct identification of legitimate users (TN) by the algorithm; at the same time, some real insiders were not detected by the algorithm (FN or type II error - missing a target), and the algorithm mistakenly counted some legitimate users as insiders (FP or type I error - false alarm) [27]. This block is general and thesis work.

Based on the above, we propose the following functional structure of a complex of algorithms (CA) for detecting insiders in a CN, consisting of two algorithms (A_1 and A_2) combined using one of the methods mentioned above. The formal recording of the complex algorithms has the following form:

$$K_A\{A_1 \oplus A_2\}, \oplus \in \{I, II, \vee, \wedge\} \quad (6)$$

where $\oplus$ is combination operations, I is the result of the complex includes insiders detected only by the first of the algorithms, II is the result of the complex includes insiders detected only by the second of the algorithms, $\vee$ is the result of the complex includes insiders detected by any of the algorithms, $\wedge$ is the result of the complex's work includes insiders detected by both algorithms simultaneously.

Let's compose a complex of the following algorithms: as the first one, we will take the algorithm based on expert rules, described in algorithm 1, and as the 2nd one, we will take the well-known algorithm based on machine learning methods, which may have several combinations according to the selected classifiers. Thus, the algorithm in Fig. 3 will be understood as complex - based on expert rules and machine learning methods. Combining algorithms and selecting the best classifier aims to reduce errors of the I and II types.

The combined use of algorithms can be presented as a model, shown graphically in Fig. 7.

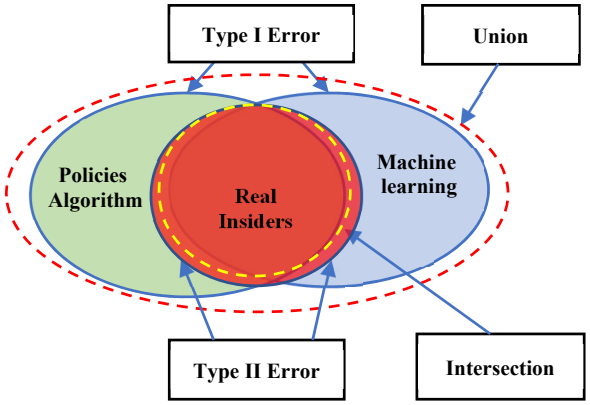

Fig. 7. Graphic interpretation of the model for combining algorithms for detecting insiders in computer networks.

The model reflects the relationship of the following entities, presented in Fig. 7 using three elliptical areas associated with insiders, where the red area represents real insiders in the CN, the green area represents the insiders detected by an algorithm based on expert polices and the blue area represents insiders detected by an algorithm based on machine learning.

Each of the green and blue areas in Fig. 7 corresponds to a separate result of each of the algorithms, the union of these areas:

- Combining the results of the work of algorithms, intersecting according to the intersection of the results of their work. The intersection of the red area with the result of one of the methods of combining algorithms corresponds to the TP measure, that is, correctly identified insiders, and the area outside the red, green and blue areas corresponds to the TN measure, i.e., users who are not correctly classified as insiders. By analogy, green or blue areas that do not intersect with red correspond to the FP measure (or type I error)
- An insider is not detected by the algorithm, and the red area, which does not intersect with the result of the work of one of the methods of combining algorithms, corresponds to the FN measure (or type II error)
- The insider's omission by the complex.

The ideal operation of a set of algorithms (that is, the result of *I*) will be the situation when either the intersection or the union of the algorithms coincides with the real insiders (that is when FP and FN are identical to 0). However, this situation is rarely achievable because any of the algorithms can both miss some insiders and identify them incorrectly.

The combination of algorithms consists of 3 stages, as shown in algorithm 2.

Algorithm 2: Combination Algorithm

Step 1: Input of data on the network activity, which is then transmitted to the input of algorithms based on expert policies and machine learning methods.

Step 2: Merging the results of the two algorithms using different methods (the second formula) mentioned in Fig. 7.

Step 3: Output a set of algorithms' results for each variation.

V. Conclusion and Future Works

In this paper, we address insider detection using multiple analysis and classification methods to increase the accuracy of insider detection. To obtain new features, the data set was analyzed using a query language different from the one used always, NoSQL. The dataset contains 1,000 online users with multiple user influences over a 12-month, generating over 32 million entries. In addition, we proposed an algorithm to detect insiders and calculate their risk score based on a set of expert policies. Finally, we proposed a new approach to combine the results of the two algorithms by interrupting the results of the best classification algorithm, which gave an accuracy of 99% XGBoost, with the proposed insider detection algorithm, unifying their results or selecting each one separately to reduce the error rate when incorrect detection of Insiders. From the results of this research, we plan to build an integrated detection system that reduces the risks of corporate insiders. The following can be highlighted as future work First, the expansion of expert polices in CN. Second, increase the number of extracted features by using complex mathematical models capable of tracking changes that occur in user behaviour. Third, it is possible to expand the number of machine learning algorithms and make them work in parallel to reduce possible errors.

Conflict of Interest.

The authors declare no conflict of interest.

Author Contributions

Buraq Almusawy, the main author who wrote all the articles, collected all the necessary information and presented it for easy understanding. Ali A.H. Alrammahi audited and reported on relevant research articles and guided the scope and focus of the review.

Funding

This work supported by the Iraqi Ministry of Higher Education (MOHESR) through the Basic Research Grants Program.

Acknowledgement

The primary researcher would like to thank the supervisor who followed up on writing and revising the research. In addition, we would like to thank everyone who contributed to providing information to complete this research.

References

[1] I. H. Sarker, A. Kayes, S. Badsha, H. Alqahtani, P. Watters, and A. Ng, "Cybersecurity data science: An overview from machine learning perspective," *Journal of Big data,* vol. 7, no. 41, pp. 1–29, 2020, doi:https://doi.org/10.1186/s40537-020-00318-5

[2] J. Kosseff, "Defining cybersecurity law," *Iowa L. Rev.,* vol. 103, 985, 2017.

[3] D. Craigen, N. Diakun-Thibault, and R. Purse, "Defining cybersecurity," *Technology Innovation Management Review,* vol. 4, no. 10, pp. 13–21, 2014.

[4] D. Le, "Machine learning based framework for user-centered insider threat detection," Dalhousie University Halifax, Nova Scotia, Canada, Aug. 2021.

[5] K. Veena, K. Meena, Y. Teekaraman, R. Kuppusamy, and A. Radhakrishnan, "C SVM classification and KNN techniques for cyber crime detection," *Wireless Communications and Mobile Computing,* vol. 2022, 3640017, 2022. doi:https://doi.org/10.1155/2022/3640017

[6] T. Al-Shehari and R. A. Alsowail, "An insider data leakage detection using one-hot encoding, synthetic minority oversampling and machine learning techniques," *Entropy,* vol. 23, no. 10, #1258, 2021.

[7] S. Kumari, D. Kumar, and M. Mittal, "An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier," *International Journal of Cognitive Computing in Engineering,* vol. 2, pp. 40–46, 2021. doi: https://doi.org/10.1016/j.ijcce.2021.01.001.

[8] T. K. Rao, N. Darapaneni, A. R. Paduri, A. Kumar, and G. Ps, "Insider threat detection: Using classification models," in *Proc. the 2023 Fifteenth Int. Conf. on Contemporary Computing*, 2023, pp. 307–312.

[9] S. K. Mamidanna, C. Reddy, and A. Gujju, "Detecting an insider threat and analysis of XGBoost using hyperparameter tuning," in *Proc. 2022 Int. Conf. on Advances in Computing, Communication and Applied Informatics*, 2022, pp. 1–10.

[10] F. Janjua, A. Masood, H. Abbas, and I. Rashid, "Handling insider threat through supervised machine learning techniques," *Procedia Computer Science,* vol. 177, pp. 64–71, 2020.

[11] G. M. Anis, A. E. Aboutabl, and A. Galal, "Machine learning for detecting cybercrime in the banking sector," *Journal of Southwest Jiaotong University,* vol. 58, no. 5, pp. 785–799, 2023.

[12] D. Le, "Machine Learning based Framework for User-Centered Insider Threat Detection," PhD dissertation, Univ. of. Dalhousie Halifax, Nova Scotia, 2021.

[13] X. Kan, Y. Fan, J. Zheng, C.-H. Chi, W. Song, and A. Kudreyko, "Data adjusting strategy and optimized XGBoost algorithm for novel insider threat detection model," *Journal of the Franklin Institute*, vol. 360, no. 16, pp. 11414–11443, 2023.
[14] M. Dosh, "Detecting insider threat within institutions using CERT dataset and different ML techniques," *Periodicals of Engineering and Natural Sciences*, vol. 9, no. 2, pp. 873–884, 2021.
[15] M. Z. Khan, F. U. Zaman, M. Adnan, A. Imroz, M. A. Rauf, and Z. Phul, "Comparative case study: An evaluation of performance computation between SQL and NoSQL database," *Journal of Software Engineering*, vol. 1, no. 2, pp. 14–23, 2023.
[16] R. Yousef, M. Jazzar, A. Eleyan, and T. Bejaoui, "A machine learning framework & development for insider cyber-crime threats detection," in *Proc. 2023 Int. Conf. on Smart Applications, Communications and Networking*, 2023, pp. 1–6.
[17] D. Ge, S. Zhong, and K. Chen, "Multi-source data fusion for insider threat detection using residual networks," in *Proc. 2022 3rd Int. Conf. on Electronics, Communications and Information Technology*, 2022, pp. 359–366.
[18] B. Bin Sarhan and N. Altwaijry, "Insider threat detection using machine learning approach," *Applied Sciences*, vol. 13, no. 1, 259, 2022.
[19] D. C. Le, N. Zincir-Heywood, and M. I. Heywood, "Analyzing data granularity levels for insider threat detection using machine learning," *IEEE Trans. on Network and Service Management*, vol. 17, no. 1, pp. 30–44, 2020.
[20] A. Hussain, M. Asif, M. B. Ahmad, T. Mahmood, and M. A. Raza, "Malware detection using machine learning algorithms for windows platform," in *Proc. Int. Conf. on Information Technology and Applications*, 2022, pp. 619–632.
[21] P. Jindal, S. Parikh, R. Sikka, S. R. Alatba, S. Babu, and G. Sriramakrishnan, "Analyzing the differences between SVM and Naive Bayes for feature extraction," in *Proc. 2023 3rd Int. Conf. on Advance Computing and Innovative Technologies in Engineering*, 2023, pp. 775–778.
[22] A. Bilen and A. B. Özer, "Cyber-attack method and perpetrator prediction using machine learning algorithms," *Peer J. Computer Science*, vol. 7, e475, 2021.
[23] Z. Chen and X. Ren, "An efficient boosting-based windows malware family classification system using multi-features fusion," *Applied Sciences*, vol. 13, no. 6, 4060, 2023.
[24] R. Ch, T. R. Gadekallu, M. H. Abidi, and A. Al-Ahmari, "Computational system to classify cyber crime offenses using machine learning," *Sustainability*, vol. 12, no. 10, 4087, 2020.
[25] J. Lever, "Classification evaluation: It is important to understand both what a classification metric expresses and what it hides," *Nature methods*, vol. 13, no. 8, pp. 603–605, 2016.
[26] W. Hong, J. Yin, M. You *et al.*, "A graph empowered insider threat detection framework based on daily activities," *ISA Transactions*, vol. 141, pp. 84–92, 2023. doi: https://doi.org/10.1016/j.isatra.2023.06.030.
[27] A. Fatima, T. A. Khan, T. M. Abdellatif *et al*., "Impact and research challenges of penetrating testing and vulnerability assessment on network threat," in *Proc. of 2023 Int. Conf. on Business Analytics for Technology and Security*, 2023, doi: 10.1109/ICBATS57792.2023.10111168.

Buraq Almusawy received the a bachelor's degree in computer science from the University of Babylon. She is currently working as a master's degree student in graduate studies. Her research interests include data analysis, cyber security and computer networks.

Ali A. H. Alrammahi received master degree in information technology from Dr Babasaheb Ambedkar Marathwada University in Aurangabad, India, and Ph.D. degree in data mining from Tambov State Technical University, Russia, in 2022. Currently, he work at the University of Kufa in Najaf, Iraq. His research interests include data analysis and data science.