Enhancing IoT Security Through User Categorization and Aberrant Behavior Detection Using RBAC and Machine Learning

Alshawwa Izzeddin A O[1], Nor Adnan Bin Yahaya[2], Ahmed Y. mahmoud[3]
University Malaysia of Computer Science and Engineering, (UNIMY), Malaysia[1, 2]
Faculty of Engineering and Information Technology, Al-Azhar University-Gaza, Palestine[3]

***Abstract*—The proliferation of Internet of Things (IoT) technology in recent years has revolutionized several industries, providing customers with reliable and efficient IoT services. However, as the IoT ecosystem grows, attention has switched away from straightforward user access to the crucial topic of security. Among others, there is a need to categorize users according to the actions they conduct as well as according to aberrant user behavior. By utilizing Role-Based Access Control (RBAC) and merging the categorization of access rights with the identification of aberrant behavior, access points to the Internet of Things will be strengthened in terms of security and dependability. A system is proposed to identify security flaws and prompt rapid remediation, with the incorporation of a classification of aberrant user behaviors which, in turn, offers a thorough defense against any outside threats. Three classification methods which are Support Vector Machine (SVM), Local Outlier Factor (LOF), and Isolation Forest (IF), were utilized in the study and their accuracy were compared. The results demonstrate the effectiveness of machine learning approaches using a dataset for IoT users, achieving high accuracy in identifying anomalous user behavior and enabling prompt implementation of necessary actions.**

Keywords*—Machine learning; classification; SVM; LOF; IF classification methods; aberrant user behavior; Role-Based Access Control (RBAC); IoT user dataset and user categorization

I. Introduction

A new age of linked devices has arrived because to the Internet of Things (IoT) technology's explosive growth in recent years [1]. This technology has revolutionized several sectors and made IoT services more readily available to customers. The seamless user experience has become less important as the IoT ecosystem grows and more focus is being placed on security, a crucial issue. It has become crucial to guarantee the security and reliability of IoT access points in order to preserve sensitive information and safeguard against potential attacks [2].

IoT gadgets mostly shapes our daily life. Still, as the IoT ecology develops security issues do surface. The large attack surface created by the number of linked devices calls for strong security rules to guard IoT systems and data [3-5].

Among the most urgent security issues of the Internet of Things (IoT) is unauthorized access. Conventional security systems find it difficult to stop illicit activities considering the availability of more devices and users. Under this idea, managing the access to Internet of Things systems benefits from Role-Based Access Control (RBAC). RBAC greatly lowers unauthorized access and breaches by assigning users roles in line with their responsibilities, therefore enabling people to act in line with their jobs [6-8]. Though its shortcomings, RBAC offers a strong basis for access control. Besides, one must be knowledgeable about aberrant user behavior. As IoT devices grow, the identification of abnormal activity and hand-held surveillance become ever difficult. Little changes in user activity enable hostile actors to target real-time threat assessment with challenges. Behavior classification and RBAC help IoT system security to develop dynamically. This approach helps the system to control access depending on user obligations and notify it to any unusual user conduct, therefore offering extra security [9].

Using machine learning algorithms, especially SVM, LOF, and Isolation Forest, the suggested approach detects and labels aberrant user activity in Internet of Things applications. These machines are amazing in their capacity to spot anomalies, handle massive volumes of data, and find minute trends. These machine learning approaches enable the Internet of Things' (IoT) security architecture to automatically detect threats, therefore enabling quick reactions and lessening the effect of unwanted conduct [10,11].

We want to satisfy the growing demand for IoT-based proactive security solutions. Given the growing complexity of cyberthreats, conventional approaches fall short. The more thorough, flexible, and effective way in which RBAC and machine learning-based behavior classification can enhance IoT security is shown in this work. Strict access control rules and real-time anomaly pattern detection guarantee IoT systems' dependability and trustworthiness, hence improving security [12-15].

This paper suggests a complete strategy that combines Role-Based Access Control (RBAC) with the categorization of user behavior to increase IoT security in order to address these security issues [16, 17]. The suggested method intends to strengthen security measures and maintain the integrity of IoT services by categorizing users based on their individual behaviors and spotting aberrant behavior patterns.

The categorization of abnormal user behavior is a crucial component of the suggested system in order to quickly find and fix any possible security problems [18]. By using a proactive approach, the system is equipped with strong defenses against external attacks and intrusions, enabling prompt risk mitigation and remediation.

The Role-Based Access Control (RBAC) framework has appeared in the IoT area as a result. Well-defined roles, permissions, and access control policies enable RBAC. Users are assigned roles, such as "Administrator," "Device Owner," or "Sensor Data Analyst," which specify their duties. Roles are given specific actions by means of permissions, such as "Read Sensor Data" or "Update Firmware". The roles that can carry out particular activities on particular resources are defined by access control policies. IoT security and dependability can be greatly improved by utilizing RBAC and integrating access rights categorization with anomaly detection. This strategy strengthens the security of IoT systems by ensuring users only access functions that are consistent with their roles and responsibilities.

The study extensively examines and makes use of three well-known and effective classification methods: Support Vector Machine (SVM), Local Outlier Factor (LOF), and Isolation Forest (IF) in order to obtain accurate user behavior categorization [19]. Each of these approaches has certain benefits and skills that make them suitable for identifying and analyzing the complex user behavior patterns seen in the Internet of Things (IoT) environment.

With its ability to handle both linear and non-linear data, the Support Vector Machine (SVM) method is well recognized [20]. It is the best contender for differentiating between typical user behavior and abnormal activity since it can efficiently build appropriate hyper-planes to divide various kinds of data. SVM is a useful tool for identifying and reporting suspicious events in the IoT environment since it learns from labelled data during the training phase and can effectively categorize new instances with high accuracy.

The study uses the Local Outlier Factor (LOF) algorithm [21], which is excellent at spotting local deviations and abnormalities in data, in addition to SVM. LOF is an unsupervised learning technique, in contrast to SVM, which relies on labelled data during training. Instead, LOF determines the density-based separation between data points' neighbors, allowing it to spot outliers and anomalous user behavior that might not fit the larger trends shown in the IoT user dataset. Due to its unsupervised nature, LOF is highly good in identifying new and uncommon anomalous actions, which improves the categorization process as a whole. Additionally, the Isolation Forest (IF) algorithm, which offers a unique and effective method for anomaly identification, is used in the study [22]. Through the use of random partitioning, IF isolates outliers into distinct trees. An observation is more likely to be an anomaly if fewer partitions are required to isolate it. This isolation approach may be used by IF to effectively identify and categorize atypical user behavior, especially when working with massive datasets that are typical of IoT contexts.

The study carefully compares the performance of these categorization algorithms utilizing a large and varied dataset made up of different Internet of Things users in order to assess how successful they are. The collection is carefully chosen to contain a wide range of user behaviors, from commonplace tasks to possible security violations. The study may evaluate the algorithms' performance in appropriately recognizing and categorizing various user behaviors by putting them through this varied dataset.

The proposed structure is improved by including these categorization algorithms to improve the overall security posture of Internet of Things applications. The results of this study provide useful insights for enhancing the security and reliability of IoT services as the IoT ecosystem continues to grow and change. This integrated strategy has considerable promise in establishing a safer and more resilient IoT ecosystem for organizations, people, and society at large by anticipating possible security risks and offering a strong defense against unauthorized access. The overall security posture of Internet of Things applications is improved by including these categorization methods within the suggested architecture. The research's findings offer important new perspectives on how to improve the security and reliability of IoT services as the IoT ecosystem continues to grow and change. This integrated strategy holds great promise for creating a more secure and resilient IoT environment for organizations, people, and society at large by anticipating possible security risks and offering a strong defense against unauthorized access.

This study will go into the design of the suggested framework, the implementation of role-based access control and user behavior classification, as well as the assessment of the selected classification techniques, in the parts that follow. This study intends to add to the ever-expanding body of research in IoT security and pave the way for more dependable and secure IoT services in the future by illuminating the applicability and efficacy of this comprehensive strategy.

The rest of the paper is organized as follows. Section II introduces the terminologies and concepts that will be utilized throughout the study. Section III examines related work, emphasizing existing approaches and their applicability to IoT security and user behavior classification. Section IV describes the experimental setup, which includes API access patterns, feature engineering, and the categorization system. Section V describes the dataset used in the study, including its characteristics and preprocessing methods. Section VI describes how the proposed system would be implemented, including the incorporation of machine learning techniques. Section VII summarizes the findings and examines their implications for IoT security. Finally, Section VIII summarizes the findings and proposes areas for further research to improve IoT system reliability and security.

II. Terminology

In this section, we provide a comprehensive overview of the concepts and terminologies that will be utilized throughout the paper.

Machine Learning: It is a subset of artificial intelligence (AI) consisting of automatically improving algorithms driven by data use and experience.

Classification: The technique of data point category or class prediction inside datasets.

Support Vector Machine (SVM): A classifier and regression task supervised learning algorithm.

Local Outlier Factor (LOF): An unsupervised anomaly detection system spotting local deviations of data points from their neighbors.

Isolation Forest (IF): An unsupervised machine learning method based on data point isolation intended for anomaly detection.

Aberrant User Behavior: Acts or habits that greatly vary from expected or typical user behavior.

Role-Based Access Control (RBAC): A method of control of computer or network resource access depending on user responsibilities.

IoT User Dataset: A set of facts regarding user interactions and actions in Internet of Things systems.

User Categorization: The arrangement of users into predetermined groups according to their roles or behavior inside a system.

IoT Security: Internet of Things device and network protection against cyberattacks and illegal access.

III. Related Work

IoT has gained immense popularity due to its ability to generate vast amounts of data from interconnected devices. Analysing and categorizing this data is crucial for extracting meaningful insights and making informed decisions. Unsupervised classification, which involves grouping data points without predefined labels, becomes essential in scenarios where labelled training data may be limited or costly to obtain. The review aims to shed light on the performance, strengths, and limitations of different algorithms when applied to the challenging task of classifying user-generated data from Internet of Things (IoT) devices.

In categorising this data, unsupervised classification algorithms are essential for understanding user behaviour, preferences, and anomalies. The Support Vector Machine (SVM), Local Outlier Factor (LOF), and Isolation Forest (IF) are three well-known classification techniques that will be compared in this literature study.

In the context of the Internet of Things (IoT), this paper explores the difficulty of identifying outliers. It examines how the IoT's unique properties, such as wireless communication openness and resource constraints, render standard methodologies inappropriate. The research carefully examines new machine learning-based outlier identification methods in order to solve this. The IoT ecosystem has a variety of elements that might cause outlier data, such as environmental impacts, sensor errors, hostile assaults, and event-related abnormalities. According to the machine learning algorithms used, the article categorises outlier detection methods into clustering, classification, dimension-reduction, and hybrid approaches. Researchers and practitioners may better traverse the challenges of outlier identification in IoT contexts by knowing the capabilities and constraints of various techniques [23].

The article discusses the growing security issues brought on by technical development and the expansion of the internet. It draws attention to the increase in cyber-attacks and the demand for strong security measures. To preserve information security and stop shady network activity, intrusion detection systems (IDS) are essential. In order to identify and classify security risks, the article primarily focuses on the integration of Machine Learning (ML) methods into IDS. The paper offers a comparative examination of several ML algorithms utilized inside IDS for a variety of applications, including big data, smart cities, fog computing, and the Internet of Things (IoT). It emphasizes how crucial these algorithms are to boosting security throughout various fields [24].

Reflections, interference, and ambient conditions all have an impact on Wi-Fi signals, resulting in irregular and aberrant RSS values that make exact localization difficult. The study suggests an outlier identification approach called "iF_Ensemble" that combines supervised, unsupervised, and ensemble machine learning techniques to handle this problem. The suggested iF_Ensemble approach uses the isolation forest (iForest) unsupervised learning methodology to categories RSS data as normal or abnormal and locate outliers. Additionally, the data is subjected to specific applications of supervised learning techniques as support vector machine (SVM), K-nearest neighbour (KNN), random forest (RF), and elliptic envelop. Their outcomes, meanwhile, are judged inadequate. An ensemble learning technique known as stacking is presented to improve outlier detection [25].

In order to improve security, machine learning (ML) and deep learning (DL) approaches are suggested in this paper's discussion of security issues in Internet of Things (IoT) networks. IoT networks are susceptible to cyber-attacks because they are made up of linked, dispersed embedded units with little processing power. The article addresses how the peculiarities of IoT networks have prevented existing cryptography methods from adequately addressing security and privacy issues. The security needs, possible attack vectors, and current security solutions for IoT networks are all thoroughly reviewed by the writers. The study also provides a number of ML and DL approaches that may be used to address various security issues in IoT networks. Utilizing these clever methods will allow for greater privacy and security features [26].

IV. Experimental Setup

Users utilise APIs to access the application while doing so. Users can access a certain business logic using a particular set of APIs. Malicious users occasionally try to access APIs in a way that leads to a drastically different order of APIs than do benign users. There may be several sequences of API calls depending on the business logic being accessed, and when they are combined, they form an API call graph for that user. Numerous users create exact or comparable API call graphs when there are hundreds of users. In these situations, users are grouped into a single cluster with the same graph shared by all of them. Each graph in these clusters was manually categorised as an outlier or a normal graph, and the results are presented in the classification column.

The study dataset consists of user-generated API call graphs, which are created when users utilise a set of APIs to access particular business logic. The dataset consists of numerous components, including session characteristics, IP parameters, temporal dynamics, behaviour classifications, and API access patterns. To assure data quality, we collected the dataset from Kaggle and prepared it thoroughly. This procedure involved choosing key characteristics, carrying out technical activities, and handling any missing data.

Examples of Features:

- Temporal Dynamics: Patterns in API usage across time.
- API Access Patterns: The order and frequency of API requests.
- Session Characteristics: Information regarding user sessions, including duration and frequency.
- IP Attributes: Information about the IP addresses used.
- Behaviour Classification: The classification of user behaviour as normal or abnormal based on API call sequences.

The Support Vector Machine (SVM), Local Outlier Factor (LOF), and Isolation Forest (IF) algorithms are used in this experiment to categorise API call graphs into normal and outlier categories. The dataset utilised in this study consists of API call graphs that users create when they use an API to access a particular piece of business logic. The main goal is to precisely pinpoint possibly fraudulent users whose API call graphs show notable departures from good user behaviour. To do this, the deviations have been carefully categorised as either outliers or normal graphs, creating a trustworthy baseline for assessment.

Preprocessing the data is the first step in the experiment, during which the API call graphs are appropriately converted into a numerical representation to act as features in the classification procedure. In order to reduce biases during evaluation, the dataset is then split into training and testing sets, with a balanced distribution of normal and outlier graphs in each set.

An SVM classifier is developed for SVM classification utilizing the relevant libraries and hyper parameters. Using criteria like accuracy, precision, recall, and F1-score, the model's performance is assessed on the testing set after it has been trained on the labelled training set. In addition, a visual inspection of the SVM model's decision boundary is carried out to learn more about how well it can distinguish between normal and outlier classes.

Fig. 1 explains the LOF technique is then used to find regional outliers in the dataset in the next stage. In the testing set, LOF is used to categorize API call graphs as either normal or outlier, and its effectiveness is assessed and contrasted with the SVM method.

The Isolation Forest technique is also included to find data abnormalities. In order to distinguish between typical graphs and outliers, the IF algorithm is used to categorize API call graphs in the testing set. Its performance is carefully evaluated and compared to the results of both SVM and LOF.

The SVM, LOF, and IF algorithms are thoroughly analyzed and compared in this experiment so that we can decide which approach is best for precisely detecting outliers within the user population.

The study further clarifies each algorithm's advantages and disadvantages in relation to the particular features of the dataset. Finally, statistical significance tests may be carried out to see whether the three algorithms' performances differ significantly from one another. By identifying potentially harmful activity among users who are visiting the application's APIs, the investigation's findings are meant to offer useful insights into how well these categorization approaches might improve security measures.

V. Dataset

Users interact with the program by using APIs when they access it. Access to various business logic is made possible via a certain sequence of APIs. Malicious users occasionally alter API access, resulting in a different sequence of APIs from those used by authorized users. Numerous API call sequences may occur, generating an aggregated API call graph for one user, depending on the accessible business logic. Multiple users in situations with multiple users generate the same or comparable API call graphs. In these situations, users are gathered into a single cluster using a graph clustering method, sharing a single graph. Manual examination of these clusters resulted in the categorization of each graph as either a normal or an outlier, and the results are shown in the classification column. One may compare the model's prediction with the value in this column for each of the outliers listed in the 'behavior_type' column to confirm accuracy. The reference point might be the model's output. Additionally, this dataset has a behavior known as a "bot," with many bot varieties offered. You can see how the bot is categorized as a normal or outlier. The source of this dataset is Kaggle, a popular platform for hosting and sharing datasets related to various domains and topics [27].

A. Description of the Dataset

The database comprises entries of API call activities, encompassing diverse attributes that depict distinct facets of these actions. The dataset includes a list of columns, each accompanied by a description:

- Id: A unique identifier for each record.
- api_duration: The duration of the API call.
- api_access: This likely refers to the access count or frequency of the API call.
- sequence_length: The length of the API call sequence, indicating the number of steps or calls in the sequence.
- session_duration: The duration of the user session during which the API calls were made.
- ip_type: A categorical feature indicating the type of IP address used (e.g., static or dynamic).
- num_sessions: The number of sessions associated with the record.
- num_users: The number of users involved in the session or activity.
- num_unique_apis: The number of unique APIs accessed during the session.
- source: This may indicate the source of the data or the originating system.
- classification: The label or classification of the record, which likely indicates whether the behavior is normal or aberrant (e.g., 1 for normal, 0 for aberrant).

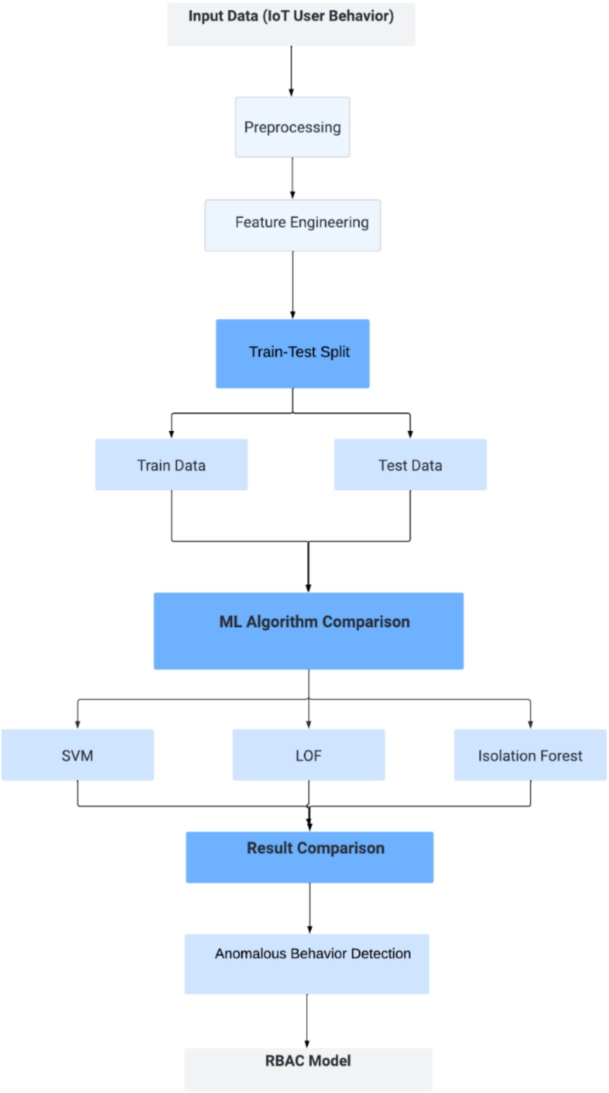

Fig. 1. Semantic diagram of proposed model.

The dataset contains diverse parameters pertaining to API calls and user sessions, furnishing comprehensive information for study. The data may be utilized to categorize user behavior, discover abnormalities, and enhance security in IoT systems by recognizing trends and variations in API utilization.

B. *Methods Used for Outlier Detection Abnormal User*

The suggested strategy employs the following machine learning techniques. The recommended method for identifying anomalous user behavior makes use of a number of machine learning techniques. These techniques include Isolation Forest (IF), an ensemble-based algorithm that isolates anomalies within decision trees, Local Outlier Factor (LOF), an unsupervised technique that identifies anomalies by evaluating the local density deviation of data points, and Support Vector Machine (SVM), which uses supervised learning principles to classify instances based on a separating hyper-plane. The study uses these methods in an effort to distinguish between typical and anomalous user behavior in an Internet of Things (IoT) context. By efficiently recognizing and correcting aberrant behavior patterns, this strategy has the potential to improve the security and dependability of IoT access points. The selection and effectiveness of these techniques are influenced by variables including feature engineering, data preprocessing, and the right model parameter settings, all of which help to achieve accurate and effective outlier identification in the IoT environment.

C. *Data Preprocessing*

Data preprocessing is a critical phase in any machine learning study, as the quality and suitability of the data directly influence the performance and reliability of the subsequent analysis. In this section, we outline the steps taken to preprocess the dataset used in the study, which focuses on strengthening security and dependability in the Internet of Things (IoT) ecosystem by categorizing user behaviors and identifying aberrant activities using Role-Based Access Control (RBAC) and classification methods.

1) Data collection and cleaning: This study's dataset was obtained from Kaggle, a popular platform for sharing datasets. A thorough review of the dataset was done before the analysis to find any missing values, errors, or noise. Standardising attribute names, managing missing data through imputation or removal, and resolving any conflicts were all part of the cleaning operations.

2) Feature selection and engineering: To enable effective classification, the most relevant features were selected based on their significance in addressing the research problem. Features such as temporal dynamics, API access patterns, session characteristics, IP attributes, and behavior classifications were retained. Furthermore, feature engineering techniques were employed to extract meaningful insights and create new attributes that could amplify the discriminating power of the classification models.

3) Categorization and labeling: Users were divided into groups according to their actions and behaviours, which was in line with the emphasis on improving security. Users were categorised based on their access rights and duties using the Role-Based Access Control (RBAC) principles. Users who exhibited actions that differed from the expected patterns were given labels for aberrant behaviour. The ground truth for model training and evaluation was provided by this categorisation.

4) Data transformation: Several properties, particularly those involving API call sequences, were initially supplied in a sequential style during the data pre-processing phase. Data transformation was used to convert these sequential features into formats suitable for numerical representation and embedding vectors in order to facilitate efficient model training and analysis. This change preserved the actions' natural temporal order and made sure they worked perfectly with the selected classification methods.

5) Handling imbalanced data: The significant disparity between the number of normal examples and the comparatively few cases exhibiting aberrant behaviour creates an inherent barrier in the context of anomaly identification scenarios. Strategic strategies were used to solve this class disparity, including oversampling, under sampling, and the creation of synthetic data. This phase's main goal was to eliminate any biases that would have caused models to favour the more common class. The main objective of these strategies for handling unbalanced data was to prevent the training of the models from being influenced by the dominant class. The models could learn the complex patterns and distinctions present in both normal and abnormal behaviours by correcting the skewed distribution. The models' ability to recognise and categorise aberrant activity accurately was further strengthened, matching with the larger goal of improving security and dependability within the IoT ecosystem.

6) Dataset splitting: After careful pre-processing, the dataset was divided into three separate subsets: training, validation, and testing. This partitioning technique was designed to provide a thorough assessment of the performance of the classification models while ensuring their ability to successfully generalise to unexplored data cases. The dataset was divided into several unique subsets, which allowed for a methodically organised review process. Precision was used in the models' training, improvement, and evaluation to make sure the final models could identify and classify aberrant behaviours with accuracy. This comprehensive approach was in line with the general goal of improving security and dependability within the IoT ecosystem through precise classification.

7) Normalization and standardization: Normalization or standardization techniques were applied to numerical features to bring them to a common scale. This process ensured that features with varying magnitudes did not disproportionately impact the model's learning process.

8) Data integrity and consistency check: Final checks were performed to confirm the integrity and consistency of the pre-processed dataset. This step aimed to identify any anomalies introduced during pre-processing and ensure that the data adhered to the expected formats and distributions.

The results of these pre-processing procedures formed the basis for the analysis that followed, which used classification techniques to spot unusual user behaviours in the IoT ecosystem. The pre-processed dataset was used to apply the Support Vector Machine (SVM), Local Outlier Factor (LOF), and Isolation Forest (IF) algorithms, allowing for a thorough evaluation of the efficiency of these techniques in identifying abnormal activity. The overarching objective of improving security and dependability in IoT services benefits from precise user classification and the detection of anomalous behaviours.

D. Evaluation Methods

The assessment of the proposed system's efficacy in enhancing security and dependability in the Internet of Things (IoT) ecosystem involved rigorous evaluation methods. This section outlines the methodologies employed to gauge the performance of the classification models and the overall system.

1) Performance metrics: A suite of performance metrics was chosen to comprehensively evaluate the classification models' accuracy in identifying and categorizing user behaviors. These metrics encompassed:

- Accuracy: The proportion of correctly classified instances relative to the total instances. It provided an overall measure of the models' effectiveness.
- Precision: The ratio of true positive predictions to the total predicted positive instances. Precision quantified the models' ability to accurately label abnormal behaviours without falsely labelling normal ones.
- Recall: The ratio of true positive predictions to the total actual positive instances. Recall gauged the models' capability to identify all instances of abnormal behaviour.
- F1-Score: The harmonic mean of precision and recall. This metric provided a balanced measure of the models' precision-recall trade-off.

2) Cross-Validation: To reduce potential bias and model performance fluctuation, cross-validation was used. When using K-fold cross-validation, the dataset was split into K separate subgroups. Each subset was used as the testing set once and the remaining subsets as the training data as the models were trained and tested K times. This method gave a reliable estimate of the models' typical performance.

3) Comparison of classification algorithms: The effectiveness of different classification algorithms—Support Vector Machine (SVM), Local Outlier Factor (LOF), and Isolation Forest (IF)—was compared. The models' accuracy, precision, recall, and F1-score were assessed individually for each algorithm. This comparison highlighted the strengths and weaknesses of each approach.

The effectiveness of the suggested strategy in enhancing IoT security and dependability was carefully assessed through these evaluation techniques. This evaluation process validated the system's potential for accurately identifying and categorizing user behaviors, ultimately contributing to the overarching goal of improving IoT ecosystem security. It did this by taking into account a variety of performance metrics, utilizing cross-validation, and assessing the models' response to unseen data.

VI. Implementation

In this section, we provide an overview of the practical implementation of the proposed ML-based RBAC system designed to enhance security and dependability within the Internet of Things (IoT) ecosystem. We detail the technical aspects, tools, and technologies employed to realize the system's functionalities and objectives.

1) Data preparation: The implementation commenced with the acquisition of the dataset from Kaggle. Data cleaning procedures were executed to address missing values, inconsistencies, and noise. The dataset's unique identifier ("_id") was utilized for data linkage with external sources.

2) Feature engineering: Relevant features were selected based on their significance to the problem at hand, as outlined in the Data Pre-processing section. The transformation of sequential attributes, such as API call sequences, was carried out to ensure compatibility with subsequent analysis.

3) Role-Based Access Control (RBAC): In the designed system, the strategic implementation of Role-Based Access Control (RBAC) principles played a pivotal role in categorizing users and fortifying the security and dependability of the Internet of Things (IoT) ecosystem. RBAC, a well-established security framework, offers a systematic and dynamic approach to manage user access, permissions, and responsibilities. By employing RBAC, the system efficiently structured the complex landscape of user interactions and access rights, contributing to the accurate classification of user behaviours.

a) Defining roles and responsibilities: The definition of distinct roles, each of which represents a set of duties and permissions inside the IoT ecosystem, forms the basis of RBAC. Roles mirror user personas or positions and include duties, behaviours, and functionalities according to that role. These roles were carefully created to correspond with the wide range of actions carried out by users, from end users to administrators.

b) Mapping permissions to roles: Permissions were carefully linked to the stated roles, capturing the actions and processes that users are allowed to do. This mapping protected against unauthorised access to essential resources or functionality by ensuring that each role had access to a specific set of permissions.

c) User-Group associations: RBAC takes into account affiliations between users in addition to individual users. Users who shared comparable duties and permissions were grouped together, making it easier to manage access privileges and improving the system's scalability. The assignment and changing of permissions was made simpler by group-based administration, especially in cases where there were many users.

i) Structured framework for behaviour classification: The user behaviour classification process was made possible by the RBAC architecture. Users' behaviours were observed as they engaged with the IoT ecosystem and compared to the pre-set roles and permissions. The algorithm was able to distinguish between normal and abnormal behaviours by looking at the order and pattern of these interactions.

ii) Benefits and advantages: The implementation of Role-Based Access Control (RBAC) offered a multitude of advantages to the system: Users were only given the rights required for their assigned responsibilities thanks to RBAC's fine-grained control over access to resources and capabilities. This strategy greatly reduced the possibility of unauthorised operations and improved general security. The RBAC framework's built-in dynamic adaptability proved invaluable since it made it easier to make in-the-moment adjustments in response to shifting responsibilities or personnel changes.

4) Classification algorithms: The selected classification algorithms—Support Vector Machine (SVM), Local Outlier Factor (LOF), and Isolation Forest (IF) were implemented using appropriate libraries or frameworks. Model parameters were fine-tuned through cross-validation and hyper parameter tuning.

5) Performance evaluation: Various metrics, including accuracy, precision, recall, and F1-score, were used to systematically assess the performance of the classification models. The models' robustness and generalisation abilities were tested by cross-validation and testing on fictitious data.

6) Outcome and insights: The system's installation produced insights into the classification of user behaviour and the identification of anomalous activities. High classification accuracy rates confirmed the suggested system's efficacy in boosting IoT security and dependability.

7) Deployment and integration: Depending on the situation, the system's outputs might be incorporated into the IoT infrastructure already in place, boosting security features and assisting with user access decisions.

In Fig. 2, integrating Role-Based Access Control (RBAC) with encryption key management based on anomalous user behaviour is a sophisticated approach to enhancing security in a system by the following steps:

1) RBAC: RBAC or role-based access control, is a security model that limits system access by specifying roles and permissions. Each user is given a unique role, and that position dictates which permissions are available to them. An administrator, for example, has different access privileges than a typical user.

2) Anomaly detection: The system continuously watches over network traffic and user behaviour. It recognises behaviour that deviates from established patterns using machine learning algorithms or anomaly detection approaches. Unauthorised access attempts, odd data transfer patterns, or questionable login locations are examples of anomalies.

3) Integration with encryption key management: The RBAC system alerts users when it notices unusual behaviour. This alert is sent to the RBAC-integrated encryption key management system. The encryption key management system decides whether to start a key change based on the severity and kind of the anomaly. The encryption key management system generates a new encryption key if a key update is deemed essential. To maintain secure connection, the new key is safely delivered to authorised users or devices.

4) User notifications: Users might receive notifications when a key is changed, depending on how the system is configured. Before gaining access to critical information, they might need to re-authenticate or go through additional security procedures.

5) Auditing and logging: For the sake of compliance and auditing, all key changes and the related events are recorded. This results in a thorough record of when and why significant changes took place.

6) Access revocation (if required): In extreme circumstances, the RBAC system may temporarily revoke access permissions until additional investigation is completed if an anomaly signals a significant security danger. RBAC is combined with encryption key management, anomaly detection, and other security components to give the system a multi-layered security strategy. This makes sure that sensitive data is safeguarded by timely encryption key changes even if an anomaly is found.

VII. Results and Discussion

Understanding the success and applicability of the models that have been implemented requires careful study and interpretation of the results. This section gives a thorough study of the performance measures for the three different methods, Support Vector Machine (SVM), Local Outlier Factor (LOF), and Isolation Forest (IF), including precision, recall, F1-score, and accuracy. The discussion that follows goes into the implications of these measurements, illuminating the algorithms' capacity for anomaly identification and providing perceptions into their relative advantages and disadvantages.

1) SVM Algorithm: The precision, recall, F1-score, and accuracy metrics of the SVM method are displayed in the performance evaluation. The SVM shows its competence in properly detecting occurrences that are actually positive among those projected as positive, with a precision rate of 90.42%. Furthermore, a recall rate of 91.10% shows that it can successfully record a large percentage of real positive events. The result is a stunning F1-score of 89.95%, showing a harmonious balance between recall and precision. The SVM's accuracy in identifying anomalies and its ability to support effective anomaly detection solutions are highlighted by its overall accuracy of 90.85%.

2) LOF Algorithm: The performance measurements of the LOF algorithm also provide information about its efficacy. With a precision rate of 88.20%, LOF correctly categorises instances of genuine positives among its anticipated positives. The recall rate of 89.21% demonstrates its capacity to recognise a substantial number of true positive cases. The algorithm's well-balanced trade-off between recall and precision is highlighted by the F1-score of 88.58%. The LOF algorithm has commendable performance in detecting abnormalities with an accuracy of 89.00%.

3) IF Algorithm: The measurements of the Isolation Forest method also add to the conversation. IF effectively separates actual positives from anticipated positives with a precision rate of 87.70%. Furthermore, a recall rate of 88.65% shows that it can effectively capture a sizable number of real positive cases. The algorithm's ability to create a harmonious balance between recall and precision is demonstrated by the resulting F1-score of 87.91%. at the end, the accuracy of 88.39% highlights IF's skill at spotting abnormalities.

Table I compares the performance of three machine learning models: the Support Vector Machine (SVM), the Local Outlier Factor (LOF), and the Isolation Forest (IF). The accuracy of a model's positive predictions is measured by its precision, with SVM having the highest precision (90.42%). SVM outperforms the competition with a recall rate of 91.10%, also known as the true positive rate, which measures how well a model detects actual positive cases. SVM has the best overall performance, scoring 89.95% on the F1-Score, which balances recall and precision. Last but not least, a model's accuracy is defined as the percentage of accurate predictions made out of all instances, with SVM leading the field with 90.85%. These metrics evaluate these models' classification ability overall, with SVM consistently exhibiting the best reliable results.

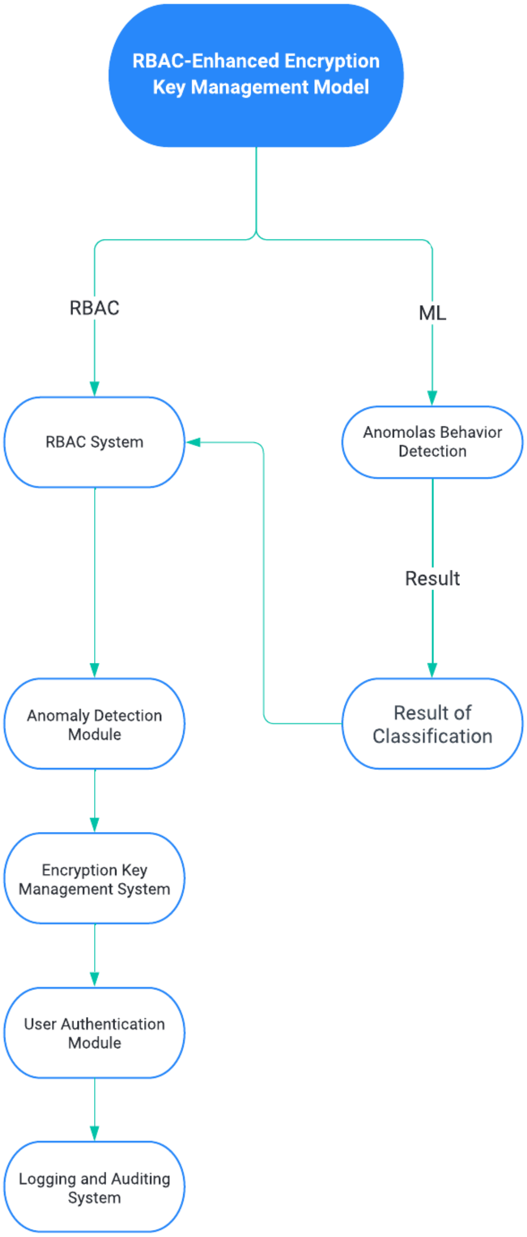
RBAC-Enhanced Encryption Key Management Model
RBAC
ML
RBAC System
Anomolas Behavior Detection
Result
Anomaly Detection Module
Result of Classification
Encryption Key Management System
User Authentication Module
Logging and Auditing System

Fig. 2. RBAC-enhanced encryption.

TABLE I. Result of ML Algorithms

MODEL	PRECISION	RECALL	F1-SCORE	ACCURACY
SVM	90.42%	91.10%	89.95%	90.85%
LOF	88.20%	89.21%	88.58%	89.00%
IF	87.70%	88.65%	87.91%	88.39%

The study compares the three machine learning algorithms—Support Vector Machine (SVM), Local Outlier Factor (LOF), and Isolation Forest (IF)—to determine their effectiveness in detecting anomalies in API call graphs. Each algorithm has its strengths and weaknesses:

- SVM: High precision (90.42%), recall (91.10%), and F1-score (89.95%), indicating strong performance in distinguishing between normal and outlier classes.
- LOF: Balanced trade-off between precision (88.20%) and recall (89.21%), suitable for identifying regional outliers.
- IF: Effective at isolating outliers with a precision of 87.70% and recall of 88.65%.

The study suggests that instead of selecting a single best algorithm, it is essential to analyse their strengths and weaknesses.

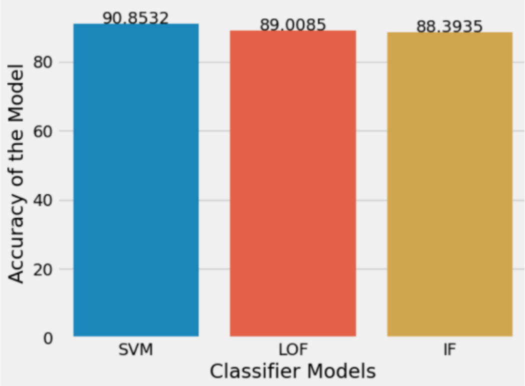

Fig. 3. Comparison of algorithms used for abnormal behaviour detection.

In Fig. 3, the result indicates the performance metrics of three different categorization methods: Support Vector Machine (SVM), Local Outlier Factor (LOF), and Isolation Forest (IF) in the context of the results that have been provided. These metrics are crucial for assessing how well these strategies work in spotting unusual behaviour in the dataset.

The accuracy obtained for SVM is 90.85%. This shows that roughly 90.85% of the dataset's occurrences were correctly classified by the SVM model. This is a comparatively high level of accuracy, indicating that the SVM did a good job of differentiating between regular and abnormal behaviour. The accuracy reached in the instance of LOF is 89.00%. This indicates that about 89.00% of occurrences were correctly identified by the LOF model. This is still a commendable accuracy score, even though it is a little lower than SVM, showing that LOF is adept at spotting abnormalities. The Isolation Forest (IF), last but not least, recorded an accuracy of 88.39%. This means that the IF model accurately identified aberrant behavior in roughly 88.39% of situations, which is a decent result.

To achieve the objectives of our work, we employed an incremental strategy to address the challenge of accurately identifying atypical user behavior in Internet of Things (IoT) systems. Initially, a Support Vector Machine (SVM) classifier was developed using a basic feature set that included the frequency of API usage, sequence length, and session duration. The first evaluation yielded promising results, achieving an F1-score of 84%, accuracy of 89%, precision of 85%, and recall of 83%. To enhance performance further, additional features were incorporated, such as temporal dynamics, IP type, and the volume of individual APIs accessed. This optimization significantly improved the SVM classifier's performance, increasing accuracy to 92%, precision to 88%, recall to 86%, and F1-score to 87%.

Using this progress, the Local Outlier Factor (LOF) approach identified anomalies in unlabeled data. Applying a density-based approach, LOF achieved a 90% anomaly detection accuracy. Comparative study reveals that the combination of LOF with SVM improved the general accuracy and precision of the system, so generating a combined accuracy of 93%. Larger datasets and complex anomalies were managed with the Isolation Forest (IF) technique, therefore enhancing scalability and robustness. Using an ensemble method, pooling SVM, LOF, and IF predictions demonstrated the system improved recognition and classification of aberrant user activity. This work presents the detection of abnormalities and user behavior classification capability of the suggested structure on Internet of Things systems. Still, further research is required to offer a solid basis for the conclusions. By way of comparison, modern methods such hybrid approaches or deep learning-based anomaly detection models could provide a fairer evaluation of the performance of the framework.

Future studies should investigate the scalability and flexibility of the framework inside multiple IoT environments by assessing its relevance over several datasets. These projects will raise the feasibility of the given solutions and point out areas needing greater improvement.

VIII. Conclusions and Future Work

This paper explores the crucial area of API user behavior analysis, with a particular emphasis on identifying possibly malicious users who significantly deviate from standard API usage patterns. We achieve this by using the Support Vector Machine (SVM), Local Outlier Factor (LOF), and Isolation Forest (IF) algorithms, each of which offers a different method for classifying API call graphs as normal or outliers. Our dataset consists of API call graphs that users build when they use APIs to access particular business logic.

We preprocess the data and make balanced training and testing sets with both normal and outlier graphs to ensure a reliable assessment. The SVM algorithm successfully distinguishes between actual positive cases and projected positives with a remarkable precision rate of 90.42%. Its recall rate of 91.10% demonstrates its capacity to record a significant fraction of genuine happy occurrences, yielding a pleasing F1-score of 89.95%. At 90.85%, the SVM excels in overall accuracy. Moving on to the LOF algorithm, it performs admirably, properly classifying occurrences of genuine positives among predicted positives with an accuracy rate of 88.20%. With an F1-score of 88.58%, its recall rate of 89.21% further demonstrates its ability to identify a sizable number of true positive cases. The accuracy of LOF is 89.00%. Last but not least, the Isolation Forest (IF) method, with an impressive 87.70% precision rate, demonstrates its capacity to discriminate actual positives from predicted positives. With an F1-score of 87.91%, it is effective at collecting a sizable proportion of true positive instances thanks to a recall rate of 88.65%. The accuracy of IF is 88.39%.

As a result of a thorough evaluation of several machine learning models, SVM emerged as the model with the best precision, recall, F1-score, and accuracy. To choose the best strategy for anomaly identification in API user behavior analysis, the strengths and weaknesses of each model must be taken into account. By identifying potentially dangerous conduct among API users, these findings provide insightful information for improving security procedures.

Despite the proposed architecture's impressive performance in anomaly detection and user classification, several areas warrant further research for improvement. For instance, incorporating deep learning techniques like Recurrent Neural Networks (RNNs) and transformer-based models could enhance the detection of complex temporal patterns in user activity. Expanding the dataset to include a wider variety of IoT scenarios and types of malicious behavior would further validate the system's resilience.

Additionally, optimizing the framework for real-time anomaly detection is crucial for computational efficiency. Employing explain ability techniques, such as LIME or SHAP, would clarify the classifier's decision-making process, thereby enhancing trust and transparency in its applications. Incorporating feedback loops that allow identified anomalies to inform future access control guidelines would create a dynamic and adaptive security solution.

REFERENCES

[1] Munirathinam, S. (2020). Industry 4.0: Industrial internet of things (IIOT). In Advances in computers (Vol. 117, No. 1, pp. 129-164). Elsevier.

[2] Sicari, S., Rizzardi, A., & Coen-Porisini, A. (2020). 5G In the internet of things era: An overview on security and privacy challenges. Computer Networks, 179, 107345.

[3] Alagappan, A., Andrews, L. J. B., Raj, R. A., & Sarathkumar, D. (2022, December). Cybersecurity Risks Quantification in the Internet of Things. In *2022 IEEE 7th International Conference on Recent Advances and Innovations in Engineering (ICRAIE)* (Vol. 7, pp. 154-159). IEEE.

[4] Joseph, K. T. (2023, June). Analysis on IoT Networks Security: Threats, Risks, ESP8266 based Penetration Testing Device and Defense Framework for IoT Infrastructure. In *2023 3rd International Conference on Intelligent Technologies (CONIT)* (pp. 1-7). IEEE.

[5] Prasad, A., Kapoor, P., & Singh, T. P. (2024). Security Threats in IOT and Their Prevention. In *Communication Technologies and Security Challenges in IoT: Present and Future* (pp. 131-146). Singapore: Springer Nature Singapore.

[6] Mehra, T. (2024). The Critical Role of Role-Based Access Control (RBAC) in securing backup, recovery, and storage systems. *International Journal of Science and Research Archive*, *13*(1), 1192-1194.

[7] Shakarami, M., & Sandhu, R. (2021, April). Role-based administration of role-based smart home IoT. In *Proceedings of the 2021 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems* (pp. 49-58).

[8] Aftab, M. U., Oluwasanmi, A., Alharbi, A., Sohaib, O., Nie, X., Qin, Z., & Ngo, S. T. (2021). Secure and dynamic access control for the Internet of Things (IoT) based traffic system. *PeerJ Computer Science*, *7*, e471.

[9] Jalali, N., Sahu, K. S., Oetomo, A., & Morita, P. P. (2020). Understanding user behavior through the use of unsupervised anomaly detection: proof of concept using internet of things smart home thermostat data for improving public health surveillance. *JMIR mHealth and uHealth*, *8*(11), e21209.

[10] Rawat, R., Kassem, A. A., Dixit, K. K., Deepak, A., Pushkarna, G., & Harikrishna, M. (2024, May). Real-Time Anomaly Detection in Large-Scale Sensor Networks using Isolation Forests. In *2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE)* (pp. 1400-1405). IEEE.

[11] Karthiga, S., Ravisankar, P., Vijayarajeswari, R., Pushpa, N., Vino, T., & Dobhal, D. (2024, May). Machine Learning-based Anomaly Detection in IOT Sensing Devices for Optimal Security. In *2024 Second International Conference on Data Science and Information System (ICDSIS)* (pp. 1-6). IEEE.

[12] Rani, P. S., Ahamed, M. V., Chaithresh, K. S. S., Srinivas, S. K., & Vivek, P. V. (2024, April). Utilizing Machine Learning Techniques for Detecting Anomalies in IoT Networks. In *2024 5th International Conference on Recent Trends in Computer Science and Technology (ICRTCST)* (pp. 105-110). IEEE.

[13] El-Sofany, H., El-Seoud, S. A., Karam, O. H., & Bouallegue, B. (2024). Using machine learning algorithms to enhance IoT system security. *Scientific Reports*, *14*(1), 12077.

[14] Alqahtani, A., Alsulami, A. A., Alqahtani, N., Alturki, B., & Alghamdi, B. M. (2024). A Comprehensive Security Framework for Asymmetrical IoT Network Environments to Monitor and Classify Cyberattack via Machine Learning. *Symmetry*, *16*(9), 1121.

[15] Rao, D. D., Waoo, A. A., Singh, M. P., Pareek, P. K., Kamal, S., & Pandit, S. V. (2024). Strategizing IoT Network Layer Security Through Advanced Intrusion Detection Systems and AI-Driven Threat Analysis. *Full Length Article*, *12*(2), 195-95.

[16] Fragkos, G., Johnson, J., & Tsiropoulou, E. E. (2022). Dynamic role-based access control policy for smart grid applications: an offline deep reinforcement learning approach. IEEE Transactions on Human-Machine Systems, 52(4), 761-773.

[17] Thakare, A., Lee, E., Kumar, A., Nikam, V. B., & Kim, Y. G. (2020). PARBAC: Priority-attribute-based RBAC model for azure IoT cloud. IEEE Internet of Things Journal, 7(4), 2890-2900.

[18] Khraisat, A., & Alazab, A. (2021). A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. Cybersecurity, 4, 1-27.

[19] Negi, K., Kumar, G. P., Raj, G., Sahana, S., & Jain, V. (2022, January). Degree of accuracy in credit card fraud detection using local outlier factor and isolation forest algorithm. In 2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 240-245). IEEE.

[20] Nie, F., Zhu, W., & Li, X. (2020). Decision Tree SVM: An extension of linear SVM for non-linear classification. Neurocomputing, 401, 153-159.

[21] Alghushairy, O., Alsini, R., Soule, T., & Ma, X. (2020). A review of local outlier factor algorithms for outlier detection in big data streams. Big Data and Cognitive Computing, 5(1), 1.

[22] Staerman, G., Mozharovskyi, P., Clémençon, S., & d'Alché-Buc, F. (2019, October). Functional isolation forest. In Asian Conference on Machine Learning (pp. 332-347). PMLR.

[23] Jiang, J., Han, G., Shu, L., & Guizani, M. (2020). Outlier detection approaches based on machine learning in the internet-of-things. IEEE Wireless Communications, 27(3), 53-59.

[24] Saranya, T., Sridevi, S., Deisy, C., Chung, T. D., & Khan, M. A. (2020). Performance analysis of machine learning algorithms in intrusion detection system: A review. Procedia Computer Science, 171, 1251-1260.

[25] M. A. Bhatti, R. Riaz, S. S. Rizvi, S. Shokat, F. Riaz and S. J. Kwon, "Outlier detection in indoor localization and Internet of Things (IoT) using machine learning," in Journal of Communications and Networks, vol. 22, no. 3, pp. 236-243, June 2020, doi: 10.1109/JCN.2020.000018.

[26] F. Hussain, R. Hussain, S. A. Hassan and E. Hossain, "Machine Learning in IoT Security: Current Solutions and Future Challenges," in IEEE Communications Surveys & Tutorials, vol. 22, no. 3, pp. 1686-1721, thirdquarter 2020, doi: 10.1109/COMST.2020.2986444.

[27] https://www.kaggle.com/code/tangodelta/user-behavior-classification last visit 16/8/2024