International Journal of Scientific Research & Engineering Trends
Volume 10, Issue 1, Jan-Feb-2024, ISSN (Online): 2395-566X

# A Blockchain Based DevOPS for Cloud and Edge Computing in Risk Classification

**Hemanth Swamy**
Senior Software Engineer
Motorola Solutions

**Abstract- Overlying environments with large volumes of data are challenging to handle on a single server. Consequently, knowing how to secure unpredictable data in a changing setting is crucial. The authors express worry about the potential security risks associated with susceptible data in a distributive system based on the mobile edge. Therefore, it would seem that edge computing is a great vantage point from which to conduct training in an ecosystem based on the edge. Data security, exposure of data, and the likelihood of a data breach may all be enhanced by combining machine learning methods with blockchain's consensus methodology and edge computing. In this study, we demonstrate how to integrate realistic ML approaches into a DevOps environment. Our system's danger assessment is a machine learning model that estimates the risk level of each authentication attempt based on digital identity variables like IP address, browser user agent, and user behavior. Using a subset of login data variables, we validated our system and built risk classifier models to determine the amount of danger posed by users. Therefore, a way to train the shared data is via the idea of machine learning. Under the watchful eye of two-factor authentication, data security was previewed in a dataset that included several exposed, vulnerable, recovered, and protected pieces of information. Data and security vulnerabilities in smart computing edge devices, as well as their fixes, are covered in this study. Machine learning methods, including various classifiers and optimization algorithms, plus the blockchain consensus approach, provide data confidentiality in the suggested model. In addition, the authors used an edge computing setting to implement the suggested techniques by sending data in several batches to various customers. Consequently, the use of blockchain servers ensured that client anonymity was preserved. In addition, the writers used the federated learning method to train separate batches of client data. This study presents the outcomes of a training model that utilizes blockchain technology in an edge-based technology setting.**

**Index Terms- Blockchain Technology, Edge computing DevOPs, Machine learning techniques, and Voting classifier**

## I. INTRODUCTION

Improvements, fixes, and patches are a regular part of modern software projects, and customers need them virtually everyday. Regular releases occur on a weekly or daily basis, with significant feature upgrades occurring every three months. Today, more than ever before, software delivery is to be agile. Many businesses are now trying to take agile-based delivery models that they were testing out in the past and make them enterprise-grade [1]. Container technology is extensively used and often integrated with microservices due to the rise of cloud native with DevOps. Ensuring the quality of service for customers is crucial when deploying container-based microservices in dispersed cloud-edge architecture. Unfortunately, when it comes to tailored deployment solutions, current container orchestration systems fall short [2]. They lack the flexibility to choose the optimal deployment site based on the user's budget.

Among the most popular models for large-scale software development, packaging, and deployment today is the microservices paradigm. Unfortunately, due to their cloud-centric origins, microservices DevOps approaches may not be the best fit for resource-constrained contexts such as the Edge. The storage space as well as network bandwidth needed to manage base images, CPU, as well as memory, among other resources, may add up quickly when each software component is deployed as a separate microservice [3]. Among the many Cloud options available today, serverless computing is quickly becoming a favorite. Cloud users, programmers, and devops teams are reaping many benefits from Function as a Service, notably reduced service costs, faster development times, and easier deployment. The Cloud-Edge Continuum may very well rely on this technology. No matter how great these functionalities are, a Cloud broker is still required to construct native FaaS apps [4]. Artificial intelligence, big data, edge-fog-cloud, development as well as operations (DevOps), and

other disruptive technologies have the potential to radically alter the industry. However, optimizing productivity via their optimal use remains a significant difficulty [5]. Cloud computing has changed the way conventional programs are built and run in the last several decades. Clouds and information centers were constructed at an increasing rate. But new collaborative apps like AI, IoT, as well as autopilot can't be supported by the present Clouds' centralized administration method because of their lack of dispersion. However, edge computing is still in its early stages of development, both conceptually and experimentally [6]. The cost-effectiveness, scalability, and simplicity of deployment of cloud-hosted services are making them more and more popular for hosting corporate applications. The field of DevOps is undergoing rapid evolution to enable the quick creation, update, as well as release process of applications hosted in the cloud. Knowledge of continuous integration with continuous delivery automation should be imparted to the next generation of software development experts via proper training [7]. Distributed systems that combine cloud, fog, with edge computing are what we now call the IoT. Ensuring their availability while speeding up their maintenance and continual development is no easy feat. Among other things, DevOps encourages regular and rapid feedback between operations and development, which helps find issues before they affect consumers. The problem is that no one has formally defined how to accomplish it [8]. Smart gadgets and equipment with varying degrees of processing capability that gather and transmit sensory data make up the Internet of Things. Distributed, portable, reusable, as well as automatically maintained in heterogeneous contexts are essential features of an IoT application because of the diverse nature of IoT ecosystems. These functionalities cannot be realized without the use of software provisioning and orchestration; doing so will improve system performance and decrease reaction time [9]. Both academics and businesses are actively discussing cloud service orchestration. There is a pressing need for agile and effective processes that simplify devops procedures due to the growing variety of Cloud offerings and the high demand for scalable and adaptable apps from consumers. The previously difficult issue of effortlessly and flexibly providing services is further complicated by the new possibility of executing software programs or parts of applications on resources situated at the network's edge [10].

**Discussions**

In order to enhance the security features of smart edge computing devices, the authors used a consensus method based on Blockchain with Machine Learning approaches. The authors acquired the dataset that included data that was vulnerable, data that was exposed, data that had been retrieved, and so on. In addition, a number of optimization methods like as SGD, Momentum SGD, RMSprop, as well as Adam were used by the writers, who employed a range of classification approaches such as Naıve Bayes's K-Neighbors, Random Forest, as well as SVM. In addition, the scientists used ensemble classification methods to integrate these classifiers on many data blocks, and the results were greatest when the ensemble classifier was used. The Ensemble classifier improved accuracy to 92% from 87% for the best classifier in the framework, the K-Neighbours Classifier. Following their training, the authors evaluated their vulnerability after implementing two-factor authentication. In addition, the authors used a multilayered perceptron framework to implement federated instruction in an intelligent Edges computing-based environment, and they delivered the data in phases to several clients. In conclusion, the suggested model was trained using a 1.51 global loss and a 96% global accuracy.

The following are some of the paper's contributions:

- In an edge technology environment, data security is constructed with vulnerable data and recoverable data utilizing a two-level-architecture.
- Using information stored in blockchain-based storage, appropriate security measures may be put in place for both vulnerable and compromised data.
- The data is trained using machine learning techniques using ensemble methods to achieve locally asymptotic stability. The model is then trained in a federated edge-based architecture to achieve global asymptotic stability.
- Separate data using the consensus method of blockchains to reduce the possibility of poisoning attacks.

Here is the breakdown of the remaining sections of the paper: A synopsis of the relevant literature is given in Section 2. In Section 3, we provide the groundwork for the planned work using machine learning with blockchain-based security. Section 4 presents the outcomes and implementations of the experiments. In Section 5, we draw some conclusions and look forward to our future work in this study.

## II. RELATED WORK

Using edge computing paradigms to support mobile IoT and DevOps integration, the project aims to provide a software architecture for applications that use the IoT, the edge, with the cloud computing continuum [11]. To facilitate reliable and adaptable deployment, the system is modularized. The various levels of the system make use of loosely coupled services and module descriptions. The paper outlines the software architecture for a DevOps-enabled Edge computing solution that supports mobile and flexible IoT solutions and is part of the IoT-Edge-Cloud computational continuity. An intelligent transport system for rolling stock domains is used as an example to validate the proposed IoT-Edge-Cloud continuum architecture. In [12], the authors review Scientific Workflow as it relates to the FaaS paradigm, with the goal of developing applications that fully use Native Serverless Workflows. They

first outline the construction of a function-interaction-describing custom Workflow Manifest DSL, and then they detail the agent's ability to deploy and coordinate architecture-independent functions in accordance with the Manifest. In the end, authors enable functions to use the Continuum tier's properties during deployment by federating the Cloud, Fog, and Edge tiers in a single environment. Giving the tools and platforms that DevOps teams need to deploy, monitor, and manage such applications is a critical challenge in the context of [13]. In such a case, the SODALITE@RT open-source platform might be useful. Here they outline the key aspects of the SODALITE@RT: by modeling cloud-edge resources as well as apps using open standards as well as infrastructure code, and then automating their deployment, monitoring, as well as administration in chosen infrastructures. The SODALITE@RT's potential is shown via an appropriate case study. By providing automated along with ongoing surveillance input from operations, the article discusses the "fast as well as continuous tracking feedback concerning network availability" (F&CF availability) operation [14], which aids in the establishment of the functioning of the IoT system. The SPEM, which stands for software as well as system process engineering, has formalized such arrangement. Following the formalization of the F&CF availability operations, teams are better equipped to detect and fix outage concerns, as shown by a real-world example. The finished result is a distributed and modifiable monitoring component called monitoring as code. Included in the IoT infrastructure is that component. This enables DevOps teams to take use of MaC's monitoring on demand capabilities, which allow for the setup of indicators and measurements during runtime. Automating, versioning, and replicating monitoring components is possible using a MaC method to explicitly implementing that process. In order to identify and install IoT applications with adjusted cost budgets, a technique based on evolutionary algorithms is proposed in [15]. As an optimization problem that aims to minimize cost while minimizing user service latency, the application distribution issue is defined as an NP-hard problem. Using a genetic algorithm allows for a more effective deployment and successful completion of that problem. The suggested approach beats four baseline algorithms—time-greedy, cost-greedy, random, and PSO—on both real and synthetic datasets. The proposed method provides tailored alternatives for deploying edge-cloud infrastructure. Based on microservices and supported by technologies such Docker for containerization, Ansible for tracking, and Kubernetes for scalability, there is a fundamental Meta-model of the IoT ecosystems provided in [16]. Using that meta-model, they may construct a cloud, fog, or edge-based object-connection system. Smart cities, houses, cars, health, farming, and industries are just a few of the many possible uses for that meta-model. Consideration of the specific features of each infrastructure is essential when assisting DevOps teams with (i) developing the right operational building for the application, (ii) transforming it into infrastructure software that automates deployment, and (iii) executing the application, as highlighted in [17]. The SODALITE architecture is designed to deal with such kind of thing. Using a case study to demonstrate their practical application, the paper highlights the core features of the inaugural version of the structure and focuses on managing cloud and HPC clusters. In [18], we have an example of a case study that shows how customizable SaaS updates were deployed at the edge of the Internet of Things (IoT). This might pave the way for new kinds of cutting-edge IoT businesses to emerge. Here we provide an architectural concept for a networked IoT network that utilizes both the cloud nor the edge, along with a CD procedure structure for customized SaaS applications operating on edge nodes. In a precision agricultural case study, the conceptual framework and CD process sequence are both implemented. The famous container orchestration platform Kubernetes is discussed in [19] by the writers. To be more specific, they offer two additions. Their experimental campaign to examine the effect of WAN connectivity on vanilla Kubernetes began with a discussion of the outcomes. Their second section is an examination of current efforts to update Kubernetes in a way that takes geo-distribution into account more effectively. In their study [20], researchers provide a novel architecture for reconfigurable clusters that can supply virtual and physical resources that are prepared to use cloud-native DevOps services. In addition, they go into depth on how the reconfigurable cluster was integrated into K-ONE Playground's actual architecture. They conclude by demonstrating its viability via operations and real-world instances of developing cloud-native services.

## III. PROPOSED WORK

### 1. Problem Statement

Machine learning algorithms, by analyzing data trends and detecting abnormalities, may significantly increase the security of blockchain networks. To aid in the prevention of assaults and unauthorized access, machine learning can detect possible cyber dangers, fraudulent actions, and unusual behaviors.

The following are the goals of this DevOps integration:

**Improved Data Analysis and Insights**

There is a lot of data generated by blockchain, but it may be difficult to understand and draw useful conclusions from it. Organizations may benefit from machine learning's fast processing and analysis of this data by gaining useful business knowledge and making data-driven choices.

**Efficient Decision-Making**

Within blockchain-based systems, decision-making processes may be optimized using machine learning algorithms. Machine learning, for instance, may examine voting patterns

International Journal of Scientific Research & Engineering Trends
Volume 10, Issue 1, Jan-Feb-2024, ISSN (Online): 2395-566X

and past results in autonomous decentralized organizations (DAOs), allowing for better-informed and consensus-driven decision-making.

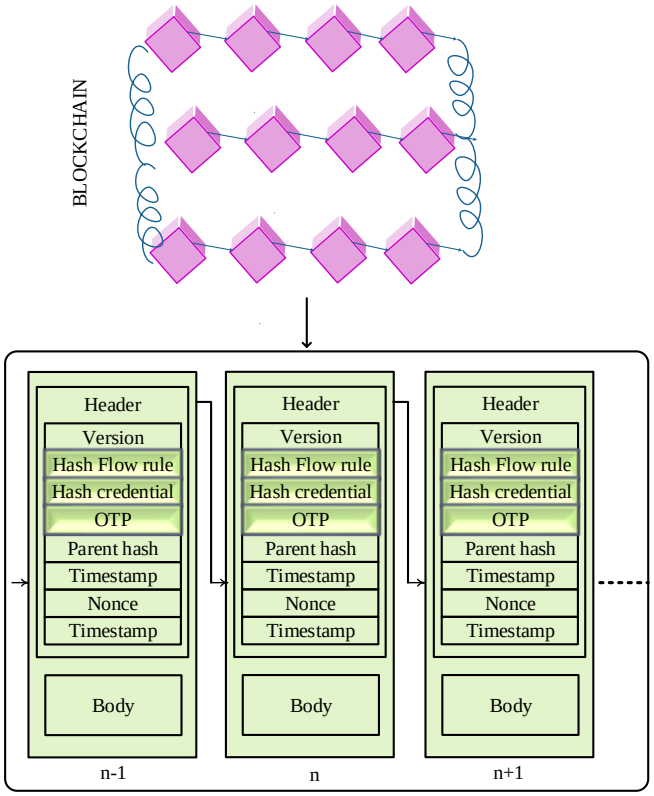


Fig.1. Blockchain Model

**Scalability and Performance**

By optimizing consensus methods and blockchain protocols, machine learning may boost performance and scalability. Improvements to the blockchain network's efficiency and throughput may be proposed by machine learning algorithms after they examine past data and network performance.

## 2. System Model

Figure 1 shows the general approach used in this investigation. We started by gathering the dataset and doing any preparation that was required. Datasets for training and testing were subsequently separated from one another. Only the training data were sampled; the test data were left unaltered. Individual as well as ensemble machine learning classifier were trained using the sampled data. Accuracy, TPR, FPR, as well as ROC-AUC score were among the assessment measures used to evaluate the models using the independent test data.

## 3. Dataset Details

We retrieved the information on bitcoin transactions from the IEEE Information Portal. There are 30,248,134 samples in all, with 30,248,026 marked as negative, meaning they do not include any fraudulent transactions. Contrarily, there are a mere 108 examples that are harmful. Consequently, the dataset is obviously quite skewed. In order to uncover insights within the dataset, Exploratory Data Analysis (EDA) was carried out. To determine whether an exchange of Bitcoins is unusual, there are twelve characteristics that might be labeled. A 1 indicates an abnormal transaction, whereas a 0 indicates a nonanomalous one.

Due to the disproportionately small number of fraudulent Bitcoin transactions compared to legitimate ones, the dataset is skewed. As a result, machine learning classifiers are skewed in favor of the dominant group. Classification accuracy may seem good most of the time, but there's usually a big gap between TPR and FPR numbers, which means the models aren't great at identifying outliers. A solution to this problem is to use either pre-built or user-defined sampling strategies to even out the dataset before training the classification algorithms. It is common for there to be few positive examples in situations involving money laundering, fraud detection, or anomaly detection. Therefore, undersampling methods might be useful for redistributing data points in a dataset and putting an emphasis on precise positive case identification. If the minority class has a very small number of positive examples or anomalies, however, the undersampling strategy may provide a small dataset that ML classifiers can be trained on. In contrast, over-sampling techniques try to bring the minority class's instance count up to par with the majority class's. A mix of majority and minority surveys is used to generate false data, although these strategies may still be useful.

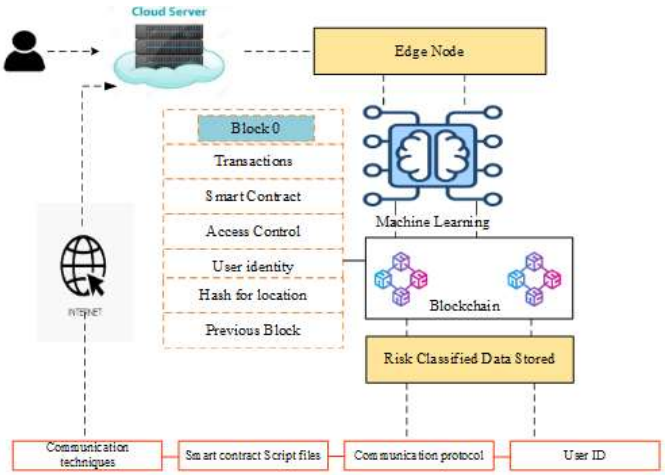


Fig.2. Flowchart for Proposed Work

Near-Miss Sampling: helps keep the dataset in check. The method mitigates the disparity by keeping a subset of majority-class instances that are near to minority-class examples. In order to keep relevant samples intact, this down-sampling method chooses examples according to how close

they are to the minority class. Near-Miss uses metrics like the Manhattan distance and the Euclidean distance to determine how far away each situation in the minority class is from every instance in the majority class. The first step of this technique is to find the k most similar cases in the majority class for each situation in the minority class. Undersampling severity is controlled by the hyperparameter k, the value of which is usually specified. Since we assigned it the number 1, we'll refer to it as NearMiss-1. In the end, it creates an under-sampled dataset by merging the minority class's instances with the chosen majority class instances.

**4. Risk Classification using Machine Learning**

We gather IP, Time Stamps, User Agent, Display Resolutions, along with Mouse and Keyboard Dynamics to classify risks.

IP Address or Internet Protocol Every device that can be accessed over a network is given a unique identifier, or IP address. Using the user's IP address, geolocation software may ascertain the user's nation, region, city, and, in many cases, ZIP code. We get IP address-based geographic location data via an external API.

It's the user agent. The user agent conforms to the requirements to determine the browser's identity (Browser ID). Details about the user's OS, browser, language, and current version are all part of UA. This string specifies the software user agent that made the request, including details about the program, operating system, vendor, and version of the software.

This is the format for user agents: <product > / <product − version ><comment > All of the major web browsers on desktop computers and mobile devices may use UA, including Chrome, Edge, Mozilla Firefox, Internet Explorer, Safari, Opera, Android WebView, Firefox for Android devices Opera for Android, Safari for iOS, Samsung Internet, and Internet Explorer. Because it evolves with software and hardware, the number of possible UA combinations can go into the millions.

**Language:** A user's preferred language in their browser is an aspect of their identification. Nevertheless, considering that a user may be fluent in more than one language, this trait may not pose a significant threat.

Resolving Display Issues A piece of hardware that doesn't directly relate to the user; yet, further checks could be necessary if this feature isn't functioning as intended.

**Time Stamps:** Daytime and nighttime are the most popular times for clients to do their job. For instance, we would see it as more dangerous if the client attempted to access resources among 2A.M. and 3A.M. rather than during typical business hours. Additional verification may be required during access times that are associated with high risk levels or when time drift is a factor; this is because the time zone connected with an IP address may vary from the time zone configured in the browser.

Mouse and Keyboard Dynamics: When compared to physiological biometrics, behavioural biometrics, including mouse and keyboard dynamics, are not as trustworthy. Due to the fact that it is dependent on several other variables, including mood, the impact of drugs, and ergonomics, it exhibits greater levels of variability. The FAR and FRR, or false acceptance and rejection rates, can go up as a result of this. We may divide the dynamics of the keyboard into eleven categories based on key presses and releases, and the dynamics of the mouse into ten categories based on movements and clicks, all of which are biometrics. Typically, we extract mouse characteristics from groups of 30 consecutive mouse events, and for the keyboard, we need to record a minimum of 10 consecutive keystrokes.

Canvas Fingerprinting: Browser, OS, as well as installed graphics hardware are the primary components of the fingerprint. It is often used for user tracking, however it does not distinguish them. Two kind of requests are identified by our system: registration and log-in. A database is maintained to house the characteristics that are obtained during registration, which serve as the primary point of reference.
Class 0 reference data includes the user information gathered upon registration. Statistics pertaining to risk level and data collected from users upon registration are used to build the training model. The class corresponding to the registered deviation is translated from ip by the surjective function ip.

$$ip \rightarrow \{0,1,2,3\}$$

Another function does the same thing for time; it only divides it into the day, night, and evening.

$$\text{" time "} \rightarrow \{\text{" day,evening,night "}\}$$

To check whether the supplied text is identical to the original or not, the translation of user agents (UA), language (L), as well as resolution (R) is set to true.

$$UA,L,R \rightarrow \{T,F\}$$

The development of different models makes use of the Canvas Fingerprinting feature and the Mouse as well as Keyboard Dynamics attribute.

Ordinal, numerical, and categorical data are the three most prevalent kinds used in ML. One way to indicate hometown is using categorical data. Some numerical data is discrete, like 0 or 1, whereas other data is continuous, like 2.32324. In ordinal data, both numerical and category information is included;

while the data is categorized, the numerical values assigned to those categories also have significance. Ordinal data is best shown by a time interval, where the values 0–6 indicate the hours between 6–6:30, 1–12 represent the hours between 6–12:30, and 2–6–12–30 represent the hours between 12–6:30.

A variety of methods may be used to improve blockchain security via the use of machine learning algorithms. To make blockchain networks more resistant to security attacks, these algorithms use AI and data analysis to spot trends, identify outliers, and provide forecasts. Presented below are some suggested machine learning algorithms with potential use in blockchain security.

By facilitating categorization tasks and strengthening the identification and prevention of counterfeit or malicious activity, supervised learning algorithms like Random Forests (RF) and support vector machine algorithms (SVM) are crucial to blockchain security. Among the many supervised learning algorithms used for blockchain security, support vector machines (SVMs) stand out. When it comes to binary classification jobs, SVMs really shine. These tasks require you to classify transactions as either genuine or fraudulent. Support vector machines (SVMs) aim to minimize classification errors while maximizing the division of data points by building a hyperplane that divides the two categories in a highly dimensional feature space. SVMs are great in situations when the data isn't linearly separable, and they have a strong theoretical basis. In order to accomplish both non-linear and linear classification tasks, kernel functions improve the management of high-handling feature spaces.

RF makes forecasts using an ensemble learning approach that employs several decision trees. This algorithm's decision trees are trained using a data subset and utilize random feature choices. It is possible to get the final forecasts by merging the predictions from each tree. Due to its resilience and capacity to process high-dimensional data, RF has promise in this area. Regression and classification applications are well-suited to this method. Through the investigation of security-related characteristics and patterns, RF is very successful in blockchain networks in establishing the veracity of transactions. Through analysis of their relationships and activities, RF may also identify nodes that pose a threat.

When comparing performance and interpretability, SVM and RF are both clearly superior. When it comes to managing big and varied datasets, RF really shines, and SVMs are known for their mastery of complicated data and appropriate decision bounds. When used to blockchain security applications, both algorithms provide trustworthy results. Keep in mind that such supervised learning algorithms can only work as well as the training data they are fed. To train the models correctly, labelled datasets with instances of both good and bad transactions and locations are required.

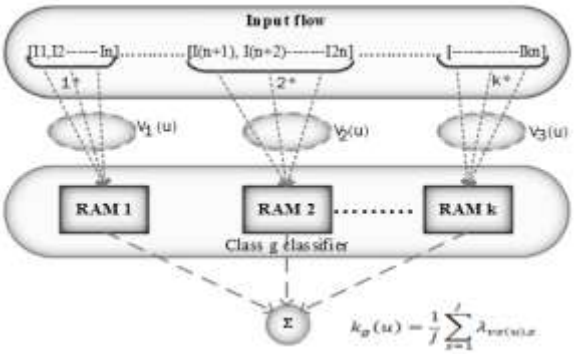

Fig. 3: Classification Process

## IV. RESULTS & DISCUSSION

Here, we evaluate the suggested ensemble models using under-sampled and over-sampled data. A thorough comparison of ensemble classifiers with single classifiers is also provided here. First, we set up the experimental setup, and then we provide the results that analyze the model's performance in many aspects of Bitcoin anomaly detection. Extensive evaluations of performance have been carried out using a dataset that includes both typical and unusual Bitcoin transactions. In Section 3, we provide a brief description of the dataset. Several Python modules, including scikit-learn, NumPy, and Pandas, were used to make performance assessment easier. Using 12.68 GB of RAM as well as 225.83 GB of storage, the Python program was ran on Colab Pro.

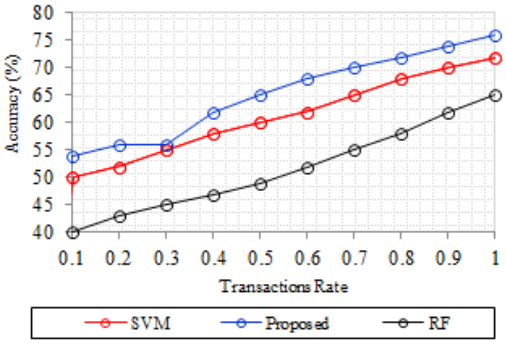

Fig.4. Accuracy Performance

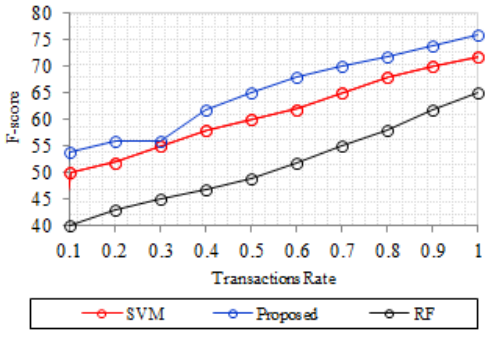

Fig.5. F-score Performance

Fig.6. Loss Performance

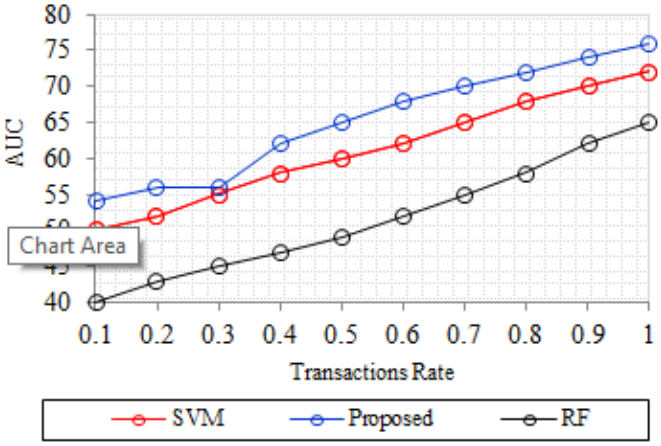

Fig.7. AUC Performance

Figures 4, 5, 6, and 7 show the results of an investigation of the performance of various edge-based algorithms that are in sync with Blockchain along with additional technologies. When it came to protecting data in a Blockchain-based and edge computing setting, many methods were effective. Global loss, f1-score, area under the curve (AUC), and global accuracy are some of the criteria used to compare these methods.

Compared to the previous algorithmic model, the one suggested here employing Edge Computing, the Blockchain consensus technique, with a voting classifier using SVM, Random Forest, K-Neighbours, and Logistic Regression clearly outperforms it. The writers mapped the data block with data that was shared between themselves and shared them with several clients.

**Discussions**

Smart contracts and blockchain network resource allocation may both be enhanced by combining blockchain with machine learning. In order to enhance the security features of smart edge computing devices, the authors used a consensus method based on Blockchain with Machine Learning approaches. The authors acquired the dataset that included data that was vulnerable, data that was exposed, data that had been retrieved, and so on. In addition, a number of optimization methods like as SGD, Momentum SGD, RMSprop, as well as Adam were used by the writers, who employed a range of classification approaches such as Naıve Bayes, Random Forests, K-Neighbors, and SVM. In addition, the scientists used ensemble classification methods to integrate these classifiers on many data blocks, and the results were greatest when the ensemble classifier was used. The Ensemble classifier improved accuracy to 92% from 87% for the best classifier in the framework, the K-Neighbours Classifier. Following their training, the authors evaluated their vulnerability after implementing two-factor authentication. In addition, the authors used a multilayered perceptron model to implement federated instruction in an intelligent Edge computing-based environment, and they delivered the data in batches to several clients. In conclusion, the suggested algorithm was trained using a 1.51 global loss and a 96% global accuracy.

**Case Studies / Applications of Blockchain**

A new paradigm in computing, edge computing opens the door to a plethora of cutting-edge software programs. There are several ways to strengthen the safety of the Internet of Things (IoT), or commonly used smart devices, according to the idea that underpins the suggested approach. Therefore, the paper's uses are various, and they are listed below:

- The consensus method of blockchain enables effective secured lines and the use of mobile edge computing to protect transactional data makes banking the initial application that appropriately specifies this behavior.
- Patient data can be kept safe and model training can be enhanced in terms of both global and local differential privacy with the help of better medical wearables and less data poisoning.
- An autonomous vehicle driving system may acquire the methods used to construct local and global model training frameworks in this article. Therefore, in regions that have been prepared for geographic models, it will enhance the quality of driving.
- The offered strategies may be used to improve cloud-based gaming experiences, leading to a secure and enjoyable gaming environment.

## V. CONCLUSION

Our system's risk assessment is a machine learning model that estimates the risk level of each authentication attempt based on digital identity variables like IP address, browsers user agent, and user behavior. Using a subset of login data variables, we validated our system and built a risk classifier models to determine the amount of danger posed by users. Therefore, a way to train the shared data is via the idea of

machine learning. Under the watchful eye of two-factor authentication, data security was precepted in a dataset that included several exposed, vulnerable, recovered, and protected pieces of information. Data and security vulnerabilities in smart computing edge devices, as well as their fixes, are covered in this study. Machine learning methods, including various classifiers and optimization algorithms, plus the Blockchain consensus approach provide data confidentiality in the suggested model. In addition, the authors used an edge computing setting to implement the suggested techniques by sending data in several batches to various customers. Consequently, the use of Blockchain servers ensured that client anonymity was preserved. In addition, the writers used the federated learning method to train separate batches of client data. This study presents the outcomes of a training model that utilizes Blockchain technology in an edge-based technology setting.

## REFERENCES

1. ((2022). Development, Operation, and Quality Assurance of Trustworthy Smart IoT Systems Periodic Reporting for period 2 - ENACT (Development, Operation, and Quality Assurance of Trustworthy Smart IoT Systems).
2. Song, H., Dautov, R., Ferry, N., Solberg, A., & Fleurey, F. (2022). Model-based fleet deployment in the IoT–edge–cloud continuum. Software and Systems Modeling, 21, 1931 - 1956.
3. Singh, S., Bharti, A.K., Pandey, H., Yadav, R.K., Sharma, D., & Shanker, N. (2023). Towards Automated and Optimized Security Orchestration in Cloud SLA. International Journal on Recent and Innovation Trends in Computing and Communication.
4. Liu, P., Silva, D.D., & Hu, L. (2021). DART: A Scalable and Adaptive Edge Stream Processing Engine. USENIX Annual Technical Conference.
5. Farchi, E., & Route, S. (2023). Quality Engineering for Agile and DevOps on the Cloud and Edge. ArXiv, abs/2302.03651.
6. Oztoprak, K., Tuncel, Y.K., & Butun, I. (2023). Technological Transformation of Telco Operators towards Seamless IoT Edge-Cloud Continuum. Sensors (Basel, Switzerland), 23.
7. López-Viana, R., Díaz, J., & Pérez, J.E. (2022). Continuous Deployment in IoT Edge Computing: A GitOps implementation. 2022 17th Iberian Conference on Information Systems and Technologies (CISTI), 1-6.
8. Judvaitis, J., Balass, R., & Greitans, M. (2021). Mobile IoT-Edge-Cloud Continuum Based and DevOps Enabled Software Framework. J. Sens. Actuator Networks, 10, 62.
9. Sicari, C., Carnevale, L., Galletta, A., & Villari, M. (2022). OpenWolf: A Serverless Workflow Engine for Native Cloud-Edge Continuum. 2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), 1-8.
10. Kumara, I., Mundt, P., Tokmakov, K., Radolovic, D., Maslennikov, A., González, R.S., Fabeiro, J.F., Quattrocchi, G., Meth, K.Z., Di Nitto, E., Tamburri, D.A., van den Heuvel, W., & Meditskos, G. (2021). SODALITE@RT: Orchestrating Applications on Cloud-Edge Infrastructures. Journal of Grid Computing, 19.
11. López-Peña, M.A., Díaz, J., Pérez, J.E., & Humanes, H. (2020). DevOps for IoT Systems: Fast and Continuous Monitoring Feedback of System Availability. IEEE Internet of Things Journal, 7, 10695-10707.
12. Tang, B., Zhang, X., Yang, Q., Qi, X., Alqahtani, F., & Tolba, A.M. (2023). Cost-optimized Internet of Things application deployment in edge computing environment. International Journal of Communication Systems.
13. Khalyly, B.E., Belangour, A., Erraissi, A., & Banane, M. (2020). Devops and Microservices Based Internet of Things Meta-Model.
14. Nitto, E.D., Gorroñogoitia, J., Kumara, I., Meditskos, G., Radolovic, D., Sivalingam, K., & González, R.S. (2020). An Approach to Support Automated Deployment of Applications on Heterogeneous Cloud-HPC Infrastructures. 2020 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 133-140.
15. López-Viana, R., Díaz, J., Díaz, V.H., & Martínez, J. (2020). Continuous Delivery of Customized SaaS Edge Applications in Highly Distributed IoT Systems. IEEE Internet of Things Journal, 7, 10189-10199.
16. Manaouil, K., & Lèbre, A. (2020). Kubernetes and the Edge?
17. Shin, J., & Kim, J. (2020). K-ONE Playground: Reconfigurable Clusters for a Cloud-Native Testbed. Electronics.