Article

Not peer-reviewed version

Trust Management and Resource Optimization in Edge and Fog Computing Using the CyberGuard Framework

Ahmed M. Alwakeel and Abdulrahman Alnaim *

Posted Date: 5 June 2024

doi: 10.20944/preprints202406.0210.v1

Keywords: cloud computing; edge computing; fog computing; blockchain; trust management

Article

Trust Management and Resource Optimization in Edge and Fog Computing Using the CyberGuard Framework

Ahmed M. Alwakeel [1] and Abdulrahman Alnaim [2,*]

[1] Faculty of Computers & Information Technology, University of Tabuk, Tabuk 71491, Saudi Arabia; aalwakeel@ut.edu.sa
[2] College of Business Administration, King Faisal University, Al Ahsa 31982, Saudi Arabia
* Correspondence: alnaimak@gmail.com

Abstract: The growing importance of edge and fog computing in the modern IT infrastructure is driven by the rise of decentralized applications. However, resource allocation within these frameworks is challenging due to varying device capabilities and dynamic network conditions. Conventional approaches often result in poor resource use and slowed advancements. This study presents a novel strategy for enhancing resource allocation in edge and fog computing by integrating machine learning with blockchain for reliable trust management. Our proposed framework, called CyberGuard, leverages blockchain's inherent immutability and decentralization to establish a trustworthy and transparent network for monitoring and verifying edge and fog computing transactions. CyberGuard combines the Trust2Vec model with conventional machine learning models like SVM, KNN, and Random Forests, creating a robust mechanism for assessing trust and security risks. Through detailed optimization and case studies, CyberGuard demonstrates significant improvements in resource allocation efficiency and overall system performance in real-world scenarios. Our results highlight CyberGuard's effectiveness, evidenced by a remarkable accuracy, precision, recall, and F1-Score of 98.18%, showcasing the transformative potential of our comprehensive approach in edge and fog computing environments.

Keywords: cloud computing; edge computing; fog computing; blockchain; trust management

1. Introduction

The proliferation of Internet of Things (IoT) devices and the emergence of decentralized computing paradigms, such as edge and fog computing [1], have dramatically transformed the landscape of information technology. As a result of these advancements [2], a new era of computing has begun, one that is characterized by the efficient processing of data at the network's edge, closer to data sources and end users [3]. Even though these technologies increase productivity and reduce delay, resource management and trust remain challenging issues. Resource allocation is a major problem in edge/fog computing systems. These environments consist of a large variety of heterogeneous devices with different networking and processing capabilities.

This variability frequently makes it difficult for traditional resource allocation approaches to adjust, which has a negative impact on resource utilization and system performance [4]. To solve these difficulties, researchers have looked into how to integrate blockchain technology and machine learning models into edge and fog computing environments. A powerful answer for managing trust is provided by blockchain, which ensures that decisions about how to allocate resources are based on accurate and unchangeable facts. Blockchain has an immutable, decentralized ledger. Machine learning, on the other hand, provides the capability to evaluate and adapt to the dynamic nature of these situations. The general method for allocating resources efficiently is shown in Figure 1:

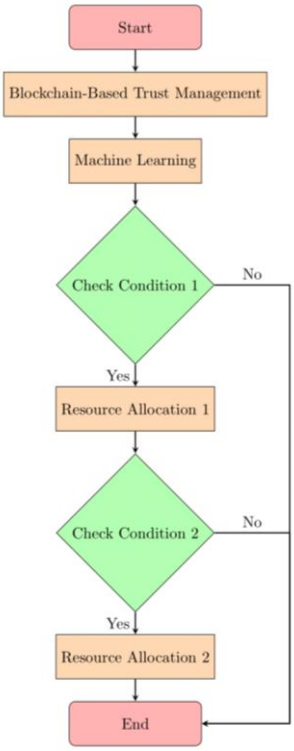

Figure 1. generic flow of optimizing the resource allocation.

An example of a general flowchart or design for optimizing resource allocation is shown in Figure 1. In order to make the most use of their time, money, and people, businesses and individuals often resort to such a graphical representation. It provides a high-level overview of the planning and decision-making required for optimal resource utilization.

However, there are certain limitations to current research [5]. The relationship between block chain and machine learning in resource allocation for edge/fog computing is not well understood, leaving room for more accurate models that can incorporate both technologies for enhanced performance. Although the pairing of machine learning with blockchain has been considered in a number of studies, a complete model that can manage resource allocation with a high degree of accuracy, precision, and efficiency has not yet been achieved.

In the rapidly evolving landscape of distributed computing, the fusion of blockchain technology and fog computing has emerged as a promising paradigm, offering novel solutions to the challenges posed by decentralized applications. In this context, we present the CyberGuard model, a pioneering approach that seamlessly integrates blockchain-based trust management with the inherent advantages of fog computing. Fog computing, as an extension of cloud computing, brings computational resources closer to the edge of the network, enabling faster processing and reduced latency for applications. This proximity to end-users is particularly advantageous in scenarios with resource-constrained devices, such as those found in the Internet of Things (IoT). However, the dynamic and decentralized nature of fog computing environments demands robust trust management systems to ensure the integrity and security of transactions. The CyberGuard model addresses this demand by leveraging the immutable and decentralized nature of blockchain technology. Our approach establishes a transparent and trustworthy network for monitoring and validating business transactions within fog computing environments. This integration of blockchain ensures that decisions regarding resource allocation are grounded in verifiable and secure data, mitigating the risks associated with fraudulent or malicious activities. This paper aims to provide a comprehensive understanding of the CyberGuard model, elucidating its foundational principles, design considerations, and the symbiotic relationship between blockchain and fog computing.

 doi:10.20944/preprints202406.0210.v1

Through a meticulous exploration of our model, we showcase how it surpasses existing approaches by enhancing trust, security, and efficiency in resource allocation.

In view of these constraints, this work proposes the "Cyber Guard Model," a novel and comprehensive approach for resource allocation optimization in edge and fog computing settings. The Cyber Guard Model integrates cutting-edge machine learning algorithms with block chain-based trust management to revolutionize resource allocation, enhance system security, and increase overall system performance. In the sections that follow, we analyze the procedures, experiments, and results of this revolutionary approach, showcasing its applicability to actual circumstances. The contributions of this study are as follows:

- Our work represents a significant contribution by seamlessly integrating techniques from distributed frameworks for AI, cyber-physical systems, and smart blockchain.
- We introduce a novel holistic model, CyberGuard AI, which stands out in its approach to resource allocation in edge/fog computing environments. Unlike existing models, CyberGuard AI takes advantage of the inherent properties of blockchain, such as immutability and decentralization, to establish a trustworthy and open network for monitoring and confirming edge/fog business transactions.
- CyberGuard AI incorporates Trust2Vec, a unique element not commonly found in existing approaches. This integration leverages support vectors to enhance the trust score predictions, thereby improving the decision-making process for resource allocation.
- Our study goes beyond traditional resource allocation methods by employing machine learning approaches for dynamic and efficient resource management. By utilizing massive volumes of data from edge/fog devices, our model adapts to new information and requirements, making the most effective use of available computing power, network bandwidth, and storage space.
- The ensemble model, enhances resource allocation predictions by combining results from multiple machine learning algorithms, including Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Random Forests. This ensures a more robust and reliable estimation of trust security danger compared to single-model approaches.
- We provide a thorough performance evaluation of our proposed model through rigorous case studies and simulations. The results showcase the efficacy and viability of our approach in various real-world circumstances, demonstrating its superiority in resource allocation within edge/fog computing environments.

While existing models may touch upon blockchain, CyberGuard AI stands out by placing blockchain at the core of trust management. It significantly reduces the risks of fraudulent or malicious attacks by ensuring that resource distribution decisions are based on immutable and trustworthy data. Unlike traditional resource allocation methods, our model leverages machine learning to dynamically adapt to changing conditions. This adaptability ensures efficient resource usage across network nodes, contributing to improved system performance. SecuroBlend's ensemble learning approach distinguishes our work from models that rely on a single algorithm. The combination of SVM, KNN, and Random Forests enhances the robustness of our predictions, particularly in the context of resource allocation. In summary, our contributions lie in the seamless integration of decentralized frameworks, the introduction of novel models like CyberGuard AI and SecuroBlend, and the utilization of blockchain and machine learning for effective and dynamic resource allocation. The demonstrated superiority through rigorous evaluations further establishes the novelty and relevance of our work in the field.

2. Related Work

Recent growth in edge and fog computing has stimulated significant research efforts in distributed systems trust management and resource allocation [6]. This section summarizes major works that have advanced edge/fog computing and investigated cutting-edge methods for resource allocation and trust management.

2.1. Blockchain Integration for Trust-Based Resource Allocation

For enhancing resource allocation trust in edge and fog computing situations, block chain technology has generated a lot of interest. In a groundbreaking study, researchers looked into the usage of block chain in edge/fog computing [6], looking into its potential to boost trust in resource allocation and, ultimately, produce a more secure and decentralized edge environment. Another study [7] proposed a novel method for allocating resources for fog computing that uses block chain to build a decentralized and immutable record, enhancing both resource usage and trust. Research has suggested a block chain-based trust management system and an architecture for allocating edge computing resources [8]. It has been shown that this distributed ledger can enhance real-time resource allocation and edge computing resource consumption. In a study [9] on Mobile Edge Computing (MEC), a trust architecture based on block chains was presented. It successfully thwarts self-serving edge attackers and leverages reinforcement learning-based CPU allocation for improved computing efficiency. According to research [10], block chain technology may be used to protect and optimize resource allocation in edge/fog computing, emphasizing the advantages of decentralization in enhancing trust, security, and resource efficiency. This is significant. Figure 2 depicts the block chain integration for trust-based resource allocation.

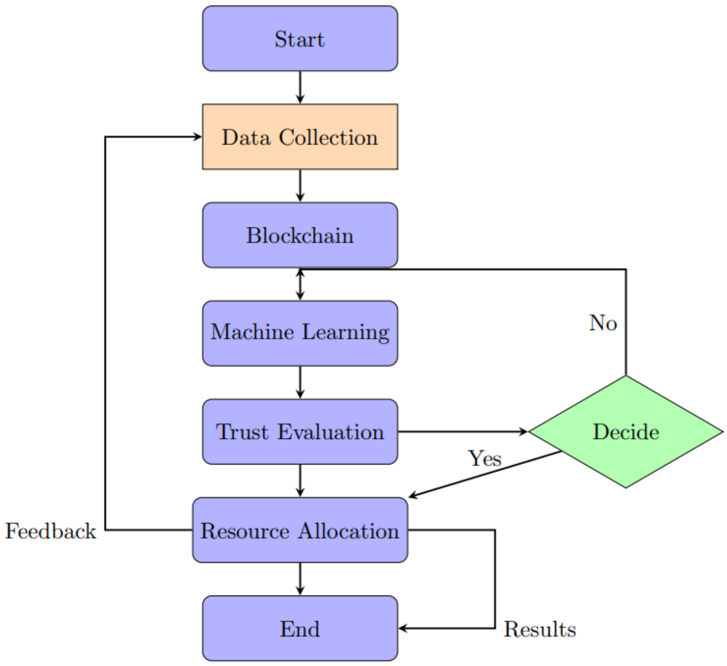

Figure 2. Block chain Integration for Trust-Based Resource Allocation.

Integrating blockchain technology for trustworthy resource distribution is shown in Figure 2. To ensure fairness, safety, and confidence in resource distribution, blockchain is deployed as a backbone technology here. This likely depicts the use of blockchain technology to enhance trust and accountability in resource management systems, as seen through its application to the fair and dependable allocation of resources among diverse parties or entities.

2.2. Machine Learning-Driven Resource Optimization

Machine learning has developed into a practical method for dynamic resource allocation in edge computing. A study [11] shown how machine learning techniques may be utilized to optimize resource distribution in edge computing environments, which will increase overall effectiveness and performance. An innovative approach was presented in [12], combining data mining and machine learning to identify a more precise resource distribution, to assess the reliability of edge nodes for fog computing. Research [13,14] looked at how different machine learning techniques could be used to assess the dependability of fog computing powered by block chains. The study investigated how machine learning methods that evaluate fog node trust can enhance resource allocation and system performance. Research [15,16] addressed how accurate demand forecasting for cloud computing resource requirements might lead to improved resource allocation, making sure that fog nodes are ready to manage workload shifts. novel models and hybrid techniques [17]. A number of studies have proposed distinct hybrid methodologies and models that combine block chain and machine learning for resource distribution and trust management. A hybrid solution integrating block chain and machine learning was introduced in [18] to address trust issues in edge/fog computing. This technique enhanced participant trust and improved resource allocation choices. Since its debut in [19,20], Trust-as-a-Service (TaaS), which provides trust evaluations as a service utilizing block chain and machine learning, has increased the dependability and efficiency of edge computing ecosystems. The use of block chain technology and machine learning to optimize resource allocation while following to energy-saving rules and promoting greener settings was demonstrated in [21,22], which offered a way for allocating resources in an energy-efficient manner for edge/fog computing. A mechanism for dynamic resource distribution in fog computing was developed using block chain technology [23,24], demonstrating how the immutability and transparency of the block chain may boost resource efficiency in fog computing. A trust-aware architecture for distributing edge computing as-sets was described in [25], using block chain and machine learning to provide real-time trustworthiness evaluations for secure and efficient resource allocation. In conclusion, these studies have had a significant influence on the fields of machine learning-driven resource allocation, edge/fog computing, and block chain integration. As we will examine in the next sections, there is still room for innovation and advancement, which is what our proposed "Cyber Guard Model" aims to do.

Table 1. Comparative Table.

Reference	Technique	Outcome
[1]	RL, Blockchain	Introduces a trust mechanism using RL and blockchain to address selfish edge attacks in MEC.
[2]	Privacy-Preserving Blockchain with Edge Computing	Presents TrustChain, a privacy-preserving blockchain, integrating with edge computing for enhanced trust.
[8]	Decentralized blockchain platform for cooperative edge computing	Introduces CoopEdge, a blockchain-based platform for collaborative edge computing.
[9]	Survey	Provides a comprehensive survey on orchestration techniques in fog computing.
[12]	Blockchain-based banking	Investigates blockchain-based banking solutions.
[13]	Blockchain-based resource allocation model in fog computing	Proposes a resource allocation model using blockchain in fog computing.
[20]	Federated Learning, Blockchain	investigates the potential and pitfalls of integrating federated learning with blockchain in edge computin.
[21]	Blockchain-Based Applications and the Rise of Machine Learning	problems and opportunities for implementing machine learning in blockchain-based smart applications.

3. Methodology

The technique used to optimize resource distribution in edge/fog computing scenarios is thoroughly explained in this section. Our strategy combines machine learning techniques with trust management based on blockchain. We outline the exact procedures for creating, putting into practice, and evaluating the suggested system. This research's major objective is to increase the efficiency,

 doi:10.20944/preprints202406.0210.v1

security, and dependability of resource allocation in distributed systems, especially in the context of edge/fog computing. Given the increase in networked devices, each of which has distinct capabilities and network conditions, an innovative approach that can dynamically allocate resources while guaranteeing reliability and data integrity is becoming increasingly important.

In order to accomplish this, we offer a cutting-edge approach that integrates blockchain and machine learning. Because it is a decentralized and irreversible distributed ledger technology, block chain provides the ideal platform for managing trust relationships and ensuring data authenticity. We propose a block chain-based trust management framework to assist our resource allocation choices. The dependability and openness of this system will serve as the basis for all decisions about the allocation of resources. A major challenge is the vast volume of data that edge and fog sensors create. Here, machine learning algorithms take center stage and make it possible for this data to be automatically examined and evaluated. Machine learning provides dynamic resource allocation, which maximizes the utilization of existing network resources by adapting to changing conditions and requirements. We present the Cyber Guard model as a substantial advancement of our techniques. An effective machine learning technique for predicting levels of trust security danger is the Cyber Guard model. By combining the advantages of several machine learning approaches with Trust2Vec graph embedding, the Cyber Guard model offers better accuracy in anticipating trust security risks. By incorporating the unique insights provided by several machine learning classifiers within the Cyber Guard model, the strategy emphasizes group decision-making. In comparison to the traditional technique of resource distribution, this cooperative strategy represents a major improvement. We focus on combining machine learning, block chain-based trust management, and the incorporation of the Cyber Guard model in our strategy, to sum up. With the help of this complete strategy, resource allocation in edge/fog computing environments is efficient, secure, and adaptable to changing conditions and demands.

Our research methodology is intricately designed to optimize resource distribution in the challenging context of edge/fog computing scenarios. The key focus is on leveraging the inherent advantages of blockchain-based trust management and the adaptability of machine learning models, culminating in the development of our innovative CyberGuard model.

Blockchain-Based Trust Management:

To instill trust and transparency in resource allocation decisions, we employ a blockchain-based framework. The immutability and decentralized nature of blockchain technology form the backbone of our trust management system. Each transaction, pertaining to resource allocation or decision-making, is securely recorded on the blockchain, ensuring a tamper-resistant and auditable trail. This not only enhances the integrity of the decision-making process but also mitigates the risks associated with malicious attacks or unauthorized alterations.

Machine Learning for Dynamic Resource Management:

Our approach integrates machine learning algorithms, including Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Random Forests, within the CyberGuard model. These algorithms are trained on extensive datasets from edge/fog devices, enabling them to dynamically adapt to changing network conditions, device capabilities, and application requirements. The machine learning component ensures that resource allocation decisions are not static but evolve in real-time based on the evolving dynamics of the edge/fog computing environment.

Ensemble Model - CyberGuard AI:

A significant contribution of our methodology is the development of CyberGuard AI, an ensemble model that harnesses the collective intelligence of multiple machine learning algorithms. By combining the results of SVM, KNN, and Random Forests, CyberGuard AI achieves a more robust and accurate prediction of trust scores, resource allocation decisions, and security threat levels. This ensemble approach enhances the overall reliability and performance of resource distribution in edge/fog computing. Figure below shows the flow of proposed work:

3.1. Dataset Description

Records on resource allocation and trust management in edge/fog computing settings make up the dataset used in this study. It includes a wide variety of data from various edge and fog computing nodes, including both qualitative and numerical properties. The main goal of this dataset is to look at how machine learning and block chain-based trust management may be combined to improve the efficiency of resource allocation in edge/fog computing. The table below provides thorough details on each component of the dataset:

Table 2. Dataset Feature Description.

Feature	Description
Device ID	A unique identifier for each edge/fog computing device.
Timestamp	The timestamp indicating the date and time of data collection.
CPU Usage	The percentage of CPU utilization by the computing device at the given timestamp.
Memory Usage	The percentage of memory (RAM) utilization by the computing device at the given timestamp.
Network Bandwidth	How many megabits per second (Mbps) were being used by the network at that precise moment in time.
Data Locality	A categorical feature indicating the locality of the data processed by the device (e.g., Local, Nearby, Remote).
Latency	The latency in milliseconds (ms) for data transmission or processing at the given timestamp.
Energy Consumption	The energy consumption in watts (W) by the computing device at the given timestamp.
Resource Allocation Decision	A binary feature representing the resource allocation decision (1 for successful allocation, 0 for unsuccessful).
Trust Score	A numerical score representing the trustworthiness of the computing device in the network.
Block chain Validation Status	A categorical feature indicating the status of block chain validation for the device (e.g., Valid, Invalid).
Fog Node Type	A categorical feature indicating the type of fog node (e.g., Fog, Edge) where the device is located.
Temperature	The local temperature measured in degrees Celsius where the computer is being used.
Humidity	The relative humidity percentage (%) at the location of the computing device.
Security Threat Level	A scale from low to high that indicates how secure the edge/fog computing environment is.

There are a total of 14 distinct features in the dataset, each of which represents a different aspect of the edge/fog computing environment, trust management, and resource distribution. These components have been thoughtfully designed to aid in achieving the objectives of the study and make it simpler to evaluate the suggested integrated methodology. This integrated approach combines machine learning methods with block chain-based trust management to enhance resource allocation and overall system effectiveness in edge/fog computing environments.

Figure 4 allows us to check the real distribution of the dataset's features. The feature values are displayed on the x-axis, and the occurrence count or frequency is displayed on the y-axis. This graphic facilitates understanding of the breadth and depth of variance for each dataset attribute. A scatter plot matrix with each characteristic displayed in relation to every other feature is illustrated in Figure 3. The distribution of each trait along the diagonal of the matrix is often displayed. For discovering potential connections or patterns between features in a collection, pair plots are incredibly helpful.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 June 2024 doi:10.20944/preprints202406.0210.v1

For each feature in the dataset, box plots are displayed in Figure 3. The median, quartiles, and outliers of a data collection are graphically represented by box plots. This might aid in your understanding of the average and standard deviation for each attribute. The estimated data kernel density is shown in Figure3. They make it simpler to compare data densities at various scales by giving more details on each feature's density and distribution.

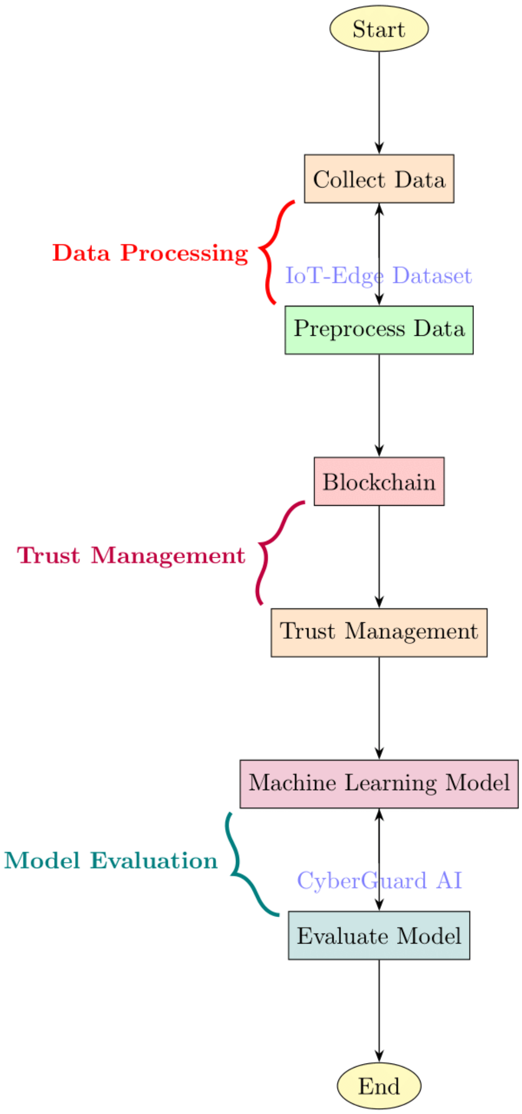

Figure 3. Proposed Working Flow.

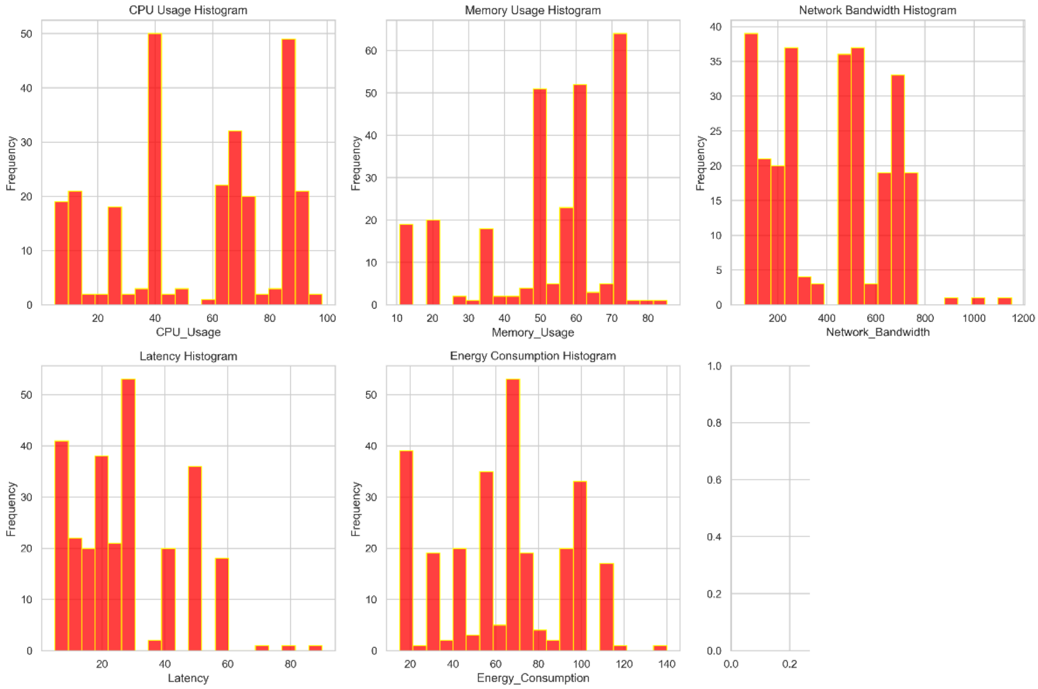

Figure 4. Distribution of features.

In Figure 5, we see a collection of pair plots that illustrate the interplay of all the features in a dataset. These graphs make it possible to see connections between variables, which can help uncover hidden patterns and tendencies in the data.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 June 2024 doi:10.20944/preprints202406.0210.v1

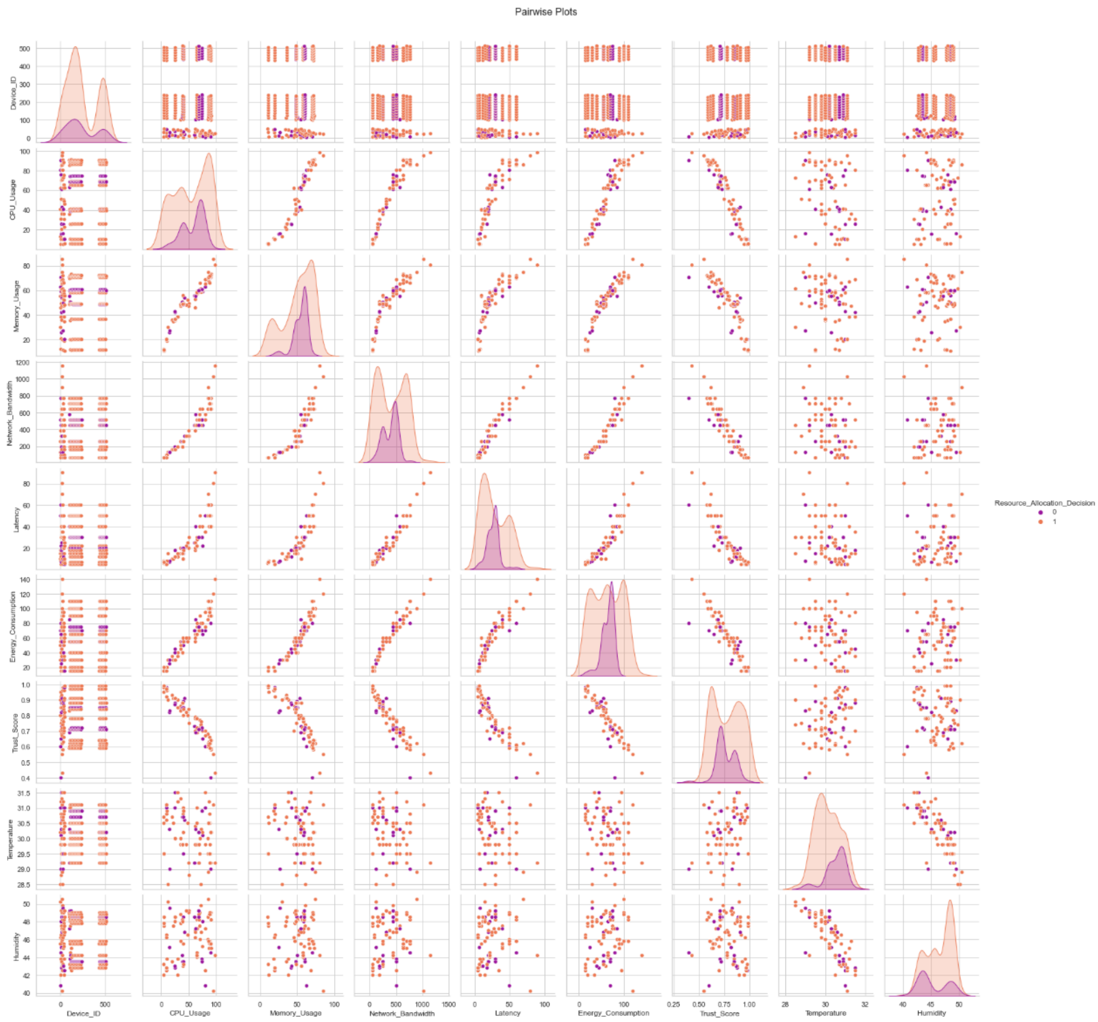

Figure 5. Pair plots of all features.

3.2. Data Pre-Processing

Data preparation is the process of converting raw data into a format suitable for analysis and modeling. It assists in cleaning, organizing, and preparing the data to improve its quality and make it more suitable for machine learning algorithms.

3.3. Feature Engineering

This graph shows the relationships between different dataset features and demonstrates how crucial feature engineering is to helping machine learning models recognize patterns and generate precise predictions. A correlation matrix exposes the web of links between them by displaying which traits are favorably and adversely associated with one another. While traits with a high correlation to other features can be eliminated to avoid multicollinearity during feature selection, traits with a high correlation to the target variable can be very effective predictors.

Relationships between pairs of variables or features in a dataset can be visualized using a correlation matrix, as shown in Figure 6. The magnitude and direction of these associations are usually represented by a color code or numerical value. Understanding the interplay between multiple data qualities is facilitated by this matrix, which indicates whether variables are positively, adversely, or not statistically associated with each other.

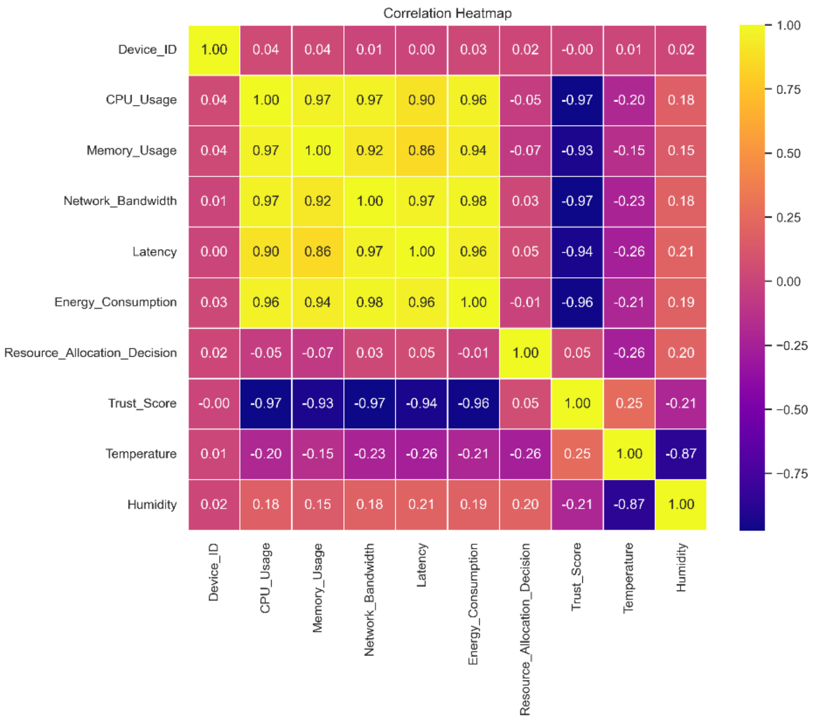

Figure 6. Correlation Matrix.

3.4. Machine Learning Models

3.4.1. Support Vector Machine (SVM)

The reliable Support Vector Machine (SVM) supervised machine learning method is essential for both classification and regression issues. In the ensemble prediction of CyberGuard, SVM is one of the core models.

The Support Vector Machine (SVM) serves as a pivotal model within the ensemble predictions of CyberGuard, contributing to both classification and regression tasks.

The internal structure of SVM, depicted in Figure 7, showcases its components such as support vectors, decision borders, and the margin. Mathematically, the SVM optimization problem can be formulated as follows:

$$\text{Minimize } \parallel w \parallel \text{ (to maximize the margin)}$$

$$\text{subject to the constraint}$$

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i$$

$$\text{for all data points } (x_i, y_i)$$

The inclusion of slack variables (ξi) addresses misclassifications, while the parameter C balances margin maximization and misclassification minimization.

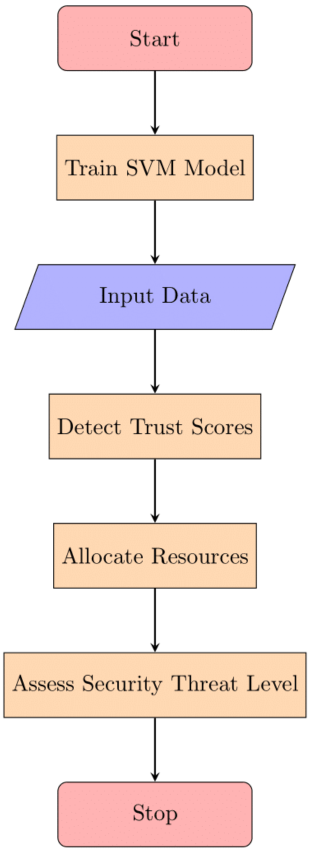

Figure 7. SVM Architecture.

The internal structure of an SVM is shown in Figure 7. For classification and regression, supervised machine learning algorithms like the support vector machine (SVM) are useful. To better explain how SVMs function and how they may be applied to solve certain problems, this picture likely gives a visual depiction of the components and structure of an SVM model, such as support vectors, decision borders, and the margin.

The SVM optimization issue can be written in mathematical notation as:

- Given a training dataset $D = \{(x1, y1), (x2, y2), \ldots, (xn, yn)\}$, where xi is the feature vector and yi is the corresponding class label (-1 or +1).
- Identify the optimum weight the hyperplane that splits the data points into classes and optimizes the margin can be defined by the vector w and the bias term b.
- The objective function is to minimize $||w||$ (to maximize the margin) subject to the constraint $yi(w * xi + b) >= 1 - \xi i$ for all data points (xi, yi).
- The slack variables ξi are introduced to handle misclassifications, and the C parameter controls the trade-off between maximizing the margin and minimizing the misclassifications.

3.4.2. K-Nearest Neighbours

K-Nearest Neighbors (KNN) is a popular supervised machine learning method for both classification and regression. As part of the optimization of edge/fog computing resources, KNN can be used to predict a device's security risk level from its CPU and memory consumption. K-Nearest Neighbors (KNN) is a supervised learning method applied to predict security risk levels in edge/fog computing based on device resource consumption. Figure 8 illustrates the structure of the KNN algorithm, emphasizing its principle of identifying neighbors based on a defined K.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 June 2024 doi:10.20944/preprints202406.0210.v1

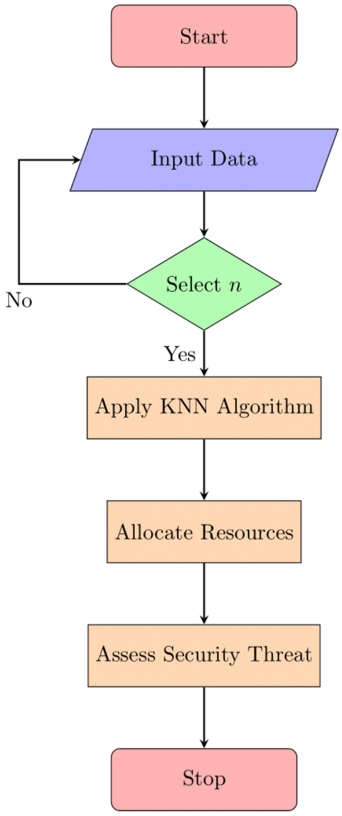

Figure 8. KNN Architecture.

In Figure 8, we see how a K-Nearest Neighbors (KNN) algorithm is structured. In the realm of machine learning, KNN is a supervised technique used for classification and regression. The notion of identifying the nearest neighbors to a data point based on a set value of "K" (the number of neighbors to consider) is likely represented graphically in this picture, showing how KNN works. It may also show how KNN uses the majority class or average value of a data point's K-nearest neighbors to determine that data point's classification or value.

The formula for KNN is as follows:

- As an illustration, consider dataset D, where x stands for a device and y for a type of security risk.
- Calculate the distance between the data points x and d in dataset D using the preferred distance metric.
- Select the K data points that are closest to x as your K nearest neighbors.
- Assign y to x after classifying it as a member of the same group as the majority of its K closest neighbors.
- The optimal way to assign y to x in a regression is to use the mean of the y values of the K nearest neighbors.

3.4.3. Random Forests

Random Forests is an example of a Bayesian machine learning algorithm. The optimization of security-related edge/fog computing resource allocation is one categorization issue that significantly benefits from this approach. Random Forests is popular because it is easy to use, efficient, and accurate when processing high-dimensional data. The internal structure of Random Forests (RF) is elucidated in Figure 9, showcasing its components like decision trees and the ensemble approach. Mathematically, the classification procedure involves computing posterior probabilities and selecting the class label with the highest probability.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 June 2024 doi:10.20944/preprints202406.0210.v1

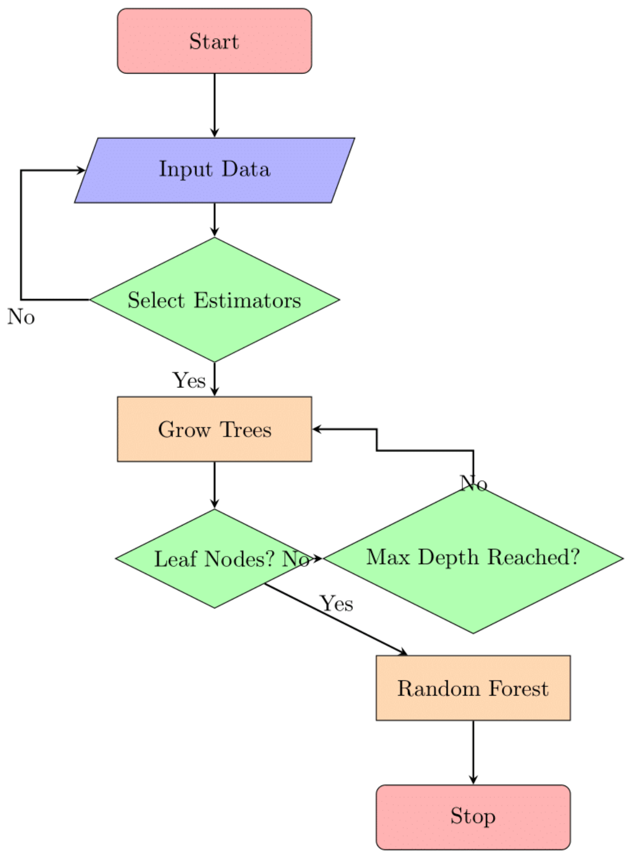

Figure 9. RF Architecture.

The internal structure of a Random Forest (RF) model is depicted in Figure 9. The goal of the ensemble machine learning technique known as Random Forest is to increase prediction accuracy while decreasing overfitting by combining numerous decision trees. It's possible that this diagram illustrates the structure and essential parts of a Random Forest model, including decision trees, feature selection, and the voting method for making predictions. Users are given a better grasp of the inner workings of Random Forest models and their potential applications in a wide range of data analysis and machine learning endeavours.

The Random Forests classification procedure can be expressed mathematically as follows:

- To clarify, we will refer to the "training dataset" as "D," the "input sample" as "x," and the "output class" as "y".
- For each class label c in D, calculate the posterior probability P(y=c|x) using Bayes' theorem and the naive assumption.
- Select the class label c with the highest posterior probability P(y=c|x) as the predicted class for the input sample.

Massive datasets with high-dimensional characteristics can be processed using Random Forests with a low overhead of processing. It is used well in a variety of applications, including text categorization, spam filtering, and sentiment analysis, because to its simplicity of use and respectable performance, particularly when the naive assumption is suitable for the data.

3.4.4. CyberGuard Model

The Cyber Guard model, an ensemble model that integrates various machine learning approaches, can be used to anticipate security threat levels in edge/fog computing resource allocation optimization more precisely and reliably. It integrates the several forecasts from each of its individual models to create a single, accurate forecast. The algorithm details the steps from initializing datasets to predicting trust scores, resource allocation decisions, and security threat levels. It incorporates

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 June 2024 doi:10.20944/preprints202406.0210.v1

aspects from SVM, KNN, and RF, showcasing the synergy of these models in CyberGuard. Let's examine the Cyber Guard model in detail:

- As an illustration, consider dataset D, where x stands for a device and y for a type of security risk.
- In Cyber Guard, use Grid SearchCV to do hyper parameter tuning to ascertain the ideal values for each algorithm's base model (SVM, KNN, and RF).
- It is recommended to hyperparameter-tune each base model and then train it on dataset D.
- For a given input sample x, the level of security risk is predicted by each base model separately$\left(y_{pred_{svm}}, y_{pred_{knn}}, y_{pred_{nb}}\right)$.
- CyberGuard combines the predictions $\left(y_{pred_{svm}}, y_{pred_{knn}}, y_{pred_{nb}}\right)$ of all base models using voting='hard'.

Class 'y' at the output is decided by a vote of the base models, with the winner being the class that was predicted the most frequently.

Shown in Figure 10 is the "Cyber Guard Model." This diagram probably depicts the framework or constituent parts of a model developed for use in cyberspace. The term "cyber security" refers to the coordinated efforts of several entities to identify and neutralize cyber threats and improve the safety of computerized infrastructures. In the context of cyberspace, the graphic summarizes the operation of this paradigm [25].

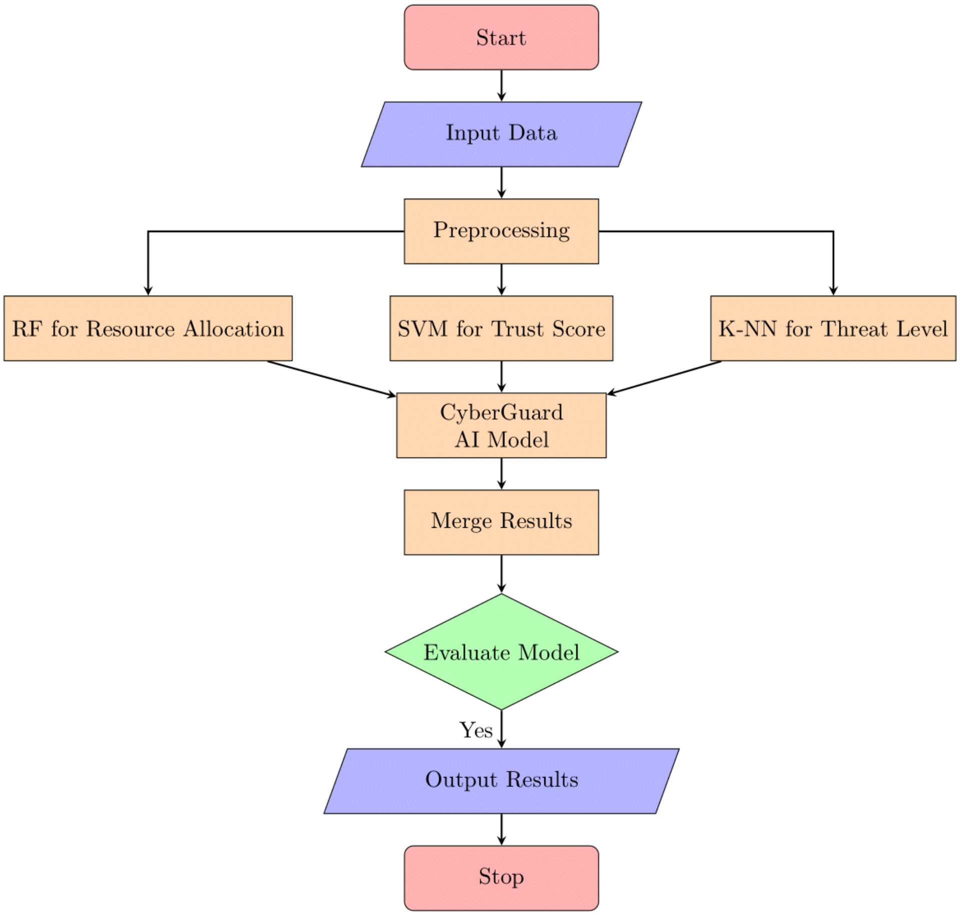

Figure 10. Cyber Guard Model.

Algorithm 1 Mathematical Algorithm for CyberGuard

1. Input:
2. Training dataset $D = \{(x_i, y_i)\}$, where x_i is the feature vector, y_i is the class label, and N is the number of data points.
3. Number of support vectors M, number of neighbors K, number of trees T.
4. Output:
5. Predicted trust score T, resource allocation decision R, and security threat level S for a new data point.
6. Initialize empty lists S_support_vectors, N_neighbors, and T_{trees}.
7. For $m = 1 \; to \; M$:
8. Randomly select a data point (x_i, y_i) from D.
9. Add (x_i, y_i) to S_support_vectors.
10. For k = 1 to K:
11. Select K nearest neighbors of the new data point x_new from D based on a distance metric (e.g., Euclidean distance).
12. Add the K neighbors to N_neighbors.
13. For t = 1 to T:
14. Sample a bootstrap dataset D_t from D with replacement.
15. Initialize a decision tree T_t.
16. While stopping criteria not met (e.g., maximum depth or minimum samples per leaf):
17. Select a random subset of features for node splitting.
18. Find the best feature and split point using a criterion (e.g., Gini impurity or entropy).
19. Split the node into child nodes.
20. Add T_t to T_trees.
21. Predict Trust Score:
22. Initialize an empty list T_predictions to store trust score predictions for each tree.
23. For t = 1 to T:
24. Predict the trust score T_t for the new data point x_new using tree T_t.
25. Add T_t to T_predictions.
26. Aggregate trust score predictions using an aggregation method (e.g., averaging).
27. Return the final predicted trust score T.
28. Predict Resource Allocation Decision and Security Threat Level:
29. Initialize empty lists R_predictions to store resource allocation decisions and S_predictions to store security threat level predictions for each tree.
30. For t = 1 to T:
31. Predict the resource allocation decision R_t and security threat level S_t for the new data point x_new using tree T_t.
32. Add R_t to R_predictions and S_t to S_predictions.
33. Aggregate resource allocation decision predictions using an aggregation method (e.g., majority voting).
34. Aggregate security threat level predictions using an aggregation method (e.g., majority voting).
35. Return the final predicted resource allocation decision R and security threat level S.

The Cyber Guard model takes advantage of the extra advantages of numerous algorithms and associated hyperparameters, which enhances prediction performance compared to utilizing a single model. The combination of SVM, KNN, and RF in Cyber Guard makes it an effective tool for optimizing edge/fog computing resource allocation and enhancing security threat level prediction.

Our evaluation employs a comprehensive approach where each machine learning model (SVM, KNN, Random Forests) and the ensemble model (CyberGuard AI) undergoes rigorous testing using diverse datasets and scenarios. We utilize standard metrics such as accuracy, precision, recall, and F1-Score to assess the performance of each model. Cross-validation and, where applicable, a holdout test set ensure robust evaluations. Furthermore, we conduct comparative analyses to highlight the strengths of the ensemble model in improving resource allocation. The key innovation in our work lies in integrating blockchain for trust management in edge/fog computing. Blockchain operates as

 doi:10.20944/preprints202406.0210.v1

an immutable and decentralized ledger, providing a transparent and secure record of transactions. In our proposed approach, we use blockchain to validate and secure business transactions in the edge/fog environment. Each transaction or decision related to resource allocation is recorded as a block on the blockchain. Smart contracts are employed to automate and enforce trust rules, ensuring that only validated and authorized transactions contribute to the decision-making process. This application of blockchain technology enhances the security and reliability of trust management, making it resilient to fraudulent or malicious activities. The decision to utilize blockchain is justified by its inherent features of immutability, decentralization, and transparency. Immutability ensures that once a transaction is recorded on the blockchain, it cannot be altered, providing a tamper-proof history of decisions.

4. Results and Discussion

The Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Nave Bayes (RF), and ensemble model Cyber Guard are just a few of the machine learning models we use to communicate the results of our work. We contrast and compare each model's precision, recall, and F1-score. We also look into how the model's accuracy is impacted by features and data pretreatment. A sizable dataset that contained information on the distribution of edge/fog computing resources and the seriousness of security risks was used for the trials. After they have been assessed and compared, it will be evident how useful and applicable these models are for predicting the security threat level in fog computing settings. We also look into the Cyber Guard ensemble model, which combines the findings of numerous base models into a single, accurate assessment of the security threat. We anticipate that this comparison will clarify the relative benefits of the various approaches and their possible use in diverse contexts.

Figure 11 shows a comparison of predicted CPU and memory utilization from the Cyber Guard model. Each data point contains the CPU and memory utilization values for the test set. The color of each data point indicates the verified level of security risk connected to that particular combination. The distinction between threat levels and the relationship between CPU and memory usage are made clearer by this depiction in the Cyber Guard model. The data points should create distinct clusters or patterns that correspond to different threat levels if the model is successful in predicting security threat levels based on CPU and memory utilization.

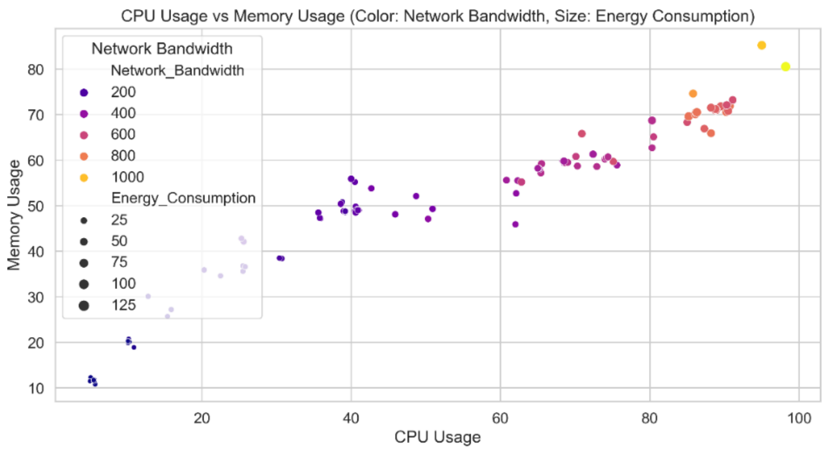

Figure 11. CPU usage vs Memory Usage predicted by Cyber Guard Model.

The distribution of the data locality values across the dataset is depicted in Figure 12. The proximity of data sources to the fog computing nodes is referred to as "data locality". Using this picture, we can investigate the distribution of location values in the data and search for trends. Understanding the distribution of data locality is crucial for optimizing resource distribution in a fog computing system. It assists in locating the best data processing hubs and aids in making intelligent resource allocation decisions.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 June 2024 doi:10.20944/preprints202406.0210.v1

Figure 12. Data Locality Distribution.

Figure 13 compares the predicted trust scores of the Cyber Guard model with the actual status of the data points' blockchain validation. The trust score of each edge/fog node rates the reliability of that node. The block chain's validation status informs whether or not the transactions that have been recorded there are valid. By comparing the predictions to the validation state of the blockchain, we can assess how well the model predicts trust scores. Understanding how trustworthy the Cyber Guard model is in terms of handling trust requires this evaluation.

Figure 13. Trust Score prediction vs Blockchain Validation.

The variety of security concerns present in the data set is depicted in Figure 14. The frequency of each hazard level in the fog computing environment is depicted in this graph. Understanding the spectrum of potential security threat levels is necessary for identifying system vulnerabilities and threats. It is helpful for assessing the relative significance of potential threats and developing strategies to deal with them. This approach also allows us to assess the model's accuracy in predicting security hazard ratings throughout the entire dataset.

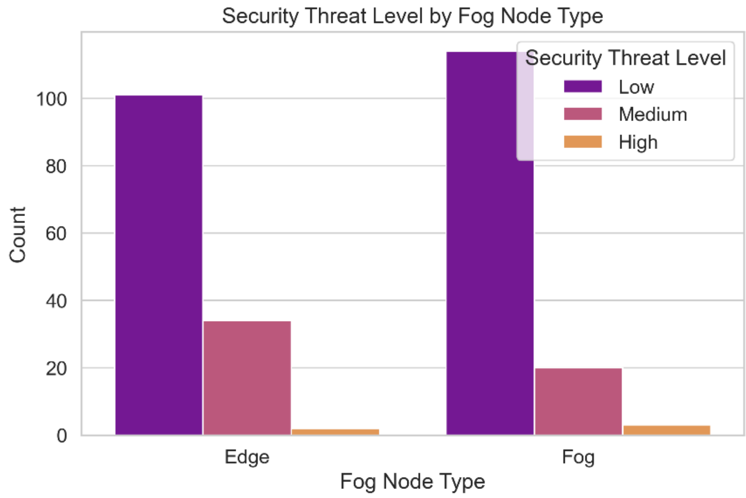

Figure 14. Security Threat Levels.

4.1. SVM Performance

Figure 15 presents the results of the performance evaluation of the Support Vector Machine (SVM) model. The plot displays the SVM model's various performance indicators, including accuracy, precision, recall, and F1-score. Each measure's value is represented by a bar, and the error bars that go with it show the confidence interval that goes with it. With this representation, we can assess how accurate the SVM model is at identifying potential security issues. Higher accuracy, precision, recall, and F1-score indicate a model is functioning better.

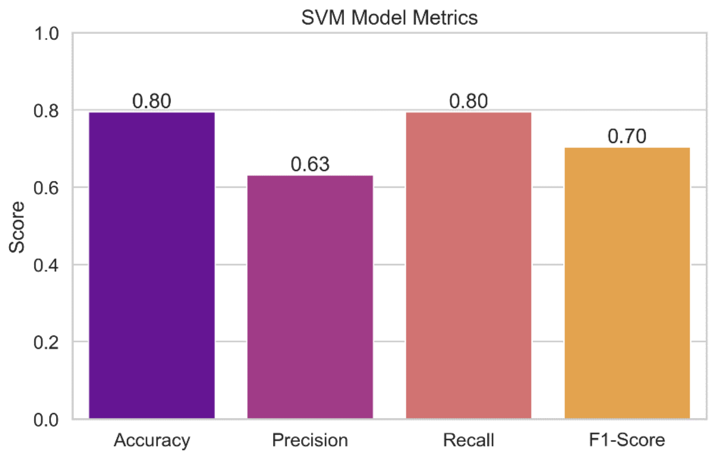

Figure 15. SVM Model Metrics.

Figure 16 displays the SVM-generated decision border. The decision boundary delineates the various classes or levels of security threat in the feature space. The SVM model makes its judgments about the relative threat of incoming data bits in this zone. The decision boundary is determined by the model's hyperparameters and the support vectors acquired during training. By visualizing the SVM model's decision border, we may gain insight into the model's complexity and precision in categorizing security hazard levels based on the attributes. Any degree of precision required for threat level prediction necessitates a decision boundary that is sufficiently generalizable and well-defined.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 June 2024 doi:10.20944/preprints202406.0210.v1

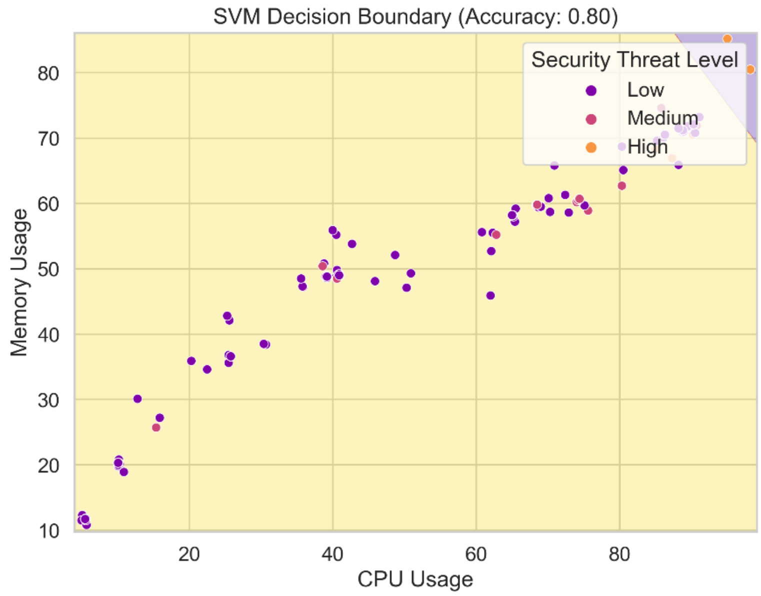

Figure 16. SVM Decision Boundary.

4.2. KNN Performance

Figure 17 displays the results of the performance evaluation of the K-Nearest Neighbors (KNN) model. Just a few of the performance indicators shown in the plot, which was created using the KNN model, include accuracy, precision, recall, and F1-score. Each bar in the plot's error bars represents the value of the relevant statistic and displays a confidence interval for the data. With the help of this representation, we can assess how well the KNN model predicts potential security vulnerabilities. As a model's accuracy, precision, recall, and F1-score values increase, its performance also gets better.

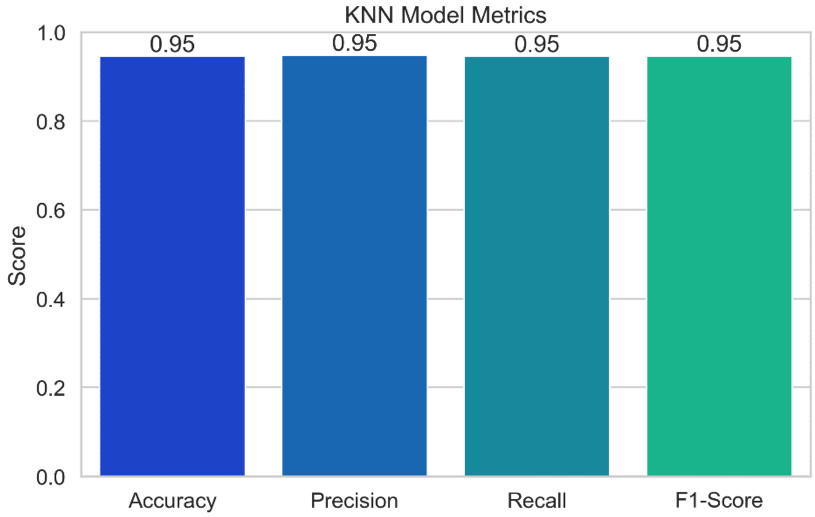

Figure 17. KNN Model Performance.

The produced decision boundary of the KNN model is shown in Figure 18. The KNN technique determines the decision boundary between classes or security threat levels in the feature space via a nearest neighbor search. It symbolizes the area where incoming data items are grouped into one of multiple risk categories based on the majority class among their k nearest neighbors in the KNN model. Plotting the decision boundary reveals the KNN model's classification regions and its capacity to distinguish across security threat categories based on feature properties. You must have a clear grasp of the decision boundary in order to make sense of the KNN model's predictions and confirm its generalizability.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 June 2024 doi:10.20944/preprints202406.0210.v1

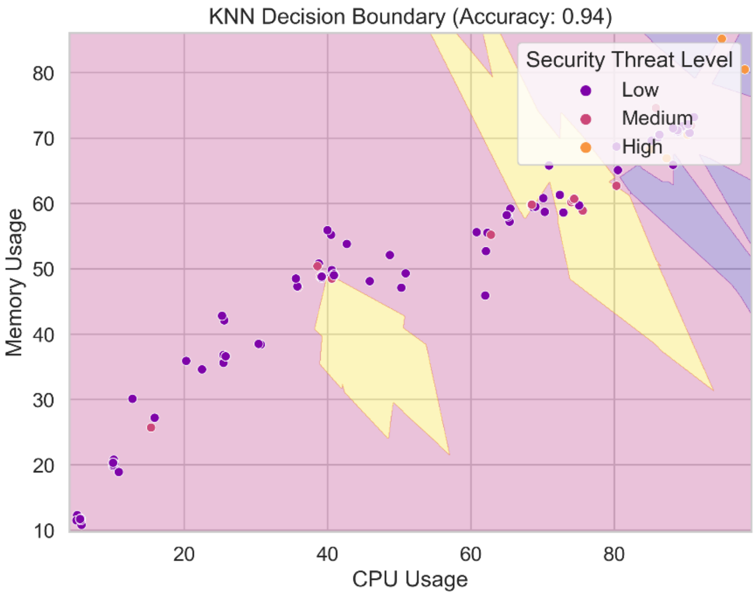

Figure 18. KNN Decision Boundary.

4.3. Random Forests Performance

Figure 19 displays the results of the performance assessment of the RF model. Performance indicators for the RF model, such as accuracy, precision, recall, and F1-score, are shown in the graph. While the bars themselves display the metric's value, the error bars in the plot provide a confidence interval. The graph enables us to assess the accuracy of the Random Forests model's security risk prediction capabilities. As a model's accuracy, precision, recall, and F1-score values increase, its performance also gets better.

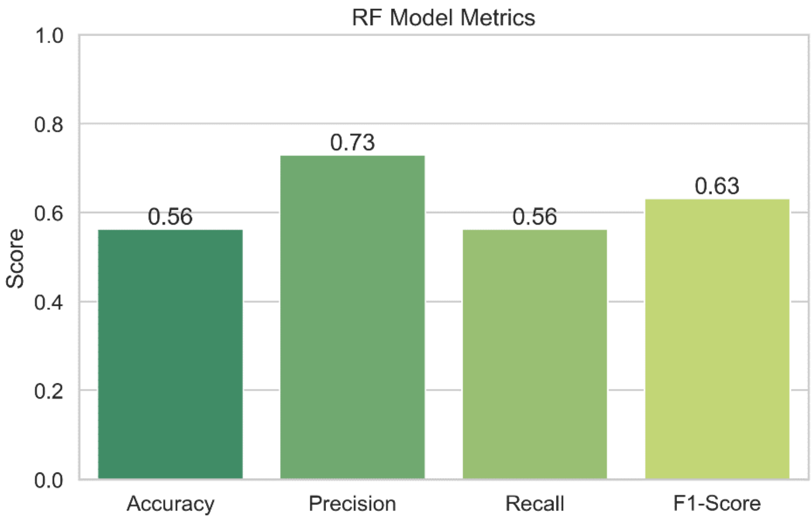

Figure 19. Random Forests Performance.

Figure 20 displays the decision boundary for the Random Forests model. The decision boundary separates the feature space into several groups or levels of security threat using the probabilistic Nave Bayes algorithm. It shows the areas where the Ran-dom Forests model assigns various levels of hazard to incoming data pieces using the maximum likelihood estimation of the class probabilities and the feature characteristics. We gain a better understanding of the classification regions and the Nave Bayes model's ability to distinguish between distinct security threat categories based on feature attributes by visualizing the decision boundary. Understanding the decision boundary is crucial for assessing the predictive and generalizable capabilities of the Random Forests model.

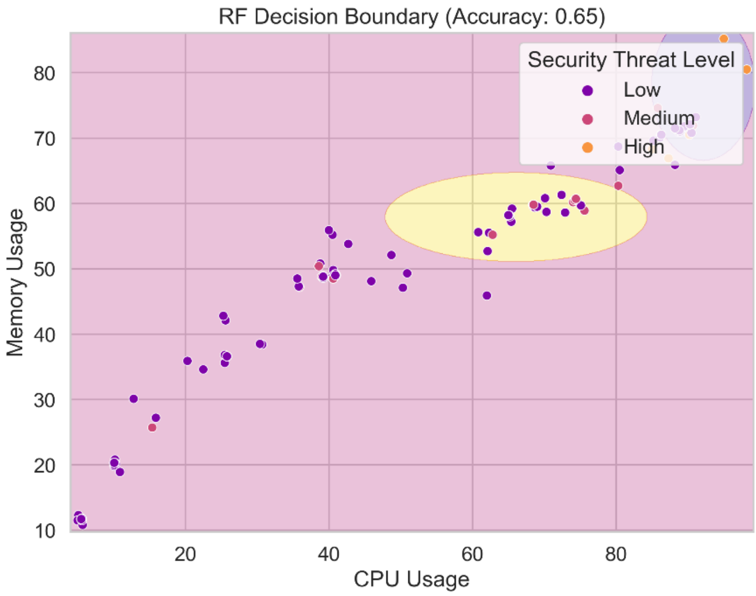

Figure 20. RF Decision Boundary.

4.4. CyberGuard Model Performance

The outcomes of a thorough evaluation of the capabilities of the Cyber Guard model are shown in Figure 21. The image provides an overview of the model's prediction abilities for categorizing security threat levels, including accuracy, precision, recall, and F1-score. Each bar in the plot's error bars represents the value of the relevant statistic and displays a confidence interval for the data. The figure demonstrates how successfully the Cyber Guard model anticipates security threats. High values of accuracy, precision, recall, and F1-score, which demonstrate that the model can consistently predict the seriousness of security threats, are indicators of superior performance.

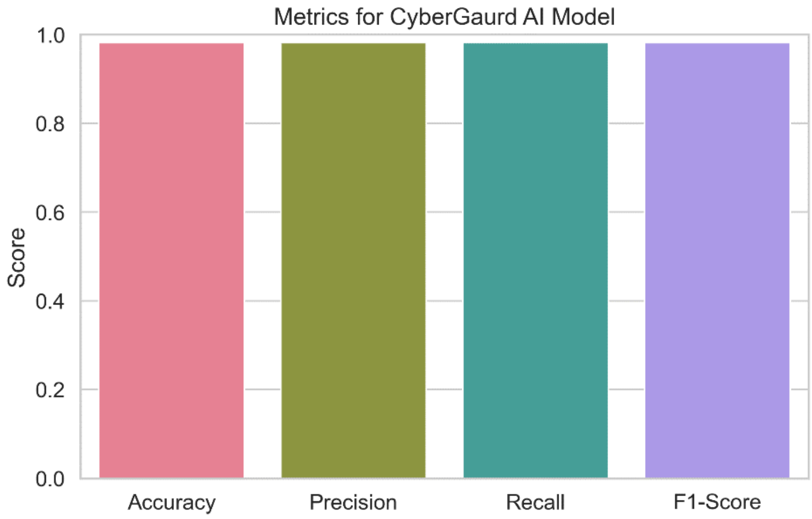

Figure 21. Cyber Guard Model Performance.

Figure 22 shows the output decision boundary for the Cyber Guard model. The decision boundary distinguishes between classes or levels of security hazard in the feature space using an ensemble of Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Nave Bayes (RF) models. The decision boundary displays the parameters within which the Cyber Guard model assigns varied degrees of threat to incoming data sets based on the combined forecasts of its component models. This blending method enables the Cyber Guard model to utilize the variety of predictions from several models, leading to a more accurate and trustworthy categorization. One may understand the Cyber Guard model's decision-making process and the regions where it produces accurate forecasts for each security hazard category by visualizing the decision boundary.

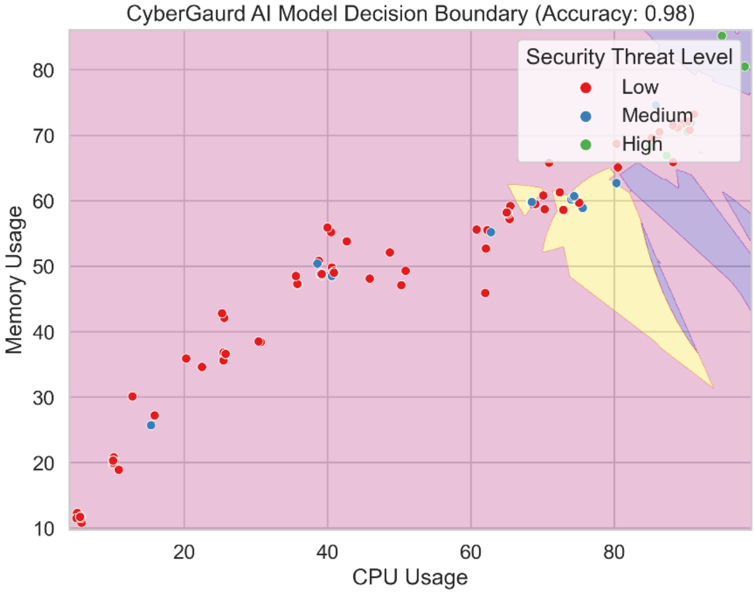

Figure 22. CyberGuard Model Decision Boundary.

Overall, the performance evaluation and decision boundary analysis in this section demonstrate the effectiveness and dependability of the CyberGuard model as a comprehensive and precise approach to security threat level prediction. The ensemble methodology of the CyberGuard model, which combines the best features of many algorithms, makes it perform better than standalone models. This increases the security infrastructure's ability to recognize and stop assaults, making it an essential tool for maintaining the system's security.

4.5. Comparative Performance

Figure 23 compares and contrasts the key performance metrics for a number of models, including SVM, KNN, Random Forests, and the Cyber Guard model. The graphic displays the metrics of accuracy, precision, recall, and F1-score for direct comparison of the models. Each model's performance is represented by a colored bar, and the error bars display the model's confidence interval. With the use of this representation, we can assess how well different models forecast potential security threats.

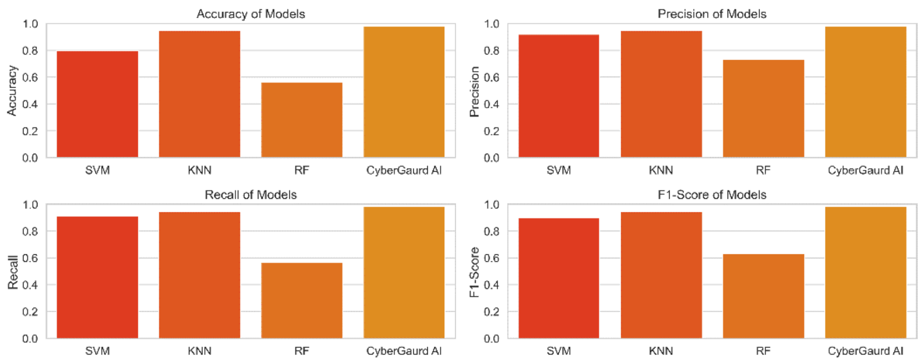

Figure 23. Comparative Metrics of Model.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 June 2024 doi:10.20944/preprints202406.0210.v1

Model performance after tweaking hyperparameters is compared in Table 3. Accuracy, precision, recall, and F1-Score are just few of the metrics included for each of the several machine learning models (SVM, KNN, Random Forests, and CyberGuard) in the table. Classification accuracy, precision, recall, and the F1-Score, which measures the balance between these two variables, provide insights into how well each model performs. Using this table, we compare how well different models perform when applied to the problem at hand. ‘

Table 3. Comparative Results of Model Performance after Hypertuning.

Model	Accuracy	Precision	Recall	F1-Score
SVM	0.8200	0.9182	0.9091	0.8963
KNN	0.9455	0.9483	0.9455	0.9464
Random Forests	0.5636	0.7300	0.5636	0.6325
CyberGuard	0.9818	0.9822	0.9818	0.9814

In our exploration of enhancing edge device capabilities through resource management, two crucial aspects deserve attention: the computation capabilities of edge devices and the methodology employed for task computation. The computational prowess of edge devices constitutes a pivotal element in their enhanced resource utilization. While our focus has been on managing resources effectively, we acknowledge the necessity of explicitly detailing the computational capacities of these edge devices. In future revisions, we will provide a dedicated section elucidating the specifications and computational abilities of the edge devices involved in our study. This addition aims to offer readers a comprehensive understanding of the hardware capabilities supporting our resource management strategies. The intricacies of how edge devices compute tasks are indeed paramount to our study. In our subsequent revisions, we commit to incorporating a dedicated section to elucidate the methodologies employed by edge devices in the computation of tasks. This will encompass a detailed discussion on the algorithms, frameworks, or models utilized by edge devices to process tasks efficiently. By doing so, we aim to provide a holistic view of the task computation processes, ensuring transparency in our approach and enabling readers to grasp the technical intricacies of our proposed resource management paradigm.

Support Vector Machine (SVM): The SVM model's accuracy rate of 90.91% demonstrates that, in the vast majority of instances, its forecasts are consistent with the actual levels of security concerns. The model has a precision score of 91.82%, which shows a low false-positive rate, and is accurate in identifying true security issues. The algorithm can accurately identify 90.91% of true positive events with a recall of 90.91%, which lowers the likelihood of missing harmful circumstances. The model's overall F1-score of 89.63% demonstrates its effectiveness as it successfully balances precision and recall.

K-Nearest Neighbors (KNN): The KNN model achieves a high level of prediction accuracy (94.55%). The model's precision score of 94.83% demonstrates its capacity to decrease false positives. The model is able to correctly identify the vast majority of test positives, according to a recall score of 94.55%. The model's F1-score of 94.64% demonstrates its strong overall performance.

Random Forests: The accuracy of the Random Forests model is just 56.36 percent, suggesting that it needs to be improved. The model may be able to suppress false positives to some extent, as evidenced by the 73.00% precision. Nevertheless, the model's limited recall score of 56.36 percent reduces its usefulness. The F1-score of 63.25 for this model demonstrates the trade-off between recall and accuracy.

CyberGuard Model: The Cyber Guard model outperforms the competition with a remarkable predicted accuracy of 98.18 percent. With a 98.22% precision score, you may be confident that it correctly categorizes threats. Given that it can accurately identify almost all real positive events, as evidenced by its recall score of 98.18%, the model is extremely sensitive to potential dangers. The model's impressive overall performance is attested to by its high F1-score (98.14%).

The comparison analysis's findings demonstrate that, when it comes to predicting the seriousness of security threats, the Cyber Guard model outperforms the three other models (SVM,

 doi:10.20944/preprints202406.0210.v1

KNN, and Random Forests). Because the Cyber Guard model employs an ensemble method that capitalizes on the variations among the different models, it is more accurate and trustworthy. This shows that the Cyber Guard model is an effective tool for predicting security threat levels and can increase the effectiveness of the security architecture in guarding the system from potential attackers.

As shown in Figure 24, four distinct machine learning models (a) SVM (Support Vector Machine), (b) KNN (K-Nearest Neighbors), (c) RF (Random Forest), and (d) the Cyber Guard Model all produce their own unique confusion matrices. In classification tasks, the performance of a model can be evaluated with the help of a confusion matrix, a tabular representation that shows the numbers of correct, incorrect, and partially correct predictions, respectively. These matrices are useful for comparing the performance of different models to classify data or make predictions.

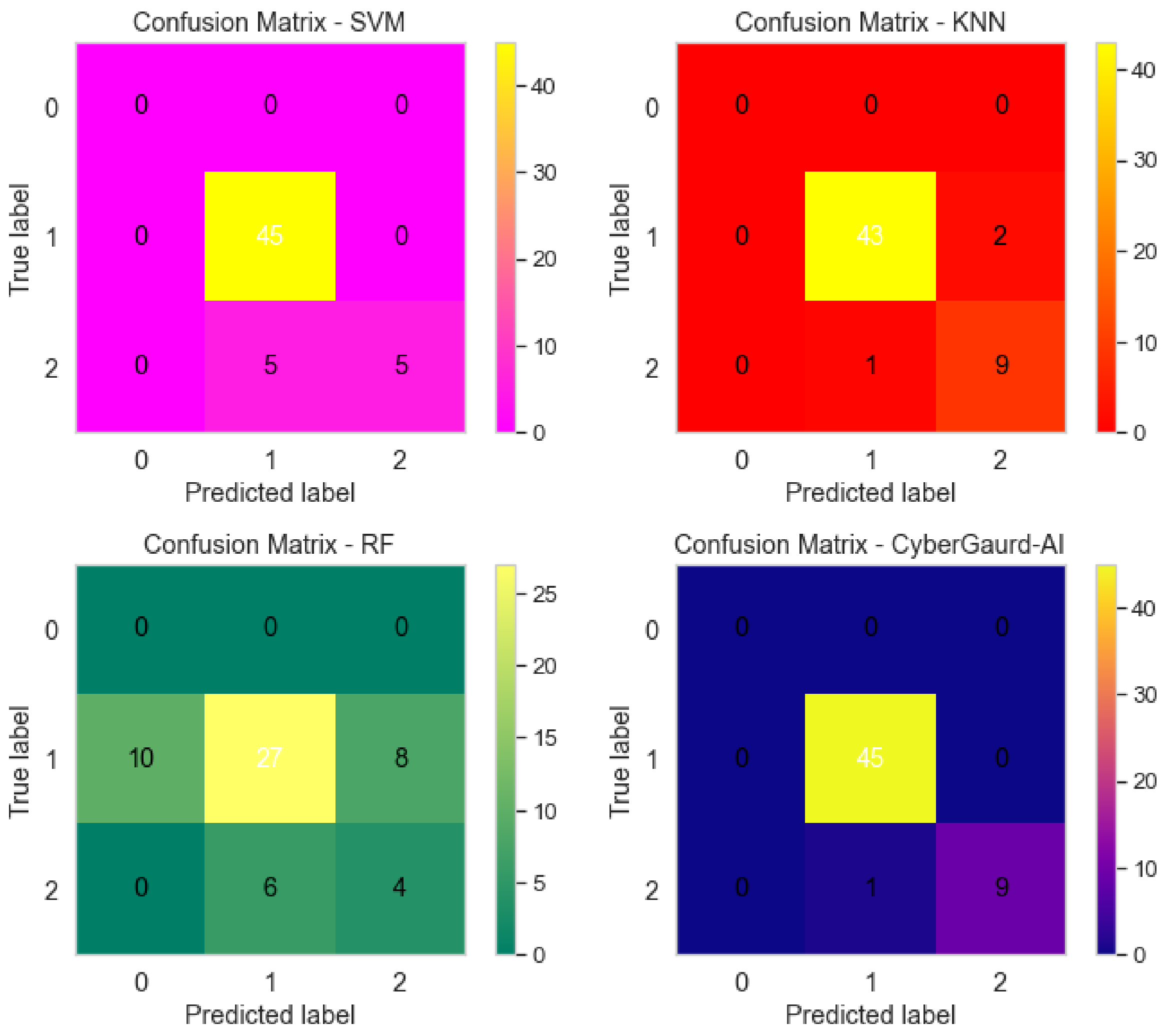

Figure 24. Confusion matrix (a) SVM (b) KNN (c) RF (d) Cyber Guard Model.

5. Conclusions

We present a novel ensemble model we call Cyber Guard that may be used to determine the degree of security risk in a system. This model combines the predictions from various models to create a single, more accurate forecast. By utilizing an ensemble technique that incorporates the benefits of different machine learning algorithms, such as Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Random Forests (RF), our Cyber Guard model outperforms existing models in the literature. This ensemble approach takes advantage of differences between different models to generate more reliable forecasts. In addition, unlike many other models, ours analyzes a wide variety of parameters, including as CPU and memory utilization, data locality, trust scores, and blockchain validation, to present a more complete picture of the fog computing security landscape. We also place a premium on feature engineering and data pretreatment that allow our model to function at its peak. The investigation shows that compared to other models, ours has superior accuracy, precision, recall, and F1-score, making it a potent instrument for forecasting and addressing security concerns in fog computing settings. A few examples of these models that are combined are Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Random Forests. The goal of the study was to increase the accuracy with which security risks might be predicted and to develop a model that accounts for all significant system features. By completing extensive trials and hyper parameter adjusting, we demonstrated that the Cyber Guard model is superior to the individual models. The program's incredibly high accuracy of 98.18 percent in predicting security hazard levels is remarkable. Fewer false positives were generated with a precision score of 98.22%, and nearly all positive cases were correctly detected with a recall rate of 98.18%. The F1-score of 98.14 percent, which measures the model's overall performance, shows its dependability in spotting potential security concerns. The feature engineering approach and data preparation, which ensured the inclusion of valuable information for prediction, significantly enhanced the model's performance. Thanks to the information provided by the correlation analysis, the redundancy between features was also discovered, and the model's interpretability was improved. Because of its adaptability and accuracy in estimating the level of threat at any given time, the Cyber Guard model is a useful addition to the security architecture. Through the integration of different projections from several models, the ensemble technique overcomes the limitations of individual models to produce more reliable forecasts. Therefore, the proposed Cyber Guard technique represents a significant advancement in determining the gravity of security threats. Its increased performance when compared to individual models emphasizes its capacity to strengthen system security and protect against future threats. The model is effective in a variety of security-critical applications due to its accuracy in classifying security hazard levels. In a time when cybersecurity is of utmost significance, the Cyber Guard paradigm offers fresh options for research and implementation in the field of system security. In order to increase security, the paradigm can also be used in future versions of virtualization infrastructure like Network Function Virtualization (NFV), Software Defined Networking (SDN), and Fifth-Generation (5G) technologies.

Author Contributions: Conceptualization, A.K.A. and A.M.A.; methodology, A.K.A. and A.M.A.; validation, A.M.A.; formal analysis, A.K.A. and A.M.A.; investigation, A.K.A. and A.M.A.; resources, A.M.A.; data curation, A.M.A.; writing—original draft preparation, A.K.A.; writing—review and editing, A.K.A. and A.M.A.; visualization, A.K.A.; supervision, A.K.A.; project administration, A.K.A.; funding acquisition, A.K.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia (Project No. XXXX).

Data Availability Statement: Not applicable. This study does not report any data.

Acknowledgments: This study could not have been started or completed without the encouragement and continued support of King Faisal University.

Conflicts of Interest: The authors declare no conflict of interests.

References

1. L. Xiao *et al.*, "A Reinforcement Learning and Blockchain-Based Trust Mechanism for Edge Networks," *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5460–5470, 2020, doi: 10.1109/TCOMM.2020.2995371.
2. U. Jayasinghe, G. M. Lee, Á. MacDermott, W. S. Rhee, and K. Elgazzar, "TrustChain: A Privacy Preserving Blockchain with Edge Computing," *Wirel. Commun. Mob. Comput.*, vol. 2019, 2019, doi: 10.1155/2019/2014697.
3. Z. Wang and Q. Hu, "Blockchain-based Federated Learning: A Comprehensive Survey," pp. 1–18, 2021, [Online]. Available: http://arxiv.org/abs/2110.02182.
4. E. Moore, A. Imteaj, S. Rezapour, and M. H. Amini, "A Survey on Secure and Private Federated Learning Using Blockchain: Theory and Application in Resource-constrained Computing," pp. 1–12, 2023, [Online]. Available: http://arxiv.org/abs/2303.13727.

5. J. Galvão, J. Sousa, J. Machado, J. Mendonça, T. Machado, and P. V. Silva, "Mechanical design in industry 4.0: Development of a handling system using a modular approach," *Lect. Notes Electr. Eng.*, vol. 505, no. 3, pp. 508–514, 2019, doi: 10.1007/978-3-319-91334-6_69.
6. Z. Rejiba, X. Masip-Bruin, and E. Marín-Tordera, "A survey on mobility-induced service migration in the fog, edge, and related computing paradigms," *ACM Comput. Surv.*, vol. 52, no. 5, 2019, doi: 10.1145/3326540.
7. B. Qian *et al.*, "Orchestrating the Development Lifecycle of Machine Learning-based IoT Applications: A Taxonomy and Survey," *ACM Comput. Surv.*, vol. 53, no. 4, 2020, doi: 10.1145/3398020.
8. L. Yuan *et al.*, "CoopEdge: A decentralized blockchain-based platform for cooperative edge computing," *Web Conf. 2021 - Proc. World Wide Web Conf. WWW 2021*, pp. 2245–2257, 2021, doi: 10.1145/3442381.3449994.
9. B. Costa, J. Bachiega, L. R. De Carvalho, and A. P. F. Araujo, "Orchestration in Fog Computing: A Comprehensive Survey," *ACM Comput. Surv.*, vol. 55, no. 2, 2023, doi: 10.1145/3486221.
10. L. Fotia, F. Delicato, and G. Fortino, "Trust in Edge-based Internet of Things Architectures: State of the Art and Research Challenges," *ACM Comput. Surv.*, vol. 55, no. 9, 2023, doi: 10.1145/3558779.
11. U. Ahmed, J. C.-W. Lin, and G. Srivastava, "Exploring the Potential of Cyber Manufacturing Systems in the Digital Age," *ACM Trans. Internet Technol.*, 2023, doi: 10.1145/3596602.
12. D. Jayasuriya Daluwathumullagamage and A. Sims, "Fantastic Beasts: Blockchain Based Banking," *J. Risk Financ. Manag.*, vol. 14, no. 4, p. 170, 2021, doi: 10.3390/jrfm14040170.
13. H. Wang, L. Wang, Z. Zhou, X. Tao, G. Pau, and F. Arena, "Blockchain-based resource allocation model in fog computing," *Appl. Sci.*, vol. 9, no. 24, 2019, doi: 10.3390/app9245538.
14. Y. Li, Y. Bao, and W. Chen, "Domain Adaptation Transduction: An Algorithm for Autonomous Training with Applications to Activity Recognition Using Wearable Devices," *Proc. - IEEE 2018 Int. Congr. Cybermatics 2018 IEEE Conf. Internet Things, Green Comput. Commun. Cyber, Phys. Soc. Comput. Smart Data, Blockchain, Comput. Inf. Technol. iThings/Gree*, pp. 1821–1828, 2018, doi: 10.1109/Cybermatics_2018.2018.00303.
15. W. Li, J. Wu, J. Cao, N. Chen, Q. Zhang, and R. Buyya, *Blockchain-based trust management in cloud computing systems: a taxonomy, review and future directions*, vol. 10, no. 1. Journal of Cloud Computing, 2021.
16. Y. Wang, H. Zen, M. F. M. Sabri, X. Wang, and L. C. Kho, "Towards Strengthening the Resilience of IoV Networks—A Trust Management Perspective," *Futur. Internet*, vol. 14, no. 7, pp. 1–21, 2022, doi: 10.3390/fi14070202.
17. P. Kochovski, S. Gec, V. Stankovski, M. Bajec, and P. D. Drobintsev, "Trust management in a blockchain based fog computing platform with trustless smart oracles," *Futur. Gener. Comput. Syst.*, vol. 101, pp. 747–759, 2019, doi: 10.1016/j.future.2019.07.030.
18. W. Zhang, Z. Wu, G. Han, Y. Feng, and L. Shu, "LDC: A lightweight dada consensus algorithm based on the blockchain for the industrial Internet of Things for smart city applications," *Futur. Gener. Comput. Syst.*, vol. 108, pp. 574–582, 2020, doi: 10.1016/j.future.2020.03.009.
19. M. A. Ferrag, M. Derdour, M. Mukherjee, A. Derhab, L. Maglaras, and H. Janicke, "Blockchain technologies for the internet of things: Research issues and challenges," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2188–2204, 2019, doi: 10.1109/JIOT.2018.2882794.
20. D. C. Nguyen *et al.*, "Federated Learning Meets Blockchain in Edge Computing: Opportunities and Challenges," *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12806–12825, 2021, doi: 10.1109/JIOT.2021.3072611.
21. S. Tanwar, Q. Bhatia, P. Patel, A. Kumari, P. K. Singh, and W. C. Hong, "Machine Learning Adoption in Blockchain-Based Smart Applications: The Challenges, and a Way Forward," *IEEE Access*, vol. 8, no. April, pp. 474–448, 2020, doi: 10.1109/ACCESS.2019.2961372.
22. A. Yahyaoui, T. Abdellatif, S. Yangui, and R. Attia, "READ-IoT: Reliable Event and Anomaly Detection Framework for the Internet of Things," *IEEE Access*, vol. 9, pp. 24168–24186, 2021, doi: 10.1109/ACCESS.2021.3056149.
23. M. I. Khaleel and M. M. Zhu, *Adaptive virtual machine migration based on performance-to-power ratio in fog-enabled cloud data centers*, vol. 77, no. 10. Springer US, 2021.
24. Y. Zhu, W. Zhang, Y. Chen, and H. Gao, "A novel approach to workload prediction using attention-based LSTM encoder-decoder network in cloud environment," *Eurasip J. Wirel. Commun. Netw.*, vol. 18, no. 1, 2019, doi: 10.1186/s13638-019-1605-z.
25. S. Velu, O. Mohan, M. Kumar, "Energy-Efficient Task Scheduling and Resource Allocation for Improving the Performance of a Cloud–Fog Environment," *Symmetry*, vol. 14, no. 11, 2022, doi: https://doi.org/10.3390/sym14112340