Acta Cybernetica **26** (2023) 149–173.

EA-POT: An Explainable AI Assisted Blockchain Framework for Honeypot IP Predictions*

Shajulin Benedict[a]

Abstract

The culpable cybersecurity practices that threaten leading organizations are logically prone to establishing countermeasures, including honeypots, and bestowing research innovations in various dimensions, such as ML-enabled threat predictions. This article proposes an explainable AI-assisted permissioned blockchain framework named `EA-POT` for predicting potential defaulters' IP addresses. `EA-POT` registers the probable defaulters predicted by explainable AI based on the approval of IP authorizers of blockchain databases. Experiments were carried out at the IoT Cloud Research laboratory using three prediction models, such as Random Forest Modeling (RFM), Linear Regression Modeling (LRM), and Support Vector Machines (SVM); and, the experimental results for predicting the AWS honeypots were explored. The proposed `EA-POT` framework revealed the procedure for including interpretable knowledge while blacklisting IPs that reach honeypots.

Keywords: blockchain, cyber security, honeypot, hyperledger, predictions

1 Introduction

Developing a secure cloud-based or IoT-enabled application is an extraordinary feat of development as newer security issues evolve, especially when the post COVID-19 scenario was considered in a connected devices world. Remote accesses to organizational resources and services are prone to security challenges in newer dimensions. Notably, as an essential part of preparedness, transferring identity credentials to employees has become a landmark shift in handling the security challenge needed to protect resources.

It is estimated by high-income companies/organizations and researchers that a reasonably high volume of budget needs to be spent to counteract evolving cybersecurity issues. For instance, Australia economists have estimated that it will spend over $7.6 billion by 2024 [32]; Investments towards cloud security tools are projected

*This work was supported by AIC IIITKottayam and BEL Consultancy Project.

[a]Indian Institute of Information Technology Kottayam, India, E-mail: shajulin@iiitkottayam.ac.in, ORCID: 0000-0002-2543-2710, www.sbenedictglobal.com

DOI: 10.14232/actacyb.293319

to increase from $5.6 billion in 2018 to $12.6 billion in 2024 [36]; `Centrify`, a company specializing in cybersecurity, highlighted the possibility of phishing attacks that could lead to a huge investment in potential IIoT industries [35].

Looking to the future, in the wake of COVID-19, many countries or organizations, especially those belonging to government sectors, have suggested newer security policies or procedures to counteract notable sprouting security challenges, such as i) phishing, ii) malicious attacks, iii) accessing orphaned accounts, iv) ransomware attacks, v) advanced persistent threat, and so forth. In fact, the *WannaCry* ransomware attack challenged over 150 countries [17]. Additionally, IoT devices, which increase day by day, require diligent secure connectivity services. The failure to provide proactive secured access to connected devices or associated cloud services, especially in the automobile and healthcare industries, could lead to unnecessary data leaks. This could disrupt important automated decisions and slow down the global economic situation. There have been a few research efforts in the recent past to address the IoT security inefficiencies [26, 27, 31].

Obviously, preventing potential attackers/hackers from breaching security needs to be handled diligently. In recent years, honeypots have been established by several leading cloud-based service providers, including AWS IoT infrastructure providers. Honeypot, in general, lies alongside the firewall inviting security challenges from potential hackers. In doing so, the specific pattern of attacks can be explored; the motivation of cybercriminals in writing code could be reduced; the activity of investing money for illegal purposes may be minimized; the intention of attackers and involved countries can be observed; and, the possibility of the attackers' evolving innovations can also be studied.

Traditionally, honeypots on cloud infrastructures address several known issues as listed below:

1. The attacks initiated by their own organizations' employees must be diligently handled. In fact, such organizational attacks are possible due to poor knowledge of utilizing cyber-physical devices/gadgets or the associated services;

2. Indigent policies of honeypots need to be dynamically handled in a decentralized environment ; and,

3. The time needed to learn about the potential attack has to be negligible compared to the time it takes to attack.

This paper proposes an `EA-POT` framework, an Explainable AI-assisted blockchain framework, for honeypot IP predictions. `EA-POT` attempts to reduce the time needed to identify potential attackers using prediction algorithms, such as Random Forest Modeling (RFM), Linear Regression Modeling (LRM), and Support Vector Machines (SVM). Unlike traditional methods, which are dependent on non-explainable parameters (black-box and temperamental), the proposed `EA-POT` framework enables the explainability features of prediction models.

The framework is combined with a hyperledger fabric-based blockchain network to register the honeypot IP addresses and to inject dynamic prevention policies on

the fly. The reasons for including the permissioned blockchain into the framework are multi-fold:

1. The blacklisted IP addresses considered to be more vulnerable become immutable as the organizations or the inner employees of an organization cannot modify them; and,
2. Specific policies can be formulated by involving permissioned organizations or stakeholders in deciding the actions against the defaulters.

In addition, experts believe that the performance of the hyperledger fabric, especially when the chaincodes are written using golang, is reasonably better than the other blockchains. Authors of [9] have studied the performance impact of transactions concerning the underlying programming languages; similarly, authors of [22] have delved into the end-to-end transaction latency factors of hyperledger fabric blockchains.

In this paper, the research work emphasized the importance of the `EA-POT` framework to register blacklisted IPs, which were explainable using prediction models, in the immutable database. Experiments were held at the IoT cloud research laboratory on distributed systems after a Kubernetes cluster of hyperledger fabric components was launched.

The major contributions of the work are listed as follows:

1. an `EA-POT` framework was proposed to register potential hackers into the blockchain database after the policies were satisfied;
2. the importance of explainable AI while predicting IPs was explored; and,
3. the experimental results were investigated and revealed to highlight the necessity of the proposed `EA-POT` framework.

The rest of the paper is organized as follows: Section 2 investigates the state-of-the-art research in the field of honeypots and the utilization of explainable AI for enhancing cybersecurity; Section 3 reveals the functionalities and components of the proposed `EA-POT` framework; Section 4 illustrates the approach of utilizing explainable AI for the framework; Section 6 manifests the experimental evaluations of the proposed framework that were carried out at the laboratory; and finally, Section 7 offers a few outlooks and conclusions for the near future research based on the proposed work.

2 Related Work

Countering cybercrime in several countries is often considered an ongoing crucial agenda. In fact, a proactive approach to handling security measures has attracted several researchers/countries in recent years. Honeypots, being a measure of luring potential hackers, have served as a foundation for proactively analyzing the characteristics of hackers and their malicious behaviors.

This section explains the state-of-the-art work of honeypot research in three different perspectives as listed below:

1. Honeypot placements (Clouds),
2. Inclusion of Machine Learning / Explainable AI,
3. Application of Blockchains.

Finally, the shortcomings of the existing works and the contributions of this article are expressed in the section.

2.1 Honeypot Research – Domains, Placements, and Clouds

Researchers/practitioners belonging to several domains, such as Clouds and Industrial IoT (IIoT), have evidenced the importance of including honeypots in their organizations. There exist several honey pot implementations, both static and dynamic, in IIoT or cloud environments. For instance, authors of [20] and [16] studied the application of honeypots for smart grids; authors of [7] revealed the importance of honeypots for capturing DDoS attacks in IIoT environments; and, authors of [24] proposed a social leopard algorithm to detect ransomware attacks using honeypots. Additionally, a few honey pot implementations for protecting buildings [3] and establishing a secured smart home infrastructure [14] have been developed in the recent past.

A sector of researchers has attempted to optimize honeypot placements in organizations based on malicious attackers – i.e., authors of [12] and [1] have applied a game-theoretic framework model to optimally choose honeypots in various locations; in [10], the authors have installed honeypots in nine countries and studied the behavior of malicious users. Besides, honeypots have been widely deployed to enable lightweight interactions in IoT-based infrastructures. For instance, authors of [26] have implemented *BoTNet*, and authors of [27] have implemented *IoTCMal* for low interaction honeypots using TelNet and SSH; authors of [6] have deployed a global honeypot infrastructure to detect industrial attacks.

The deployment of honeypots has been studied in cloud environments as potential attacks on public cloud infrastructures, such as AWS Cloud, Google Compute Engine, and Microsoft Azure. This process has become an inevitable activity. Accordingly, a few researchers have oriented their analysis and studies towards cloud infrastructures. For instance, the authors of [21] have developed a high interaction system using Kerberos authentication, Virtual Private Cloud, and Elastic File System to understand the malicious nature of attackers; the authors of [13] have developed honeypot as a service model for luring attackers. This honeypot-as-a-service is implemented as a plug-and-play model, which could be hosted on gateways for capturing the malicious attackers; and, a few practitioners have listed the AWS honeypot data that suggested potential hackers, who attempted to maliciously attack AWS cloud services, including AWS IoT services.

2.2 Machine Learning and Explainable AI

Traditionally, machine learning has been applied in several domains, including IoT to predict machine behavior or future events [8]. Several variants of machine learning, including federated learning aspects, have reached the market for efficient learning processes of IoT and cloud services [28, 4]. Additionally, the application of own decision-making algorithms, such as neural networks, has been practiced to classify security attacks [2].

In fact, proactively learning the behavior of malicious attackers or potential IP addresses of the attacker's needs a diligent skillset that modern computational efforts are required. With the evolution of several machine learning platforms and tools, in recent years, the identification of attackers and the classification of the severity of attacks have become a widely discussed topic of research. Authors of [23] have studied the application of probabilistic models for proactively estimating the honeypot detection. The authors have confined their research to TELNET and SSH-based communications in IoT domains. Authors of [15] have applied an outlier detection mechanism to project anomalies from the outlier information. To do so, the authors have utilized unsupervised machine learning approaches for honeypots.

A few machine learning researchers have predicted attacks using statistical modeling methods, including GARCH models. For instance, the authors of [19] have characterized the honeypot captured data using statistical approaches. The authors have pointed out the importance of explainable statistical approaches for efficiently handling the prediction problems in honeypot data using case studies. The same authors have additionally predicted cyber-attack rates using GARCH prediction models in their following works [30]. Obviously, robust prediction models are crucial for proactively identifying the potential hackers or malicious attackers in modern networking applications, including IIoT or cloud services.

Apart from the normal prediction approaches which predict the potential hackers or their activities, a few researchers have devised honeypot mechanisms to protect against vulnerabilities arising out of the adversarial learning processes. Authors of [29] have suggested learning models that protect against adversarial errors opted by automated machine learning algorithms. For instance, IIoT applications, guided by machine learning services, could be exposed to wrong learning advice which could end up with hazardous results. To override such effects, honeypots were utilized to protect failures and rectify prediction failures.

As observed, there exists a few research works that utilize machine learning algorithms and mechanisms, including the Cloud services domain, for predicting or characterizing hackers. However, there are very few works that utilize explainable AI for validating the importance of honeypot predictions levied by machines or computing domains in an organization. It could be noticed from the literature that explainable AI has emerged in the recent past to justify the blackbox prediction approaches or prediction algorithms.

2.3 Blockchains for Honeypots

Although attackers and associated vulnerabilities could be predicted, the findings need to be protected. Insider attacks in most organizations have been highly dangerous due to the modifications and corrections held to the findings by potential inner-organizational hackers. Blockchains could protect against tampering with data in such environments. In addition, the security policies could vary depending on regional/organizational policies. For instance, blacklisting certain IP addresses depends on several factors, including the organizational relationships with associated countries.

There exists a few research works of honeypots relating to blockchains. However, they were applied in a different context. For instance, researchers of [25] and [5] have established data science algorithms to learn the potential fraudulent activities such as fraud payments due to the Ethereum smart contracts.

A very few research works have applied the permissioned blockchains to quickly validate the blacklisting IPs that reach honeypots.

This work endeavored to apply prediction models, such as RFM, LRM, and SVM to predict the potential hacking IPs and commit the information into the permissioned blockchain ledger. The entry of information into the ledger is governed by a few approvers, including the Explainer-AI of the proposed framework (see Section 3).

3 `EA-POT` Framework

Honeypots have been reasonably deployed alongside production systems in recent years to study the behavior of potential cyberattackers. Accordingly, the honeypots pave way for the security team of organizations to protect their systems from several vulnerable attacks. In fact, the potential attackers should be predicted in an explainable manner before the information were listed in an immutable database.

This section explains the inner details of the proposed `EA-POT` framework for blacklisting potential cyber attackers using explainable prediction models and blockchains.

The proposed `EA-POT` framework consists of the following entities:

- Honeypot Data Engine,
- Prediction Models,
- Explainable AI Components,
- Policy Stakeholders,
- Blokchain Network, and,
- `BlackBlock` Database.

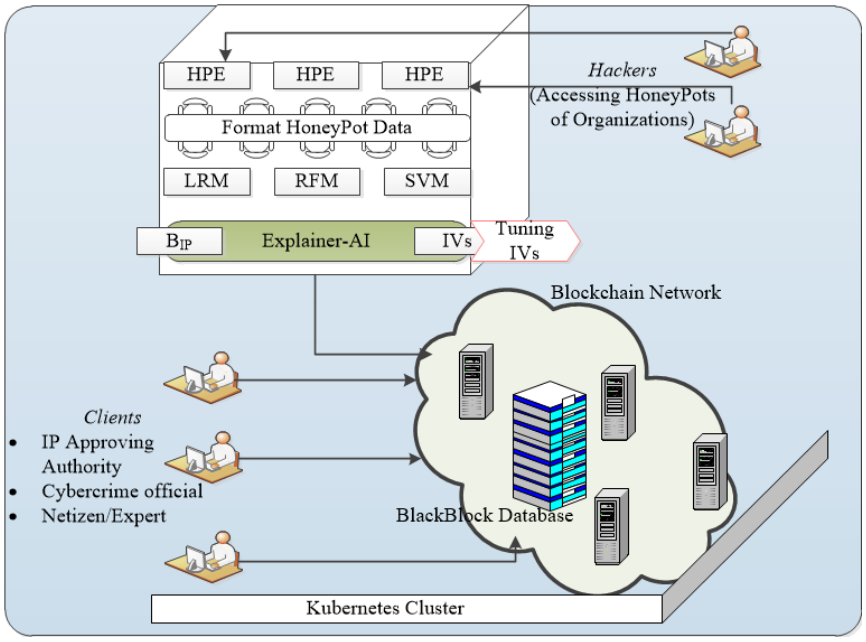


Figure 1: `EA-POT` Framework

The crucial functionalities of these entities are explained below.

3.1 Honeypot Data Engine

The *Honeypot Data Engine* is an entity that resides on honeypots that are located nearer to the firewall component of the organizations. It collects information, such as IP addresses, source port address, destination port addresses, connection protocols, such as TCP or UDP, country of origin, and so forth, of defaulters. Besides, it formats the information into CSV, XML, and JSON formats, in a periodic manner and keeps them ready for further processing of the intended prediction models of the framework.

3.2 Prediction Models

The framework utilizes a few notable algorithms, such as RFM, LRM, and SVM for predicting the potential hackers and their IP addresses. One battle in which the traditional honeypot engines allied to defeat progress was the timely identification of potential hackers' IP addresses. In doing so, several countermeasures could be adopted for overriding the issues.

The synopsis of the three prediction algorithms applied in the `EA-POT` framework is given in the following paragraphs.

Random Forest Modeling (RFM) Random Forest Modeling (RFM), the concept initially conceptualized by Breiman et al. [40], has been widely applied for creating prediction models that resemble real-world situations. It is an ensemble-based learning approach that creates decision forests based on modeling features. The models are created for the dependent variable of the dataset. For instance, the independent variable for honeypot IP prediction includes the IP addresses of potential hackers.

The decision forests consist of tens of hundreds of decision trees that analogously represent rules and inferences. Based on the creation of decision forests considering the decision rules for training data, the predictions are applied to the testing data. During the process of predictions, in the case of honeypot IP predictions, the independent variables, such as source port addresses, destination port addresses, latitude and longitude of locations, and so forth, are considered as modeling features – i.e., the independent variables.

RFM-specific tuning parameters[41], such as number of trees to grow in a forest (`ntree`), number of trials (`mtry`), and so forth, define the prediction accuracy on the testing dataset. For instance, increasing the number of trees could improve the prediction accuracy on large datasets.

Support Vector Machine (SVM) Support Vector Machine (SVM) is a supervised learning algorithm [11], as similar to RFM, where it attempts to produce hyperplanes that split data with sufficient distinctions. It attempts to increase the decision boundary of categorizing training data so that predictions could be much easier. The accuracy of the prediction algorithm is highly dependent on the dataset that is utilized – i.e., if the algorithm could not find sufficient hyperplanes, the error rate for predictions is typically higher than the expected ones.

During the training processes, the SVM algorithm iteratively prepares hyperplanes based on the independent variables of the dataset. To do so, it utilizes kernels, such as linear, polynomial, radial, and sigmoid, to transform the training data to a high dimensional space so that the process of creating hyperplanes is comparatively carried out elegantly.

Linear Regression Modeling (LRM) Linear Regression Modeling (LRM) is considered to be the simplest prediction model that identifies the relationship between the dependent and independent variables of a dataset. It highlights the potential changes that could happen in the dependent variable while modifying the independent variables. Not all independent variables are inclined towards the dependent variables of a dataset.

During the training processes of the linear regression algorithm, linear equations or mathematical formulas are created for the dependent variable based on the training dataset. In the proposed work, ML algorithms, such as RFM, SVM, and LRM are sufficient for learning the blacklisted IPs as decisions on confirming them are governed by a few stakeholders of blockchain networks. Accordingly, the policies could be varied as specified in the blockchains and the predictions are faster than

learning algorithms, such as neural networks.

3.3 Explainable AI Components

The prediction models are reasoned using the explainable AI components augmented in the framework. The framework applies explainable AI components to explore the inference levied from the models. For instance, the framework feeds specific modeling parameters to understand the R^2 values of the models. The R^2 values determine the closeness of the model and the dependent variables. The framework iterates over the available independent variables to identify the best set of independent variables $IV_{1...n}$ which offer the best R^2 values.

3.4 Policy Stakeholders

The policies for registering an IP to blacklists and for releasing the IPs from the blacklists need to be guided/formulated by multiple stakeholders. For instance, email hackers, the IP addresses, and port numbers of hacking applications need to be blacklisted depending on genuine reasons. Notably, blacklisting IP addresses due to technical failures reduces the reputation of an organization. Hence, in `EA-POT` framework, an array of policy stakeholders are represented for validating the genuineness of blacklisting IP addresses. In addition, it involves the explainable AI features to evaluate the necessity of blacklisting an IP into the immutable database.

3.5 Blockchain Network

The policy stakeholders of the `EA-POT` framework are connected to each other using a P2P blockchain network. These policy stakeholders are responsible for running policies or chaincodes; and, to interpret the data on server components. These server components, mostly established as a docker farm, are connected to each other using the blockchain network.

3.6 BlackBlock Database

The potential blacklisted IP addresses that are predicted and validated using the blockchain stakeholders of the network are registered into the blockchain ledger of `EA-POT` framework named as *BlackBlock* database. The reason to set up a blockchain database to register blacklisted IPs into the ledger is to protect the vulnerability raised by potential hackers, mostly the vulnerability due to the inner threats by colleagues of the same organizations. Figure 1 depicts on the entities involved in the `EA-POT` framework.

4 Explainable AI and Predictions

The recent era of machine learning development, in various research domains, has seen a proliferation of prediction models which can often be classified as blackbox

mechanisms. At this juncture, the evolution of explainable AI concepts has improved the trust levied by researchers on blackbox models. This section explains the interpretability procedure of prediction models of the `EA-POT` framework.

In general, a blacklisting of IP addresses happens due to several reasons:

1. an execution of a malicious program in a machine, including sensor nodes;

2. varying policies of organizations, which protect against the utility of certain types of applications – for instance, a military organization does not permit access to unauthorized military services;

3. inappropriate content, such as illicit videos and images in the services; and

4. spying of services within intra- and inter-organizations.

Predicting the blacklisting of an IP address in `EA-POT` framework attempts to avoid threats and strengthens the firewall policies depending on the learning inferences. In addition to a normal prediction process, `EA-POT` framework applies explainable features of AI to bolster the accuracy of predictions.

There may be various reasons for the formidable range of issues and inaccuracies of prediction models in modern applications:

1. the learning parameters are not appropriately chosen;

2. the modeling algorithms learn almost all available data – i.e., the model is biased concerning the data;

3. the training datasets are comparatively low; and so forth.

Obviously, it is an impressive activity for the user to understand the reason for predicting the blacklisting IP, an independent variable B_{IP}, with a specific level of accuracy considering dependent variables $X_{i...n}$. `EA-POT` utilizes local independent variable information of models for collecting $X_{i...n}$ that influence the predictions.

The major advantages of including the explainable features of the model in the `EA-POT` framework are:

- the features of *Explainer-AI* reveal the level of confidence of prediction models in R^2 percent; and,

- they establish a set of permutations from the observation instances and highlight the inclination of dependent variables towards the independent variable.

The *Explainer-AI* identifies the best suitable modeling parameters based on the R^2 values of the prediction models. Accordingly, the algorithms impose the choice for registering IP addresses into the blockchain ledger.

5 Immutability of `BlackBlock` and Processes

In `EA-POT` framework, *BlackBlock* database is established to list the blacklisted IP addresses that are predicted to be registered into the honeypots of organizations. In this section, the formation of a blockchain network, *Blackblock* database, and the processes involved in ensuring immutability are discussed.

5.1 Blockchain Network

The *BlackBlock* database of `EA-POT` framework is a distributed ledger that is established in the nodes of a Kubernetes cluster. The Kubernetes cluster [39] is chosen for the scalability and reliability features of distributed ledgers.

In general, Kubernetes is an orchestration tool that manages the containerized workloads of applications. It is manifested that the performance of Kubernetes clusters is better than many other orchestration tools while executing the containerized applications on them [18].

In `EA-POT` framework, the stakeholders of blockchains are represented as docker instances, which are containerized instances. The inclusion of the Kubernetes cluster enables users to evaluate and modify the state of docker machines, typically, the peer nodes of blockchains in the network.

The policy stakeholders of the framework that are represented in the docker instances include:

- IP Approving Authority,
- Explainer-AI,
- Cybercrime official, and
- Netizen/Expert.

These stakeholders have provisions to interact with the docker instances through docker client instances (see Figure 1). The docker instances, which represent the stakeholders of the `EA-POT` framework, install and launch chaincodes, the policies, for understanding the inferences of explainable AI, and for manipulating the entry of IP addresses into the ledger.

The chaincodes of the framework are written in `golang` language. These chaincodes are responsible for implementing policies of stakeholders where the Explainer-AI or similar stakeholders could determine the approval of transactions – i.e., the registering of blacklisted IPs into the database. The chaincodes are instantiated, installed, packaged, and queried using specific commands as shown below:

```
peer chaincode install/instantiate/...
```

The *Blackblock* database is protected within a specific channel that has connections to the permissioned stakeholders. The channel configurations and associated

information are defined before starting the blockchain network. The channel is responsible for establishing a sub-network where peer nodes could share the database within the organizations.

The blockchain network of the `EA-POT` framework emphatically complies with promoting a trustless trust environment using the distributed docker instances. The network offers ledger services across the connected nodes. It enables the nodes to readily keep the database for querying or modifying or manipulating the records in the database.

5.2 *BlackBlock* Database

The proposed `EA-POT` framework has a specific data structure to append blacklisted IP addresses into the *BlackBlock* database. The data in the database is appended as backlinked listed blocks for every initiation of transactions by peer nodes. The blocks are identified by hashes which include the previous hash values of the blockchain and the state of the blocks [37].

Each block is appended with a data structure that includes IP addresses, source port addresses, destination port addresses, and country information. Typically, the chaincode policies determine the entry of the blacklisted IP into the *BlackBlock* database.

The data appended into the database is sequential and immutable. The moment an entry would be registered into the ledger in a channel, the data will be visible to all available peer nodes of the channel.

5.3 Processes Involved

The processes involved in the entire life cycle of the `EA-POT` framework for registering the blacklisted IP addresses into the permissioned blockchain are described as steps below:

1. *Initiation:* In this step, the honeypot data engine and blockchain networks are initiated on top of the Kubernetes cluster. This means that the services are enabled at servers to attract potential hackers. In addition, the channel and peer networks are activated for implementing chaincode policies.

2. *Predictions:* Based on the available data, the learning models are created using sophisticated algorithms, such as RFM, LRM, and SVM. The generated regression models are utilized for predicting the future potential hacker IP addresses.

3. *Explanations:* Using the generated prediction models, explanations are developed using the independent variables of the models in the EA-BOT framework. The explanations are linked to the chaincode policies of the policy stakeholders of the blockchain network such that the stakeholders govern the control of blockchain transactions, including Explainer-AI.

4. *Chaincode Instantiation:* The stakeholders of the `EA-POT` framework are responsible for collectively agreeing upon entering the predicted IPs to blacklist them. Chaincode policies are defined in the `EA-POT` framework such that the stakeholders are diverse in nature – i.e., one stakeholder is *Explainer-AI*. This stakeholder evaluates the model that manifests a higher threshold of agreement while blacklisting IPs; another stakeholder is a representative of country authorities who approves and disapproves the blacklisted IPs – i.e., *IP Approving Authority*. This stakeholder evaluates the IPs concerning the country-wide policies set up for IPs; the other stakeholder is a netizen/expert who has a wide experience in executing a similar kind of applications and have the knowledge to judge the genuineness of actions to some confidence level. This stakeholder is named as *Cybercrime Official*, and the last stakeholder of the `EA-POT` framework is responsible for evaluating the IPs based on the genuineness of country-specific information.

5. *Transactions:* Once when the stakeholders agree on the possibility of the vulnerability of an IP address impacted on honeypots, the transaction to blacklist the IP address as an entry to the *BlackBlock* database is initiated by the orderer service of the hyperledger fabric-based permissioned blockchain [38]. Figure 2 illustrates the processes involved in the `EA-POT` framework in a pictorial form.

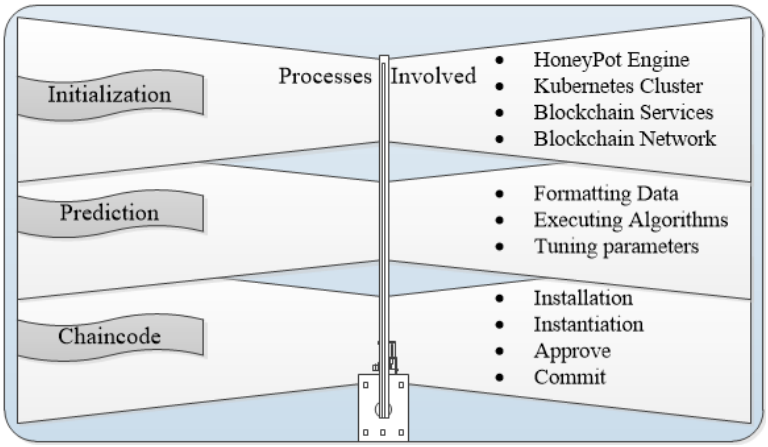


Figure 2: Processes Involved in the `EA-POT` Framework

6 Experimental Results

This section explains the experiments held at the IoT Cloud research laboratory. At first, the experimental setup is explained; next, the validation and prediction results

of identifying the potential IP addresses of hackers using prediction algorithms are explored; next, the application of explainable AI concepts, while including them in blockchains, for approving the transactions is discussed; and, at last, the entry of predicted IPs into the `BlockBlack` immutable database is showcased.

6.1 Experimental Setup

To mimic the scenario of receiving IP addresses into the honeypot engine of the `EA-POT` framework, AWS honeypot dataset [33] was utilized in the experiments. The honeypot dataset had 451581 rows of data with information, such as hacker IP addresses, country of origin, source port address, destination port address, latitude and longitude of the hacker, postal code, protocol, and date/time of the incident. Although any honeypot dataset could be applied for predicting potential hackers, in this work, the AWS honeypot dataset was utilized for the prediction models RFM, LRM, and SVM, to reveal the capability of the framework.

All experiments were carried out on four machines of IoT cloud research laboratory – i) a DELL precision tower 7810 machine which consists of 48 CPUs. This node serves as the master node of the Kubernetes cluster; and, ii) three i7 processor machines which serve as the worker node of the cluster. These nodes were interconnected based on the *Calico* networking policies [34] of the Kubernetes cluster.

On top of the Kubernetes cluster, a hyperledger-based permissioned blockchain was set up with the following configurations: fabric v2.0, dockerv19.03, and `golang` version 1.14. Four docker instances were established that represent the policy stakeholders of `EA-POT`, such as:

```
explainer-ai.com,
ip-approve-authority.com,
cybercrime-aiciiit.com, and
netizen.com.
```

The blockchain network was established using these docker machines that represent the organizations. Each organization had one peer for installing, instantiating, and executing the chaincode policies; the blockchain network had one channel to hold the blockchain ledger consisting of honeypot IPs; the peer of the `cybercrime-aiciiit.com` served as the `orderer` of the permissioned blockchain setup of the `EA-POT` framework.

For providing predictions, algorithms, namely, RFM, LRM, and SVM were written using R version 4.0.0. The prediction algorithms utilized 50 percent training data and the other 50 percent testing data during the validation processes.

6.2 Honeypot Data – Validation of Algorithms

Analyzing honeypot data of AWS using prediction algorithms, such as RFM, LRM, and SVM of the `EA-POT` framework could provide a better insight into the efficiency of the algorithms. Hence, the validation of subsets of data was analyzed. Figure 3

reveals the R^2 values of the prediction results. For RFM experiments, the number of trees was chosen as 100 (ntree=100) and the number of variables sampled at each split was chosen as 2 (i.e., ntry=2). For SVM experiments, the kernel was fixed as "linear"; the coefficient value was fixed as "0"; cache memory was chosen as 40 MB; the tolerance of termination criterion was chosen as 0.001; and, the epsilon value was fixed as 0.1. Additionally, the model was allowed to undergo probability predictions. For LRM experiments, the model type was chosen as "responsive". All prediction experiments were carried out such that the variable "ipnumber" of the dataset was chosen as the dependent variable; and, the independent variables were considered as "country code", "source port address", and "destination port address".

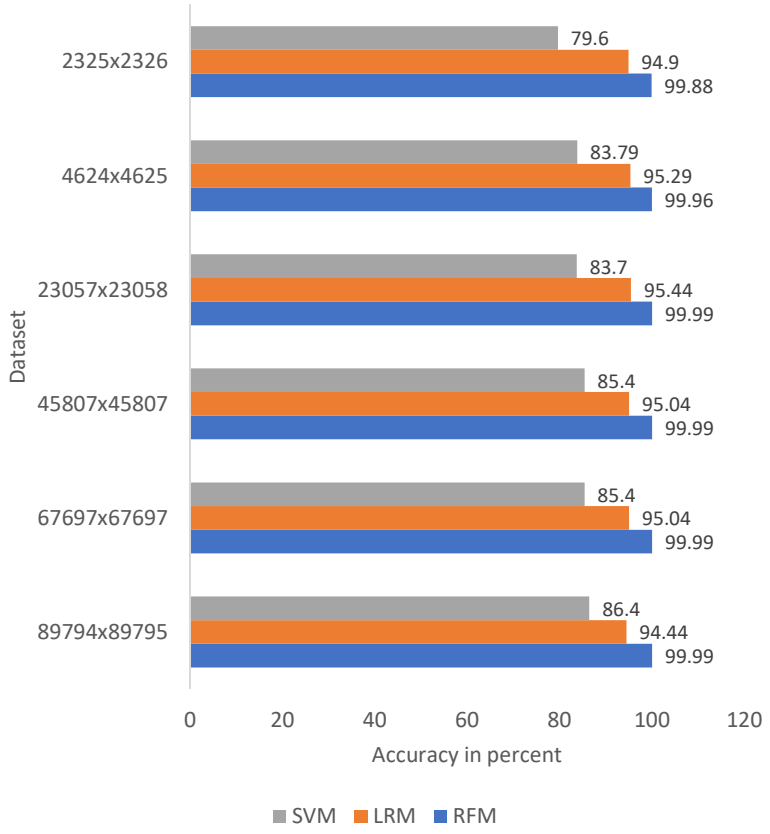


Figure 3: R^2 Values of RFM, LRM, and SVM

The following points could be observed from the Figure 3:

1. RFM algorithm performs well when compared to the other two algorithms of consideration. It could be observed that RFM has achieved around 99.99 percent accuracy when compared to the 85.4 percent accuracy of SVM.

2. Similarly, the prediction algorithm performs better when the training data size increases. For instance, the R^2 value of the SVM algorithm improved from 79.6 percent to 86.4 percent when the data size was increased from 2326 to 89795.

In addition, experiments were performed to study the variation of the prediction accuracy (R^2) while choosing different parameters in modeling algorithms. For instance, the R^2 value of SVM was reduced to 76.4 when the SVM modeling algorithm was executed with kernel="radial", coefficient=0, tolerance=0.01, epsilon = 1, and the probability of prediction was set to `TRUE`.

The time required for predicting these algorithms increased for a certain subset of analysis data. Table 1 illustrates the time required for processing data TDP, time for modeling data TM, and time for predicting data TP.

Table 1: Time Measured in Seconds For Data Processing, Modeling, and Prediction

Dataset	**Algorithm**	**TDP**	**TM**	**TP**
2325x2326	RFM	2.52	0.76	0.02
	LRM	2.42	0.01	0.002
	SVM	2.33	1.3	0.08
4624x4625	RFM	2.76	2.12	0.04
	LRM	2.45	0.017	0.0014
	SVM	2.37	4.95	0.32
23057x23058	RFM	3	3.75	0.23
	LRM	2.66	0.04	0.002
	SVM	2.57	2.3	7.1
45807x45807	RFM	3.26	1.68	0.75
	LRM	3.05	0.211	0.01
	SVM	3.18	11.8	27.3
67697x67697	RFM	3.04	3.48	1.13
	LRM	3.14	0.19	0.04
	SVM	3.82	1.58	58.51
89794x89795	RFM	3.14	5.98	1.107
	LRM	3.13	0.32	0.004
	SVM	3.69	1.08	65.71

Table 1 pinpoints that the modeling time was dependent on the available dataset. Increasing the data size of the dataset had an increase in the modeling and prediction time – i.e., RFM algorithm required $TM = 0.76$ seconds and $TP = 0.02$ seconds for 2325 x 2326; whereas, the same algorithm took over $TM = 1.08$ seconds and $TP = 65.71$ seconds for 89794 x 89795.

Another feature that was observed from Table 1 is the increasing prediction time of SVM when compared to LRM or RFM. Note that the prediction time of SVM reached 65.71 seconds when compared to RFM of 1.107 seconds. The average data

processing time reached 3 seconds for all these prediction algorithms. The data processing involved loading data, initializing dependent and independent variables, and splitting the training and testing dataset of the AWS honeypot data.

In addition, it was observed that varying the parameters of modeling algorithms influenced the TM. For instance, the RFM algorithm showed an increasing modeling time when experimented with more number of splits in variables while constructing the random trees – i.e., TM reached 10.92 seconds when RFM was executed with *ntree=4* in contrary to $TM = 5.98sec.$ for *ntree=2* of 89794 x 89795 dataset (see Table 1).

6.3 Prediction Results

Having validated the model, the potential hacker IP addresses were predicted for the specific location using RFM, LRM, and SVM algorithms. The prediction results obtained for a few candidate locations, when experimented with the RFM algorithm, are shown in Table 2.

The prediction of potential IP addresses that fall prey to the honeypot engine of organizations was reported in Table 2 using RFM prediction algorithm. In fact, the other algorithms could also be reported as similar to RFM. However, the reason for choosing RFM is because of its higher prediction accuracy when compared to the other algorithms namely LRM and SVM.

As shown in Table 2, the potential IP addresses that could harm organizations, that reach the honeypots, were initially predicted as numbers. Later, the numeric IP addresses were converted to IP numbers based on the `iptools` utility of R programs.

Table 2: Prediction of IP Addresses of Honeypot using RFM

Sl.No	**Latitude**	**Longitude**	**IP Addresses**	**Country**
1	37.49	127.02	218.237.65.47	South Korea
2	40.45	-105.46	129.82.138.44	United States
3	52.35	4.9167	8.16.85.133	Netherlands
4	55.154	61.429	31.207.238.106	Russia
5	39.715	-75.5281	199.59.160.152	United States
6	31.8639	117.2808	25.9.68.20	China
7	37.4906	127.02	60.173.14.88	China

6.4 Explainability Analysis

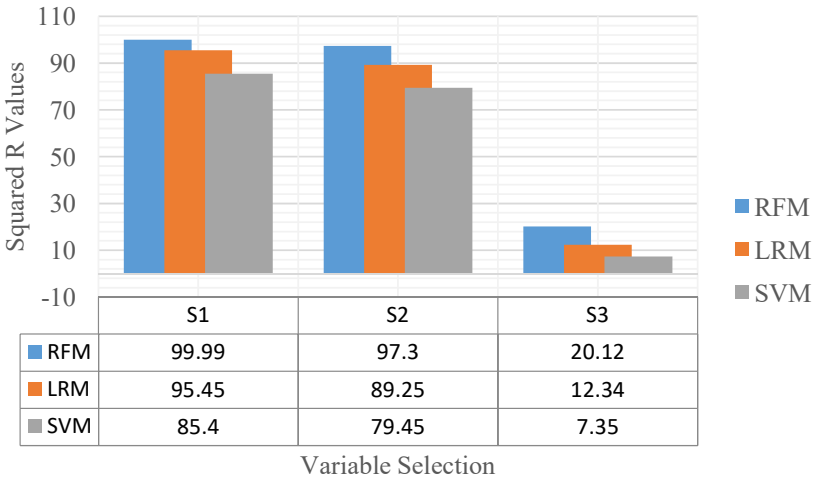
Explainability features of prediction algorithms reveal the prior importance of accurate predictions. The predictions carried out at `EA-POT` framework utilizes R^2 values to explain the importance of independent variables of prediction algorithms.

It is a known fact that most of the available models are considered black boxes – i.e., they may bestow better predictions without hinting at the reasons for achieving a better accuracy or without pointing out the most impacting independent variables for achieving the accuracy. In succinct, apt independent variables must be chosen for gaining better prediction results.

To manifest the influence of the choice of independent variables in the prediction results, experiments were carried out by varying the involvement of independent variables in the prediction processes of prediction algorithms.

In the experiments, three selection options $S1, S2$, and $S3$ were considered. The selections were organized to choose certain columns of the dataset – i.e., $S1$ utilized latitude, date, longitude, country code, source port, and destination port as independent variables while predicting the IP addresses; $S2$ utilized protocol, source port, and destination port addresses; and, $S3$ utilized all variables, such as date of occurrence, hostname, latitude, longitude, country code, source port, destination port, country name, and postal code for predicting the blacklisted IP addresses.

The explainability features of prediction algorithms were utilized by the blockchain chaincodes of the `EA-POT` framework. Figure 4 manifests that the variation in choosing inappropriate independent variables could lead to potential prediction inaccuracies – for instance, $S3$ has only 7.35 percent accuracy while predicting the IP addresses that reach the honeypots.



	S1	S2	S3
RFM	99.99	97.3	20.12
LRM	95.45	89.25	12.34
SVM	85.4	79.45	7.35

Figure 4: Variations in R^2 Values Depending on Independent Variables

6.5 Blockchain Transactions

It is not advisable to blindly choose the predicted IP address and protect the intended organizations or take countermeasures on the defaulters. Listing potential

IP addresses, therefore, needs to be diligently handled.

In `EA-POT` framework, permissioned blockchains using hyperledger fabric were applied. The master and worker nodes are set up such that docker instances representing the peer nodes of the blockchain network are executed on the Kubernetes cluster. During the experiments, the time taken to establish the Kubernetes cluster by the master node on three working nodes was 245.72 seconds. The clustering has several steps, such as creating the master node, joining worker nodes, deploying docker pods that represent the blockchain organizations, and specifying the domain names of the organizations for the fully operational cluster.

Once the predictions were carried out by the master node of the Kubernetes cluster, the predicted IPs are initiated as blockchain transactions by a peer node of the blockchain network. Note that all peer nodes install and instantiate the chaincodes – i.e., the policies for defining whether to register the IP address as blocklist into the *BlackBlock* database. In `EA-POT` framework, the organizations that approve the blockchain transactions are i) IP Approving Authority, ii) `EA-POT` Explainer, iii) CyberCrime Official, and iv) Netizen/Expert.

The time taken by the peer nodes of the blockchain network to install and query chaincodes was 13.35 seconds of which 12.78 seconds were spent on the installation of chaincode policies.

Predicting blacklisted IPs may not be successful at all times due to the accuracy of algorithms. Accordingly, it is not a good solution to blacklist all predicted IPs. Hence, in the proposed framework, the stakeholders of blockchains, based on the policies, decide to collectively agree on the IP addresses before they were registered in the immutable database.

To demonstrate the viability of choosing stakeholders for deciding the registry of IP addresses in the *BlackBlock* database, a few experiments reported in Table 2 were repeated. It was observed that all IP addresses that were predicted by the RFM algorithm in the experiments were not committed to the database – i.e., IP addresses "8.16.85.133" and "25.9.68.20" pointing to the latitude and longitude of countries, such as the Netherlands and China were incorrect. This is because a

Table 3: IP Addresses Committed to the `BlackBlock` Database

Table Entry	Approver 1 (IP Approving Authority)	Approver 2 (Explainer-AI)	Approver 3 (Cybercrime Official)	Approver 4 (Netizen Expert)	BlackBlock (Committed)
1	✓	✓	✓	✓	✓
2	✓	✓	✓	✓	✓
3	✗	✓	✓	✓	✗
4	✓	✓	✓	✓	✓
5	✓	✓	✓	✓	✓
6	✗	✓	✓	✓	✗
7	✓	✓	✓	✓	✓

few predictions could lead to wrong IP addresses when applied using prediction models. Accordingly, all policy stakeholders of the Blockchain network, namely the IP Approving Authority, did not approve the entry of registering the IP addresses to the *BlackBlock* database (see Table 3). Hence, only the IP addresses that were approved by all stakeholders were committed to the database.

Table 3 illustrates the records that were registered into the *BlackBlock* database. The database, being an immutable database, could not be modified by participants, including the intra-organizational participants. Thus, the proposed `EA-POT` framework achieves better efficiency in handling cybercrimes without any modifications to the predicted honeypot IP addresses.

7 Conclusion

The process of converting potential cyber threats into threat discoveries, learning, and ultimately developing security-enabled products, such as honeypots has been evidenced in recent years in various domains, such as IIoT and Cloud environments. Initial efforts to predict the potential hackers, either by establishing honeypots or the other cybersecurity features, predominantly save time and protect the limited compute resources from hackers, especially on cloud-based IoT services. Prediction approaches of the past indicate that blackbox prediction approaches were practiced with limited utility. Additionally, the hacker information was not well-protected, especially when the hacking was carried out within an organization by an insider employee.

This article proposed an Explainable AI-Assisted Blockchain Framework for honeypot IP predictions named `EA-POT` framework. The proposed framework applied explainable features of prediction models, such as Random Forest Modeling, Support Vector Machine, and Linear Regression Modeling, to approve the registry of predicted blacklisted IPs into the Blockchain database along with the other approvers, such as CyberCrime official of a country/region.

Experiments were carried out in the IoT Cloud research laboratory by establishing a hyperledger-fabric permissioned blockchain on top of the Kubernetes cluster consisting of four experimental compute nodes. The experiments manifested the efficiency of the proposed `EA-POT` framework using AWS honeypot use cases. The article explored the findings and reported how the `EA-POT` framework blacklisted potential IPs based on the policy stakeholders involving the explainable AI features of prediction models.

Acknowledgement

The author thanks AIC-IIITKottayam and BEL funding agencies for supporting this research work. In addition, the author thanks the reviewers and the editorial team of the journal for processing the article on time.

References

[1] Anwar, Ahmed H., Kamhoua, Charles, and Leslie, Nandi. Honeypot allocation over attack graphs in cyber deception games. In *2020 International Conference on Computing, Networking and Communications (ICNC)*, pages 502–506, 2020. DOI: 10.1109/ICNC47757.2020.9049764.

[2] Arivudainambi, D., Varun Kumar, K.A., Sibi Chakkaravarthy, S., and Visu, P. Malware traffic classification using principal component analysis and artificial neural network for extreme surveillance. *Comput. Commun.*, 147(C):50–57, nov 2019. DOI: 10.1016/j.comcom.2019.08.003.

[3] Bauer, Johann, Goltz, Johannes, Mundt, Thomas, and Wiedenmann, Simeon. Honeypots for threat intelligence in building automation systems. In *2019 Computing, Communications and IoT Applications (ComComAp)*, pages 242–246, 2019. DOI: 10.1109/ComComAp46287.2019.9018776.

[4] Benedict, Shajulin. Energy efficient aspects of federated learning – mechanisms and opportunities. In Patel, Kanubhai K., Garg, Deepak, Patel, Atul, and Lingras, Pawan, editors, *Soft Computing and its Engineering Applications*, pages 38–51, Singapore, 2021. Springer Singapore. DOI: 10.1007/978-981-16-0708-0_4.

[5] Camino, Ramiro, Torres, Christof Ferreira, Baden, Mathis, and State, Radu. A data science approach for detecting honeypots in Ethereum. In *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–9, 2020. DOI: 10.1109/ICBC48266.2020.9169396.

[6] Dodson, Michael, Beresford, Alastair R., and Vingaard, Mikael. Using global honeypot networks to detect targeted ICS attacks. In *2020 12th International Conference on Cyber Conflict (CyCon)*, Volume 1300, pages 275–291, 2020. DOI: 10.23919/CyCon49761.2020.9131734.

[7] Du, Miao and Wang, Kun. An SDN-enabled pseudo-honeypot strategy for distributed denial of service attacks in industrial Internet of Things. *IEEE Transactions on Industrial Informatics*, 16(1):648–657, 2020. DOI: 10.1109/TII.2019.2917912.

[8] Elijah, Olakunle, Rahman, Tharek Abdul, Orikumhi, Igbafe, Leow, Chee Yen, and Hindia, MHD Nour. An overview of Internet of Things (IoT) and data analytics in agriculture: Benefits and challenges. *IEEE Internet of Things Journal*, 5(5):3758–3773, 2018. DOI: 10.1109/JIOT.2018.2844296.

[9] Foschini, Luca, Gavagna, Andrea, Martuscelli, Giuseppe, and Montanari, Rebecca. Hyperledger fabric blockchain: Chaincode performance analysis. In *2020 IEEE International Conference on Communications (ICC)*, pages 1–6, 2020. DOI: 10.1109/ICC40277.2020.9149080.

[10] Hara, Kazuki, Sato, Teppei, Imamura, Mitsuyoshi, and Omote, Kazumasa. Profiling of malicious users using simple honeypots on the Ethereum blockchain network. In *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–3, 2020. DOI: 10.1109/ICBC48266.2020.9169469.

[11] Hearst, M.A., Dumais, S.T., Osuna, E., Platt, J., and Scholkopf, B. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, 1998. DOI: 10.1109/5254.708428.

[12] Horák, Karel, Bošanský, Branislav, Tomášek, Petr, Kiekintveld, Christopher, and Kamhoua, Charles. Optimizing honeypot strategies against dynamic lateral movement using partially observable stochastic games. *Comput. Secur.*, 87(C), nov 2019. DOI: 10.1016/j.cose.2019.101579.

[13] Jafarian, Jafar Haadi and Niakanlahiji, Amirreza. Delivering honeypots as a service. In *Proceedings of the 53rd Hawaii International Conference on System Sciences*, pages 1835–1844, 2020. DOI: 10.24251/HICSS.2020.227, http://hdl.handle.net/10125/63966.

[14] Kostopoulos, Alexandros, Chochliouros, Ioannis P., Apostolopoulos, Thodoris, Patsakis, Constantinos, Tsatsanifos, George, Anastasiadis, Miltos, Guarino, Alessandro, and Tran, Bao. Realising Honeypot-as-a-Service for Smart Home solutions. In *Proceedings of the 5th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, pages 1–6, 2020. DOI: 10.1109/SEEDA-CECNSM49515.2020.9221787.

[15] Lynda, Boukela, Zhang, Gongxuan, Bouzefrane, Samia, and Zhou, Junlong. An outlier ensemble for unsupervised anomaly detection in honeypots data. *Intelligent Data Analysis*, 24:743–758, 07 2020. DOI: 10.3233/IDA-194656.

[16] Mashima, Daisuke, Li, Yuan, and Chen, Binbin. Who's scanning our smart grid? Empirical study on honeypot data. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2019. DOI: 10.1109/GLOBECOM38437.2019.9013835.

[17] Mattei, Tobias A. Privacy, confidentiality, and security of health care information: Lessons from the recent WannaCry cyberattack. *World Neurosurgery*, pages 972–974, 2017. DOI: 10.1016/j.wneu.2017.06.104.

[18] Pereira Ferreira, Arnaldo and Sinnott, Richard. A performance evaluation of containers running on managed kubernetes services. In *2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 199–208, 2019. DOI: 10.1109/CloudCom.2019.00038.

[19] Rajaboyevich, Gulomov Sherzod, Rustamovna, Salimova Husniya, and o'g'li, Ganiyev Asadullo Mahmud. Characterizing honeypot-captured cyber-attacks:

Statistical framework and case study. *International Journal of Innovative Analyses and Emerging Technology*, 2(5):63–67, May 2022.

[20] Rowe, Neil C., Nguyen, Thuy D., Kendrick, Marian M., Rucker, Zaki A., Hyun, Dahae, and Brown, Justin C. Creating effective industrial-control-system honeypots. In *Proceedings of the 53rd Hawaii International Conference on System Sciences*, pages 1845–1854, 2020. DOI: 10.33423/ajm.v20i2.3003.

[21] Saxena, Ms. Apurva, Ubnare, Gaurav, and Dubey, Anubha. Virtual public cloud model in honeypot for data security: A new technique. In *Proceedings of the 2019 5th International Conference on Computing and Artificial Intelligence*, ICCAI '19, page 66–71, New York, NY, USA, 2019. Association for Computing Machinery. DOI: 10.1145/3330482.3330516.

[22] Shalaby, Salma, Abdellatif, Alaa Awad, Al-Ali, Abdulla, Mohamed, Amr, Erbad, Aiman, and Guizani, Mohsen. Performance evaluation of hyperledger fabric. In *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, pages 608–613, 2020. DOI: 10.1109/ICIoT48696.2020.9089614.

[23] Surnin, Oleg, Hussain, Fatima, Hussain, Rasheed, Ostrovskaya, Svetlana, Polovinkin, Andrey, Lee, JooYoung, and Fernando, Xavier. Probabilistic estimation of honeypot detection in Internet of Things environment. In *2019 International Conference on Computing, Networking and Communications (ICNC)*, pages 191–196, 2019. DOI: 10.1109/ICCNC.2019.8685566.

[24] Tian, Wen, Ji, Xiaopeng, Liu, Weiwei, Liu, Guangjie, Zhai, Jiangtao, Dai, Yuewei, and Huang, Shuhua. Prospect theoretic study of honeypot defense against advanced persistent threats in power grid. *IEEE Access*, 8:64075–64085, 2020. DOI: 10.1109/ACCESS.2020.2984795.

[25] Torres, Christof Ferreira, Steichen, Mathis, and State, Radu. The art of the scam: Demystifying honeypots in Ethereum smart contracts. In *Proceedings of the 28th USENIX Conference on Security Symposium*, SEC'19, page 1591–1607, USA, 2019. USENIX Association. DOI: 10.5555/3361338.3361449.

[26] Vidal-González, Sergio, García-Rodríguez, Isaías, Aláiz-Moretón, Héctor, Benavides-Cuéllar, Carmen, Benítez-Andrades, José Alberto, García-Ordás, María Teresa, and Novais, Paulo. Analyzing IoT-based botnet malware activity with distributed low interaction honeypots. In Rocha, Álvaro, Adeli, Hojjat, Reis, Luís Paulo, Costanzo, Sandra, Orovic, Irena, and Moreira, Fernando, editors, *Trends and Innovations in Information Systems and Technologies*, pages 329–338, Cham, 2020. Springer International Publishing. DOI: 10.1007/978-3-030-45691-7_30.

[27] Wang, Binglai, Dou, Yu, Sang, Yafei, Zhang, Yongzheng, and Huang, Ji. IoTCMal: Towards a hybrid IoT honeypot for capturing and analyzing malware. In

2020 IEEE International Conference on Communications (ICC), pages 1–7, 2020. DOI: 10.1109/ICC40277.2020.9149314.

[28] Ye, Dongdong, Yu, Rong, Pan, Miao, and Han, Zhu. Federated learning in vehicular edge computing: A selective model aggregation approach. *IEEE Access*, 8:23920–23935, 2020. DOI: 10.1109/ACCESS.2020.2968399.

[29] Younis, Fadi and Miri, Ali. Using honeypots in a decentralized framework to defend against adversarial machine-learning attacks. In Zhou, Jianying, Deng, Robert, Li, Zhou, Majumdar, Suryadipta, Meng, Weizhi, Wang, Lingyu, and Zhang, Kehuan, editors, *Applied Cryptography and Network Security Workshops*, pages 24–48, Cham, 2019. Springer International Publishing. DOI: 10.1007/978-3-030-29729-9_2.

[30] Zhan, Zhenxin, Xu, Maochao, and Xu, Shouhuai. Predicting cyber attack rates with extreme values. *IEEE Transactions on Information Forensics and Security*, 10(8):1666–1677, 2015. DOI: 10.1109/TIFS.2015.2422261.

[31] Zhang, Weizhe, Zhang, Bin, Zhou, Ying, He, Hui, and Ding, Zeyu. An IoT honeynet based on multiport honeypots for capturing IoT attacks. *IEEE Internet of Things Journal*, 7(5):3991–3999, 2020. DOI: 10.1109/JIOT.2019.2956173.

[32] Australian cybersecurity expenditure. https://www.austcyber.com/resources/sector-competitiveness-plan-2019/chapter1, accessed in Oct. 2022.

[33] AWS honeypot data. https://www.kaggle.com/casimian2000/aws-honeypot-attack-data, accessed in Oct. 2022.

[34] Calico networking. https://docs.projectcalico.org/getting-started/kubernetes/self-managed-onprem/onpremises, accessed in Oct. 2022.

[35] Centrify forecasts. https://www.centrify.com/about-us/news/press-releases/2020/remote-working-increased-risk-cyber-breach/, accessed in Oct. 2022.

[36] Cloud security forecast. https://www.darkreading.com/cloud/cloud-security-spend-set-to-reach-%24126b-by-2023/d/d-id/1334473, accessed in Oct. 2022.

[37] Hyperledger block data structure. https://hyperledger-fabric.readthedocs.io/en/release-2.2/ledger.html, accessed in Oct. 2022.

[38] Hyperledger fabric. https://hyperledger-fabric.readthedocs.io/en/release-2.2/getting_started.html, accessed in Oct. 2022.

[39] Kubernetes cluster architecture. https://kubernetes.io/docs/concepts/architecture/, accessed in Oct. 2022.

[40] Manual on setting up, using, and understanding random forests v3.1. https://www.stat.berkeley.edu/~breiman/random-forests.pdf, accessed in Oct. 2022.

[41] Randomforest in R. https://cran.r-project.org/web/packages/randomForest/randomForest.pdf, accessed in Oct. 2022.

Received 1st June 2021