An AI-Driven VM Threat Prediction Model for Multi-Risks Analysis-Based Cloud Cybersecurity Deepika Saxena, Member, IEEE, Ishu Gupta, Member, IEEE, Rishabh Gupta,

Ashutosh Kumar Singh, Senior Member, IEEE, and Xiaoqing Wen, Fellow, IEEE

Abstract-Cloud virtualization technology, ingrained with physical resource sharing, prompts cybersecurity threats on users' virtual machines (VM)s due to the presence of inevitable vulnerabilities on the offsite servers. Contrary to the existing works which concentrated on reducing resource sharing and encryption/decryption of data before transfer for improving cybersecurity which raises computational cost overhead, the proposed model operates diversely for efficiently serving the same purpose. This paper proposes a novel Multiple Risks Analysis based VM Threat Prediction Model (MR-TPM) to secure computational data and minimize adversary breaches by proactively estimating the VMs threats. It considers multiple cybersecurity risk factors associated with the configuration and management of VMs, along with analysis of users' behaviour. All these threat factors are quantified for the generation of respective risk score values and fed as input into a machine learning based classifier to estimate the probability of threat for each VM. The performance of MR-TPM is evaluated using benchmark Google Cluster and OpenNebula VM threat traces. The experimental results demonstrate that the proposed model efficiently computes the cybersecurity risks and learns the VM threat patterns from historical and live data samples. The deployment of MR-TPM with existing VM allocation policies reduces cybersecurity threats up to 88.9%.

Index Terms—Hypervisor vulnerability, Network-cascading, Risk analysis, Side-channel, Unauthorized data access.

1. INTRODUCTION

CYBERCRIMES are gobbling up the utility of the cloud services for the beneficiaries, including Cloud Service Providers (CSP)s as well as the end users. According to the estimation of Norton Security, in 2023, cybercriminals will be breaching 33 billion records per year [1]. Also, it has been reported that the misconfiguration and mismanagement associated with the virtualization technology at the cloud platform are the topmost causes of leakage of terabytes of sensitive data of millions of cloud users across the world [2]. Though the CSPs employ sharing of physical resources among multiple users in the view of maximizing the revenues [3]–[7] the discrepancies and unpatched susceptibilities developed during

D. Saxena and R. Gupta are with Department of Computer Science & Engineering, The University of Aizu, Japan. E-mail: 13deepikasaxena@gmail.com, rishabhgpt66@gmail.com. virtualization, produce misconfigured VMs and hypervisors, expediting the occurrence of cyberattacks. A malicious user may initiate a number of VMs and exploit the misconfigured or vulnerable VMs in multiple ways to impose a threat on a target VM [8], [9]. Moreover, the vulnerability of hypervisor due to misconfigured virtualization, devastates the cybersecurity by acquiescing all the coresident VMs to be compromised effortlessly [10]. The mismanagement during physical resource distribution yields co-residency of vulnerable VMs and malicious user VM appealing security threats such as leakage of user's confidential data, hampering of data, unauthorized access via insecure interfaces, hijacking of accounts, etc. [11]– [15]. Therefore, the key challenge for the CSP is: How to minimize the cybersecurity threats due to misconfiguration and mismanagement of shared resources on a cloud platform?

1

A. Related Work

The considerable works presented for preserving cybersecurity of computational data via VM allocation have focused on both defensive strategies as well as preventive strategies. The defensive strategies include minimization of resource sharing by reducing the number of users per server [11], [16], raising the difficulties for achieving co-residency [17], [18], and eliminating side-channel based cyberthreats [19]. While some other researchers have provided preventive strategies merely by periodic migration of VMs [20], [21]. Levitin et al. [22] have presented a method to resist co-residence data theft attacks and improve service reliability by incorporating threshold voting-based N-version programming (NVP). Wu et al. [23] presented a secure and efficient outsourced K-means clustering (SEOKC) scheme for cloud data protection by applying a fully homomorphic encryption with the ciphertext packing technique to attain parallel computation without any excess cost. This scheme preserves data privacy by furnishing database security, privacy of clustering results, and hidden data access patterns. Zhang et al. [24] presented a double-blind anonymous evaluation-based trust model which allows suitable matching between anonymous users and service providers and employed node checking to detect malicious behaviour. A Previously-Selected-Servers-First (PSSF) policy was proposed in [11] for minimization of exposure of benign VMs to malicious ones. Every server maintained a record of a list of users whose VMs were ever hosted on it. The previously assigned servers that have ever hosted VMs of an old user are considered first for allocation of their new VMs. If no such server exists, then a server with more resource capacity among

This article has been accepted in IEEE Transactions on Systems, Man, and Cybernetics: Systems Journal © 2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. This work is freely available for survey and

I. Gupta is with International Institute of Information Technology (IIIT) Bangalore, India. E-mail:ishugupta23@gmail.com

A. K. Singh is a Director with Indian Institute of Information Technology Bhopal, India and a Professor with the Department of Computer Applications, National Institute of Technology, Kurukshetra, India. E-mail: ashutosh@nitkkr.ac.in

Xiaoqing Wen is with the Department of Creative Informatics and the Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology, Fukuoka 8208502, Japan E-mail: wen@cse.kyutech.ac.jp

the remaining servers, is considered for VM placement. Miao et al. [25] improved PSSF by adding a rule that a new VM should be co-located with the user VM to whom it is already co-resident. A hierarchical correlation model for analyzing and estimating reliability, performance, and power consumption of a cloud service is proposed in [26] to locate common causes of co-located multilple VM failures sharing multicore CPUs.

SEA-LB [16] allocates VMs considering minimum power consumption and side-channel attacks with maximum resource utilization by applying modified genetic algorithm approach. The security is provided by minimizing the number of shared servers at the cost of resource utilization. Saxena et al. [15] presented a security embedded resource allocation (SEDRA) model in which the performance of network traffic and inter-VM links are considered to detect and mitigate VM threats by utilizing a random tree classifier. Han et al. [17] proposed a two-player game-based defence mechanism against sidechannel attack, where the potential differences between the attackers' and legal users' behaviour were examined by using clustering and semi-supervised learning techniques for the classification of users. As a result, the attacker's efficiency of achieving co-residency with a target VM raised drastically, thus denying an attack on computational data executing within a VM. A data security risk analysis based VM placement is discussed in [27], where a secure and multi-objective VM allocation is formulated and solved by applying an evolutionary optimization. A Vickrey Clarke-Groves bidding mechanism based defence system was presented in [28] to maximise the difficulty for the adversary to locate the target VM.

B. Our Contributions

In the light of the aforementioned approaches, it is revealed that rigorous control over VM-centered cybercrimes is still in the infancy stage which marks the need to proactively estimate the intensity of VM threats in real-time. Since, machine learning algorithms are capable of extracting and learning useful patterns from known malicious activities rapidly by profiling devices such as VMs and servers, and understanding regular activities, it can intelligently identify previously unknown forms of malware and help protect VMs from potential attacks. Owing to the effective machine-learning capabilities of Extreme Gradient Boosting (XGB) approach including handling missing values, parallelization, distributed computing, and cache optimization, we have devised an XGB inspired VM threat prediction model. Correspondingly, a Multiple Risks Analysis based VM Threat Prediction Model (MR-TPM) is proposed that predicts cyberthreats associated with VMs misconfiguration and their insecure allocation at the cloud platform. To the best of the authors' knowledge, such a proactive VM threat prediction model by considering multiple security risk factors for alleviation of cyberthreats, is presented for the first time. The key contributions are fourfold:

• A novel concept of multiple risks analysis based cybersecurity pertaining to VMs, is proposed to maximize the security of computational data executing on VMs. Also, the ill-effects of misconfiguration and insecure VM management are minimized by considering the intended multiple risk factors.

- · Quantification and assessment of all the considered security threat factors for the periodic training of the newly developed artificial intelligence (AI) driven VM threat prediction model is introduced.
- Implementation and evaluation of the proposed model • using real VM threat traces reveals that MR-TPM predicts threats with precise accuracy and helps to mitigate them before the occurrence.
- Deployment of the proposed model with existing VM • placement policies demonstrates its compatability and applicability in improving the security of user data during execution by exploiting and analysing multiple VM risks for threat prediction. Additionally, its workload prediction component helps to optimize resource utilization, power consumption substantially by minimizing the number of active servers.

A bird eye view of the proposed model is shown in Fig. 1, where multiple types of VM security attack factors $(\{R_1, R_2, ..., R_n\} \in R)$ are gathered, quantified, and analysed to periodically train a machine learning based VM threat estimator for accurate prediction of future threats on VMs.



Organization: The paper is structured as follows: Section II discusses the problem formulation. A detailed elaboration of proposed MR-TPM is conferred in Section III. The multiple cyberthreat factors associated with VMs, including user behaviour analysis, configuration-dependent factors, and allocation-dependent factors, are entailed in Section IV, Section V, and Section VI, respectively. The operational design and complexity of MR-TPM is presented in Section VII. The performance evaluation followed by conclusive remarks and the future scope of the proposed work are presented in Section VIII and Section IX, respectively. Table I shows the list of symbols with explanatory terms used throughout the paper.

2. PROBLEM FORMULATION

A cloud datacenter environment is considered where multiple users requests for execution of their workloads or appli-

TABLE I: Notations with explanatory terms

S: server, V: VM, U: user, P: number of servers, Q : number of VMs,
M: number of users, ω : mapping among server, VM, user
R: security risk factor, L:VM vulnerability, H: Hypervisor vulnerability
S^{Hyp_scor} : Server's hypervisor own vulnerability, Ξ : Threat,
C: Co-residency effects, N: Network cascading effects,
W^p : predicted workload, \mathbb{F} : features used for prediction,
BL: Base Learners in XGB, Θ :unauthorized access, Rq : job request,
\mathbb{H}^{\ddagger} : record of malicious actions, $\mathbb{P}(\Xi)$: probability of threat
RU: resource utilization, PW: power consumption, \mathcal{M}_{c} : migration cost
CC: status of VM after migration, G:status of server after migration
$\mathbb{D}(S_k, S_j)$: distance between servers S_k and S_j

cations. The users can be categorised into legitimate (normal) and malicious (threat-imposing) users. During workload execution, the inter-dependent VMs need to communicate and exchange information to complete the application execution. However, some malicious user VMs may intrude this operation and seek for the security loopholes to exploit various opportunities for launching successful threats to legitimate users' VMs for stealing sensitive information via an unauthorized access. The security of VMs is compromised by exploiting either configuration discrepencies of VMs and associated host servers or insecure allocation and mismanagement of VMs. Accordingly, a problem configuring research assumptions and design goals is formulated in the following subsections.

A. Assumptions

The assumptions addressing conditions for VM threats and the capabilities of malicious user VMs during workload distribution and execution are as follows:

- Only CSP decides mapping between VMs and servers, and it may or may not have the knowledge of legitimate and malicious VMs.
- Each active VM belongs to one user only. However, the user can have multiple number of VMs over time.
- Malicious user may run one or multiple VMs to exploit means of security escape for imposing a threat on target VM(s). The VM threats can be executed in three ways: one-to-one (one specific malicious VM attacks one target VM), one-to-many (one specific malicious VM attacks multiple target VMs in networking), and many-to-many (group of malicious VMs attack many target VMs).
- VM(s) are migrated either to handle over/under-load on the source server or to protect them from malicious activity only. Otherwise, the VM is assumed to run on the same server until the user terminates it.

B. Problem statement and Design Goals

Specifically, the problem is to develop a VM threat prediction model which is trained with data samples considering n probable risk factors addressing security loopholes that estimates VM(s) security threats proactively to improve cybersecurity during cloud workload processing. Based on the aforementioned problem assumptions and statement, the design goals of the proposed model are as follows:

• To develop a machine learning-driven model that will determine VM threats prior to occurrence in real-time. This model must not effect the efficiency of VM management and it must be adaptable and compatable for operation with any VM allocation policies.

- To generate a knowledge database for training of the corresponding VM threat predictor by identifying and computing risk score values for all the probable security factors associated with VM(s).
- To accurately detect security threats on legitimate VM(s) due to presence of malicious VM and vulnerabilities of VM(s) configuration and management.

3. PROPOSED VM THREAT PREDICTION MODEL

Consider a cluster of P servers $\{S_1, S_2, ..., S_P\} \in \mathbb{S}$ hosts Q VMs $\{V_1, V_2, ..., V_Q\} \in \mathbb{V}$ of M users $\{U_1, U_2, ..., U_M\} \in$ U. Let S_1 hosts x VMs such that $\{V_1^1, V_2^1, ..., V_M^H\} \in V^1$; S_2 and S_P host y VMs $\{V_1^2, V_2^2, ..., V_y^2\} \in V^2$ and z VMs $\{V_1^P, V_2^P, ..., V_z^P\} \in V^P$, respectively, where $\{V^1, V^2, ..., V^P\} \in V$ and $\{x \cup y \cup z\} \subseteq Q$. A mapping $\omega | \omega : \mathbb{U} \times \mathbb{V} \mapsto \mathbb{S}$ assigns VMs of each user on a specific server such as $\omega_{ji}^k = 1$ iff j^{th} VM of k^{th} user is deployed on i^{th} server. The comprehensive description of the essential blocks and intrinsic information flow of MR-TPM is depicted in Fig 2. The proposed cyberthreat prediction model records and analyses multiple security risk factors $\{R_1, R_2, R_3, R_4, R_5\} \in R$ associated with a VM configuration, such as VM vulnerability $\{L_1, L_2, ..., L_Q\}$, server Hypervisor vulnerability $\{H_1, H_2, ..., H_P\}$; and VM allocation, including Side-channel effect $\{C_1, C_2, ..., C_Q\}$ and Network cascading effect $\{N_1, N_2, ..., N_Q\}$; User behaviour $\{U_1^*, U_2^*, ..., U_M^*\}$; and previous records of VM threats $\{\Xi_1, \Xi_2, ..., \Xi_n\}$. During time-interval $\{t_a, t_b\} \in t$, all the security risk factors and threat information are collected and categorized into four classes:

- User behaviour analysis $\{U_1^*, U_2^*, ..., U_M^*\}$ (Section 4)
- VM Configuration-dependent factors for computation of the scores of VM vulnerability {L₁, L₂, ..., L_Q} and server hypervisor vulnerability scores {H₁, H₂, ..., H_P} (Section 5)
- VM Allocation-dependent factors for assessment of sidechannel effects { $C_1, C_2, ..., C_Q$ } and network cascading effects { $N_1, N_2, ..., N_Q$ } (Section 6)
- Records of live threats or malicious actions on VMs for updation of VM threat database

Definition 1. VM cyberthreat prediction: The mechanism intended for computation and analysis of various security escapes and unpatched discrepancies associated with a VM along with proactive threat estimation, is designated as VM cyberthreat prediction.

MR-TPM proactively estimates the workload information $\{W_1^p, W_2^p, ..., W_Q^p \in W\}$ by utilizing a neural network based workload predictor (Pr), which is periodically trained with the latest and historic resource utilization $\{RU_1, RU_2, ..., RU_Q\}$ by VMs $\{V_1, V_2, ..., V_Q\}$ to determine active VMs $\{\hat{V}_1, \hat{V}_2, ..., \hat{V}_{Q^*}, Q^* \subseteq Q\}$ having predicted workload $(W^p) > 0$. The prior knowledge of active VMs is utilized for analysis of the placement of VMs during the next



Fig. 2: Multiple Risks Analysis based VM Threat Prediction Model (MR-TPM)

time interval $\{t_{a+1}, t_{b+1}\} \in t$. The consecutive processes of feature selection (\mathbb{FS}) and threat prediction (\mathbb{TP}) is performed for active VMs based on the predicted workload information $\{W_1^p, W_2^p, ..., W_Q^p\}$ for VMs $\{V_1, V_2, ..., V_Q\}$ of users $\{U_1, U_2, ..., U_M\}$ during time-interval $\{t_{a+1}, t_{b+1}\}$. The historical database of VM threats is utilized for feature selection, followed by training of online VM threat predictor \mathbb{TP} , which is periodically re-trained with the latest data samples for online VM threat prediction $\{\Xi_1^p, \Xi_2^p, ..., \Xi_{Q^*}^p\}$. Among all the collected and analysed features $\{L, H, \check{C}, N, \mathbb{V}, \mathbb{S}, \mathbb{U}, \mathbb{U}^*, W^p, \omega\}$ etc.} $\subseteq \mathbb{F}$, only useful features are filtered (i.e., \mathbb{F}^*) by applying Recursive Feature Elimination (RFE) to train threat predictor \mathbb{TP} to estimate VM threats $\{\Xi_1^p, \Xi_2^p, ..., \Xi_{O^*}^p\}$ with improved accuracy and reduced computation time. The data samples containing selected features $\{\mathbb{F}_1^*, \mathbb{F}_2^*, ..., \mathbb{F}_s^*\} \in \mathbb{F}^*$ are split into training samples $\{\bar{\mathbb{F}}_1^{**}, \bar{\mathbb{F}}_2^{**}, ..., \bar{\mathbb{F}}_{s^*}^{**}\} \in \bar{\mathbb{F}}^{**}$ and testing samples $\{\mathbb{F}_1^{**}, \mathbb{F}_2^{**}, ..., \mathbb{F}_{t^{**}}^{**}\} \in \mathbb{F}^{**}$ subject to the constraints: (i) $\mathbb{F}^* = \overline{\mathbb{F}}^{**} \cup \mathbb{F}^{**}$ (ii) $\overline{\mathbb{F}}^{**} \cap \mathbb{F}^{**} = \emptyset$ (iii) $\{s^*, s^{**}\} \subseteq s$ where s is total number of data samples. A mapping $\{\Omega | \Omega : \overline{\mathbb{F}}^{**} \times \mathbb{TP} \Rightarrow \mathbb{TP}^*\}$ trains threat predictor \mathbb{TP} with $\overline{\mathbb{F}}^{**}$ to generate Trained Predictor (\mathbb{TP}^*) during training phase while $\{\Omega^* | \Omega^* : \mathbb{F}^{**} \times \mathbb{TP}^* \Rightarrow \mathbb{TP}^{**}\}$ evaluates \mathbb{TP}^* with unseen test data \mathbb{F}^{**} to generate Tested Predictor (\mathbb{TP}^{**}) for online VM threat prediction.

The proposed VM threat predictor utilizes an Extreme-Gradient Boosting (XGB) based machine learning algorithm to learn and develop the precise correlations among extracted patterns for accurate prediction of cyberthreats: $\{\Xi_1^p, \Xi_2^p, ..., \Xi_{Q^*}^p\}$. Let a threat predictor (TP) is composed of l base learners (i.e., decision trees) $\mathbb{BL}^* = \{\mathbb{BL}_1^*, \mathbb{BL}_2^*, ..., \mathbb{BL}_l^*\}$ and predicts output $\mathbb{O}^* = \{\mathbb{O}_1^*, \mathbb{O}_2^*, ..., \mathbb{O}_l^*\}$ using Eq. (1), where $\mathbb{F}_i \subseteq \mathbb{F}^*$ such that \mathbb{F} represents the input vector of size s^* . During each iteration, decision trees are trained incrementally to reduce prediction errors and the amount of error reduction is computed as gain or loss term using Eq. (2). The expressions $L(\mathbb{O}^*, \mathbb{O}_{t-1}^* + \mathbb{BL}_t^*(\mathbb{F}_i))$ and $\Psi(\mathbb{BL}_t^*)$ are loss term and a regularization term, respectively. Taylor expansion is applied to compute the exact loss for different possible base learners, which updates Eq. (2) to Eq. (3); where $g_i = \partial_{\mathbb{O}_{t-1}^{**}} L(\mathbb{O}^*, \mathbb{O}_{t-1}^{**})$, and $h_i = \partial_{\mathbb{O}_{t-1}^{**}}^2 L(\mathbb{O}^*, \mathbb{O}_{t-1}^{**})$ are first and second order derivatives of loss function in the gradient, respectively. The term $\Psi(\mathbb{BL}_t^*)$ is computed using Eq. (4), where γ and λ are L_1 and L_2 regularisation coefficients, respectively, w is internal split tree weight and K is the number of leaves in the tree.

$$\mathbb{O}^* = \sum_{z=1}^{l} \mathbb{BL}_z^*(\mathbb{F}_i) \quad \forall i \in \{1, 2, ..., s^*\}$$
(1)

$$L_{t} = \sum_{i=1}^{s} L(\mathbb{O}^{*}, \mathbb{O}_{t-1}^{**} + \mathbb{BL}_{t}^{*}(\mathbb{F}_{i})) + \Psi(\mathbb{BL}_{t}^{*})$$
(2)

$$L_{t} = \sum_{i=1}^{s^{*}} \left[g_{i} \mathbb{B} \mathbb{L}_{t}^{*}(\mathbb{F}_{i}) + \frac{1}{2} h_{i} \mathbb{B} \mathbb{L}_{t}^{*}(\mathbb{F}_{i}) \right] + \Psi(\mathbb{B} \mathbb{L}_{t}^{*}) \quad (3)$$

$$\Psi(\mathbb{BL}_t^*) = \gamma K + \frac{1}{2}\lambda ||w||^2 \tag{4}$$

During each time-interval $\{t_a, t_b\} \in t, a < b$, live selected features $\hat{\mathbb{F}}^{**}$ are given as input to the above discussed threat predictor \mathbb{TP}^{**} to estimate the status of threat Ξ for VMs $\{\hat{V}_1, \hat{V}_2, ..., \hat{V}_Q^*\}$ in the next time-interval $\{t_{a+1}, t_{b+1}\} \in t, a < b$. Accordingly, the process of VM-threat handling is performed for the VMs with predicted threat-status ($\hat{V}_i^{\Xi} > 0$: $i \in [1, 2, ...Q^{**}], Q^{**} \subseteq Q^* \subseteq Q$) by shifting them to a server where the possibility of threat is least ($\hat{V}_i^{\Xi} = 0$). A detailed description of VM security risk factors is provided in the subsequent sections.

4. User behaviour analysis

 $\{U_1, U_2, ..., U_M\}$ submit Users job requests $\{Rq_1, Rq_2, ..., Rq_M\}$ during time-interval $\{t_a, t_b\}$ at the cloud platform as depicted in Fig. 3, where the users are classified into Trusted, Non-trusted and Unknown users. The k^{th} user U_k behaviour is defined in accordance with the actions of its VMs as follows: Trusted: The user behaviour is trusted when the VMs of known user U_k (having historical records of VM resource usage), execute assigned load efficiently without interrupting and interfering with other co-located VMs via an unauthorised access, irrespective of the presence of any vulnerabilities of software or hardware. Non-trusted: A user U_k is non-trusted in case of the users VM attempt any kind of cybercrime or malicious activity such as unauthorized data access, data hijacking, data phishing, etc. by leveraging the susceptibilities of cloud virtualization technology. Unknown: The new user for which there are no records of any previous VM usage, is considered as unknown user. User behaviour analysis deals with the process of critical monitoring, recording, and examining the traces of their previous VM usage and the inter-relationships among co-resident VMs of different users periodically to interpret or investigate the occurrence of cyberthreats in the presence of intended vulnerabilities of cloud environments. The class of user and the selected VM placement policy are passed to the load balancer, which makes VM management decisions. Accordingly, the VMs are deployed on different servers to compute the users' data $\{Rq_1, Rq_2, ..., Rq_M\}$. Concurrently, the VM usage traces or type of data access information is collected and passed to 'VM usage database' for examination of the user behaviour.

Definition 2. VM usage database (DB): The historical repository of data values concerning VM usage related attributes such as its ephemeral user ID, CPU, memory, and bandwidth usage, inter-communication links with other VMs, types of authorized access, etc., constitute VM usage database which is utilized for multiple risks computation, training of resource usage predictor, and VM threat predictor.



Fig. 3: User classification

The new VM of k^{th} user (U_k) is allocated according to the analysis of U_k behaviour by applying Eq. (5), where Θ_k represents malicious actions for e.g., unauthorized access executed by U_k .

$$U_{k} = \begin{cases} Trusted(0), & If(\Theta_{k} = 0)\\ Non - trusted(1), & If(\Theta_{k} > 0)\\ Unknown(-1), & \text{otherwise} \end{cases}$$
(5)

Theorem 1. The behaviour of user U_k^* having VM $V_i^{k^*}$ hosted on server S_j is bounded by Θ such that for a given time-period $\{t_a, t_b\}$ and VM usage database ($\mathbb{DB} \neq \phi$), if $\Theta_k^* = 1$, U_k^* is non-trusted; otherwise, it is trusted.

Proof. Let $\Theta_{ij,k\Rightarrow i^*j,k^*}$ represents a data access by a user U_k^* owning VM $V_{i^*}^{k^*}$ to k^{th} user U_k VM V_i^k during time $\{t_a, t_b\}$, is formulated in Eq. (6):

$$\int_{t_a}^{t_b} \Theta_{ij,k\Rightarrow i^*j,k^*} dt = \int_{t_a}^{t_b} (\omega_{ij}^k \times \omega_{i^*j}^{k^*}) \times \uplus_{ij,k\Rightarrow i^*j,k^*}^v dt$$
(6)

where, $\boxplus_{ij,k\Rightarrow i^*j,k^*}^{k^*}$ represents inter-VM relationship between V_i^k and $V_{i^*}^{k^*}$, $\forall \{i, i^*\} \in Q, j \in P$. It is a Boolean value, 1 for unauthorised data access (i.e., non-trusty relation) and 0 for trusty relation. Assume LA (stated in Eq. (7)) specifies set of authorized inter-VM links for i^{th} VM of k^{*th} user. The inter-VM relationship $(\boxplus_{ij,k\rightarrow i^*j,k^*}^v)$ between V_i^k and $V_{i^*}^{k^*}$ placed on j^{th} server is evaluated in Eq. (8) which corresponds

to the inter-VM links $\mathbb{L}A_{ij,k\Rightarrow i^*j,k^*}$ between them.

$$\mathbb{L}\mathbb{A}^{V_{i,k^*}} \in \{\mathbb{L}\mathbb{A}_1^{V_{i,k^*}}, \mathbb{L}\mathbb{A}_2^{V_{i,k^*}}, ..., \mathbb{L}\mathbb{A}_n^{V_{i,k^*}}\}$$
(7)

Hence, when the user U_k^* has attempted an unauthorised access, the inter-VM relationship parameter $\bigcup_{ij,k\to i^*j,k^*}^{v}$ is equal to 1 and applying Eq. (8) in Eq. (6), $\Theta_k^* = 1$ for U_k^* is non-trusted, and trusted, otherwise.

Corollary 1. The user U_k^* behaviour is also reflected by the relationship $\bigcup_{ij,k\to S_j}^{S}$ between user U_k^* and server S_j which is 'non-trusty' for malicious records $(\mathbb{H}^{\ddagger}_j)$ greater than 0, otherwise, it is trusty.

Proof. Let an unauthorized data access $\Theta_{ij,k^* \Rightarrow S_j}$ from i^{th} VM of k^{*th} user to server S_j during time $\{t_a, t_b\}$ is formulated in Eq. (9). The term $\bigcup_{ij,k^*}^{S} = \{0,1\}$ signifies a relationship between S_j and U_k^* , such that it is equals to a Boolean value, 1 for an unauthorized data access via malicious hypervisor, and 0 otherwise.

$$\int_{t_a}^{t_b} \Theta_{ij,k\Rightarrow S_j} dt = \int_{t_a}^{t_b} \omega_{ij,k} \times \bigcup_{ij,k\to S_j}^{S} dt \quad \forall \{i\} \in Q, j \in P$$
(9)

Suppose the relation $(\bigoplus_{ij,k \to S_j}^S)$ between user U_k and server S_j is analysed using Eq. (10), where \mathbb{H}^{\ddagger}_j represents malicious actions records computed using Eq. (11).

$$\mathbb{H}^{\ddagger}_{j} = \sum \omega_{ij}^{k} \times \omega_{i^{*}j}^{k^{*}} \times \Theta_{ij,k \Rightarrow i^{*}j,k^{*}}$$
(11)

If user U^{k^*} is non-trusty, then $\Theta_{ij,k \Rightarrow i^*j,k^*} = 1$ (as proved in Theorem 1). Accordingly, the value of the term \mathbb{H}^{\ddagger}_j is also 1. Putting $\mathbb{H}^{\ddagger}_j = 1$ in Eq. (10) when $H_j > H_{Thr}$, the value of $\biguplus_{ij,k \to S_j}^S$ becomes 1. Hence, $\mathbb{H}^{\ddagger}_j > 0$ for a non-trusty behaviour of user U^{k^*} .

Further, the total threat information or unauthorized data access Θ_k for the duration $\{t_a, t_b\}$ by U^k is compiled by applying Eq. (12):

$$\int_{t_a}^{t_b} \Theta_k dt = \int_{t_a}^{t_b} (\Theta_{ij,k \Rightarrow i^*j,k^*} + \Theta_{ij,k \Rightarrow S_j}) dt$$
(12)

The Random Forest Classifier (RFC) classifies users $U_1, U_2, ..., U_M$ on the basis of their future behaviour by utilizing the learning capability of different base learners or decision trees and knowledge driven via extracted correlated patterns from their historical information, where Eq. (5) is evaluated periodically for duration $\{t_a, t_b\} \in t$. RFC is composed of n^* base learner estimators that produce n^* outcomes and apply majority voting to predict absolute behaviour of user U_k .

5. CONFIGURATION-DEPENDENT FACTORS

The vulnerabilities of virtualisation technology and VM security loopholes which are governed by the susceptibilities related to the creation and installation of VMs, including sharing of a common physical machine, hypervisor or guest OS installation, are confined to configuration-dependent risks. MR-TPM considers two configuration dependent security risk factors (R_2, R_3) , including VM vulnerability (L) [29] and Hypervisor vulnerability (H) [30]. A malicious user (U^{Mal} : $U^{Mal} \subseteq \mathbb{U}$) launches multiple applications $(A_p, A_q, ..., A_t)$ to compromise the target benign VM $(V^{Ben} : V^{Ben} \subset \mathbb{V})$ by achieving co-residency and exploiting VM and hypervisor vulnerabilities, as shown in Fig. 4. The application A_p of U^{Mal} exploits the hypervisor vulnerability of server S_1 (i.e., $H_1 > H_{Thr}$) and compromises multiple VMs. At server S_p , the applications A_s and A_t of U^{Mal} utilize the vulnerability of VM V_2 (i.e., $L_2 > L_{Thr}$) to launch the attack and hamper computational data on it. The parameters H_{Thr} and L_{Thr} specify threshold values of hypervisor vulnerability and VM vulnerability, respectively. At server S_2 , both kinds of vulnerabilities are absent, i.e., the threshold values of VM vulnerability as well as hypervisor vulnerability are lesser than their respective threshold values, and all the VMs deployed on it are secured ($V^{\Xi} = 0$).

The vulnerable VMs are deprived of standard security features with respect to the operating system, applications like e-mail, web-browsing, and network protocols, and are prone to loose administrative control. Besides this, vulnerable hypervisors of servers leads to hyperjacking where U^{Mal} can easily gain unauthorized access of hypervisor to compromise all the hosted VMs and the applications running on them. It is typically launched against Type 2-Hypervisors running over a host operating system. A mapping $\{\varpi | \varpi : A_m \times U^{Mal} \times V_i \Rightarrow$ $V_i^{Mal}\}$ defines malicious VMs such that an i^{th} VM (V_i) becomes malicious, if it hosts m^{th} application (A_m) of U^{Mal} . The probability of threat ($\mathbb{P}(\Xi_i)$) for i^{th} VM over time-interval $\{t_a, t_b\}$ can be defined using Eq. (13),

$$\mathbb{P}(\Xi_{i}) = \begin{cases} 1, & If(L_{i} > L_{Thr} & \&\& & \omega_{ji}^{k} \cap \omega_{ji^{*}}^{k^{*}} = S_{j}) \\ 1, & If(H_{j} > H_{Thr} & \&\& & \omega_{ji}^{k} \cap \omega_{ji^{*}}^{k^{*}} = S_{j}) \\ 0, & \text{otherwise} \end{cases}$$
(13)

where $t_a < t_b$ and $\omega_{ji}^k \cap \omega_{ji^*}^{k^*} = S_j$ signifies co-location of i^{th} VM (V_i) of k^{th} being user $(U^{Ben}|U^{Ben} \subseteq \mathbb{U})$ and i^{*th} malicious VM $(V_{i^*} \subseteq V^{Mal})$ of k^{*th} malicious user U^{Mal} at j^{th} server (S_j) .

A. VM vulnerability

1

The VMs vulnerability score list is generated using vulnerability scanner tools, such as Common Vulnerability Scoring System (CVSS), Nessus and Qualys [30]. The CVSS measures the severity of vulnerabilities of a hardware or software and produces a score in the range [0, 10]. It quantifies the vulnerability risk score (L) of i^{th} VM in the range [0, 1] by applying Eq. (14).

$$L_i = \frac{V_i^{Score}}{10} \quad \forall i \in [1, Q], V^{Score} \in [1, 10]$$
(14)



Fig. 4: VM and hypervisor vulnerability based threats

B. Hypervisor vulnerability

The security risk of a hypervisor (H) depends on its own vulnerability (S^{Hyp_scor}) as computed in Eq. (15) by applying CVSS score system and the vulnerability of the VMs hosted on it. The overall vulnerability score of hypervisor H_j is given by Eq. (16), where $max(L_i \times \omega_{ij})$ represents maximum VM vulnerability score (L) among all VMs hosted on server S_j , $\forall i \in [1, Q], j \in [1, P], \omega_{ij} = 1$ if S_j hosts V_i .

$$S_j^{Hyp_scor} = \frac{S_j^{Score}}{10} \quad \forall S^{Score} \in [1, 10]$$
(15)

$$\int_{t_a}^{t_b} H_j dt = \int_{t_a}^{t_b} S_j^{Hyp_scor} (1 + max(L_i \times \omega_{ij})) dt \quad (16)$$

6. Allocation-dependent factors

The cybersecurity risk factors pertaining to the distribution of physical resources and assignment of VMs on physical servers subject to resource availability constraints, characterize allocation-dependent risk factors. The VM security risks due to the Side-channel effect and Network cascading effect depend upon the placement of VMs of different users on available servers (i.e., $\mathbb{U} \times \mathbb{V} \Rightarrow \mathbb{S}$). Two VMs V_i and V_j are interdependent $iff(V_i, V_j) \in \mathbb{LA}$, where \mathbb{LA} implies Legal Access subject to the constraints:

• $V_i(\mathbb{LA})V_i \quad \forall_{V_i} \in \mathbb{LA},$

•
$$V_i(\mathbb{LA})V_j = V_j(\mathbb{LA})V_j \quad \forall_{V_i,V_j} \in \mathbb{LA},$$

•
$$\{V_i(\mathbb{LA})V_j \cup V_j(\mathbb{LA})V_k\} \Rightarrow V_i(\mathbb{LA})V_k,$$

•
$$\forall_{V_i, V_i, V_h} \in \mathbb{LA}$$

As depicted in Fig. 5, a malicious user U^{Mal} executes an application at V_1 hosted on the server (S_1) having an effective VM vulnerability, i.e., $L_1^1 > L_{Thr}$, achieves co-residency with one of the inter-dependent VMs ($\{V_1, V_2, ..., V_Z\} \in \mathbb{L}A$), where Z is the number of inter-dependent VMs. The malicious VM (V_1^{Mal}) successfully launches side-channel threat on vulnerable VM (V_2) and the threat propagates to multiple VMs

crossing physical boundaries of network devices using network cascading effect via inter-communication links among VMs: $\{V_1, V_2, ..., V_Z\} \in \mathbb{LA}$. The probability of threat $(\mathbb{P}(\Xi_i))$ for i^{th} VM over time-interval $\{t_a, t_b\}$ is defined using Eq. (17), where $\mathbb{C}_{ii^*j}^* = \omega_{ij} \times \omega_{i^*j} = \{0, 1\}$ is a Boolean variable which specifies co-location between i^{th} VM (V_i) and i^* malicious VM (V_i^*) at server (S_j) .

$$\mathbb{P}(\Xi_i) = \begin{cases} 1, & If((L_i > L_{Thr} \lor H_j > H_{Thr}) \land \mathbb{C}^*_{ii^*j}) \\ 1, & If(\Pi^Z_{k=1}(\mathbb{C}^*_{ki^*j^*} \times \mathbb{C}^*_{ikj} \times L_k) > L_{Thr}) \\ 0, & \text{otherwise} \end{cases}$$
(17)

A. Side-channel effect

Let a malicious VM V_j^{Mal} and benign VM V_i^{Ben} are hosted on server S_k . If V_j^{Mal} compromises any VM on server S_k , then it can compromise other co-resident VMs eventually. Hence, the survival of V_i^{Ben} depends on its own vulnerability score (L_i) and its co-resident VMs vulnerability score. The side-channel risk score (C) of V_i^{Ben} during time-interval $\{t_a, t_b\}$ is calculated as stated in Eq. (18), where $\omega_{jk} \times \omega_{ik}$ represents co-location of i^{th} and j^{th} VM on k^{th} server, $\forall i, j \in [1, Q], k \in [1, P]$.

$$\int_{t_a}^{t_b} C_i dt = \int_{t_a}^{t_b} 1 - \prod_{j=1}^Q (1 - L_j \times \omega_{jk} \times \omega_{ik}) dt \quad (18)$$

B. Network cascading effect

The impact of cascading network connections among VMs on cloud security establishes the network cascading effect. It is computed in terms of network cascading score (N) respective to VM V_i during time-interval $\{t_a, t_b\}$ using Eq. (19), where V_i and V_j are connected via legal access network link and hosted on different servers S_k and S_{k*} such that



Fig. 5: Side channel and Network cascading threats

 $\forall i, j \in [1, Q], i \neq j$. If a malicious VM V^{Mal} hosted on server S_{k^*} , is successful in compromising the VM V_j , then it can compromise VM V_i and all other VMs that are connected via common network by exploiting the network paths.

$$\int_{t_a}^{t_b} N_i dt = \int_{t_a}^{t_b} 1 - \prod_{j=1}^Q (1 - L_j \times \omega_{ik} \times \omega_{jk^*}) dt \quad (19)$$

7. OPERATIONAL DESIGN AND COMPLEXITY

MR-TPM utilizes values of current state of multiple security attack factors $\{R_1, R_2, R_3, R_4, R_5\}$ and three historical databases namely (i) VMs' resource utilization { RU_1 , RU_2 , ..., RU_Q $\in \mathbb{RU}_{db}$, (*ii*) user-records $\{U_1, U_2, ..., U_M\} \in \mathbb{U}_{db}$ and (*iii*) VM threats traces $\{\Xi_1, \Xi_2, ..., \Xi_n\} \in \mathbb{T}\mathbb{h}_{db}$. The set of users $U_1, U_2, ..., U_M$, servers $S_1, S_2, ..., S_P$ and $V_1, V_2, ...,$ V_Q are initialized followed by a mapping $\mathbb{U} \times \mathbb{V} \Rightarrow \mathbb{S}$ among VMs, users and servers. The VMs are allocated to servers using some suitable VM placement strategy, for example, First-Fit Decreasing (FFD), Best-Fit, Greedy, Random-Fit etc. Thereafter, for each consecutive time-intervals $\{t_a, t_b\}$, current resource utilization of V_1 , V_2 , ..., V_Q are passed as input into a workload predictor [31] trained with $\mathbb{R}\mathbb{U}_{db}$ to estimate their resource utilization during next time-interval. The threat status prediction is conducted for the VMs with predicted workload estimation ($W^p > 0$). To predict future threat status of VM V_i , values of R_1, R_2, R_3, R_4 associated with V_i are assessed by applying Eqs. (14)-(19). The assessment of R_5 is done by analysing the behaviour of co-resident users of V_i by using RFC based user classifier trained with \mathbb{U}_{db} . The current score values of R_1, R_2, R_3, R_4, R_5 are fed as input into threat predictor (\mathbb{TP}^{**}) trained and tested with \mathbb{Th}_{db} , to predict the future threat status (\hat{V}_i^{Ξ}) of V_i . Accordingly, the VMs with $\hat{V}_i^{\Xi} > 0$ are migrated to server where $\hat{V}_i^{\Xi} = 0$ by applying Eq. (20). The migration cost is computed using Eq. (21), where $\mathbb{D}(S_k, S_j)$ is the distance or number of hops covered by migrating VM V_{mig} from source (S_k) to destination server S_j , $\{j, k \in [1, P]\}, V_{mig} \in \mathbb{TP}_V, WW(V_{mig}) = V_{mig}^{CPU} \times V_{mig}^{Mem}$ is the size of migrating VM, \mathbb{TP}_V is the list of VMs with 'unsafe' status or VMs on overloaded server (S_k) . The first term $\sum \mathbb{CC}_{mig.j}\mathbb{D}(S_k, S_j) * WW(V_{mig})$ signifies network energy consumed during VM migration. The second term $\sum \mathbb{G}_j \mathbb{B}_j$ specifies server state transition energy, where if i^{th} VM is placed at j^{th} server after migration, then $\mathbb{CC}_{mig.j} = 1$, otherwise 0. If j^{th} server receives one or more VMs after migration, then $\mathbb{G}_j = 1$ else it is 0. Similarly, if $\mathbb{B}_j = 0$, then j^{th} server is active before migration, otherwise, $\mathbb{B}_j = \mathbb{E}_{tr}$ where $\mathbb{E}_{tr} = 4260$ Joules which is energy consumed in switching a server from sleep to active state [32], [33].

$$V_i^{mig_status} = \begin{cases} 1 & If(\hat{V}_i^{\Xi} > 0) \\ 0 & \text{otherwise} \end{cases}$$
(20)
$$\mathcal{U}_c = \sum \mathbb{CC}_{mig.j}(\mathbb{D}(S_k, S_j) * \mathbb{WW}(V_{mig})) + \sum \mathbb{G}_j \mathbb{B}_j$$
(21)

Л

The operational summary for proposed work is depicted in Algorithm 1. Step 1 initializes the list of VMs, servers, users

Algorithm 1: MR-TPM Operational Summary
1 Initialize: $List_{\mathbb{U}}$, $List_{\mathbb{V}}$, $List_{\mathbb{S}}$, ω ;
2 Allocate $V_1, V_2,, V_Q$ to $S_1, S_2,, S_P$ by defining a
mapping $\mathbb{U} \times \mathbb{V} \Rightarrow \mathbb{S}$;
3 for each time-interval $\{t_a, t_b\}$ do
4 $[V_i^{Pred}] \leftarrow \text{Workload Prediction}(V_i)$
$\forall i \in \{1, 2,Q\}$;
5 if $(V_i^{Pred} > 0)$ then
6 $[\hat{V}_i^{\Xi} > 0] \Leftarrow \text{Threat Predictor } (\mathbb{TP}^{**});$
7 if $\hat{V}_i^{\Xi} ==$ 'unsafe' then
8 Migrate V_i to server S_k such that $\hat{V}_i^{\Xi} ==$
'safe';
9 Compute \mathcal{M}_c by applying Eq. (21);
10 else
11 Keep V_i at same server until user
terminates it;
12 end
13 else
14 VM threat prediction is not required;
15 end
16 end

(owners of these VMs) producing a complexity of $\mathcal{O}(1)$. The time complexity of step 2 depends on the type of chosen VM placement policy. Steps 3-16 repeat for Y time intervals, wherein the step 4 has complexity of $\mathcal{O}(W)$ [31]. The complexity of step 6 is $T \leftarrow \mathcal{O}(thzlogn)$, where t is the number of trees, h is the height of the trees, and z is the number of non-missing entries in the training data. Prediction for a new sample consumes time $\mathcal{O}(th)$. Therefore, the total timecomplexity of MR-TPM operational algorithm is $\mathcal{O}(YWT)$.

8. PERFORMANCE EVALUATION

A. Experimental Set-up and Implementation

The simulation experiments are executed on a server machine assembled with two Intel[®] Xeon[®] Silver 4114 CPU with 40 core processor and 2.20 GHz clock speed in Cloud data center simulation framework implemented in Python Jupyter Notebook. The computation machine is deployed with 64-bit Ubuntu 16.04 LTS, having main memory of 128 GB. The datacenter environment is set up with three different types of servers and four types of VMs configuration shown in Tables II and III. The resource features like power consumption (P_{max}, P_{min}), MIPS, RAM and memory are taken from real server configuration; IBM [34] and Dell [35], where S_1 is 'ProLiantM110G5XEON3075', S_2 is 'IBMX3250Xeonx3480' and S_3 is 'IBM3550Xeonx5675'. Furthermore, the experimental VMs configurations are inspired from the VM instances of the Amazon website [36].

TABLE II: Server Configuration

Server	PE	MIPS	RAM(GB)	Storage(GB)	PW_{max}	PW_{min}/PW_{idle}
S_1	2	2660	4	160	135	93.7
S_2	4	3067	8	250	113	42.3
S_3	12	3067	16	500	222	58.4

TABLE III: VM Configuration

VM type	PE	MIPS	RAM(GB)	Storage(GB)
v_{type1}	1	500	0.5	40
v_{type2}	2	1000	1	60
v_{type3}	3	1500	2	80
v_{type4}	4	2000	3	100

B. Datasets and Simulation parameters

MR-TPM is evaluated using two benchmark VM traces from a publicly available real workload datasets: OpenNebula Virtual Machine Profiling Dataset (ONeb) [37] and Google Cluster Data (GCD) [38]. ONeb provides information gathered by the monitoring system for six VMs over 63 Hours via executing a set of probe programs provided by OpenNebula. It reports VM threats respective to the server status, basic performance indicators, as well as VM status, and resource capacity consumption of server hosting these VMs. The exact values of VM and hypervisor vulnerability scores are not reported in the original VM threat database. Therefore, to prepare VM threat database including attributes: $\{V_i d_i^{victim}, v_i d_i^{victi$ $S_id, V_id^{Mal}, V_i^{CPU}, V_i^{BW}, V_i^{memory}, R_i^{score}, L_i, H_i, C_i,$ $N_i, V_i^{status}, ..., etc.$, the VMs of ONeb dataset that have experienced attacks, are assigned vulnerability score higher than the threshold value of VM threat (which is considered 0.5 for the experiments) and the rest of the risk scores are computed using Eqs. (13)-(19). These VM threats information is learned by the VM threat predictor for estimation of threats on VMs before occurrence.

Also, we have utilized a realistic workload of Google Cluster Data (GCD)¹, which provides resource usage percentage for each job in every five minutes over period of

¹https://github.com/HiPro-IT/CPU-and-Memory-resource-usage-from-Google-Cluster-Data

twenty-four hours. GCD contains capacity usage information of resources CPU, memory, disk I/O request information of 672,300 jobs executed on 12,500 servers for the period of 29 days. The VM vulnerability (L) and server hypervisor vulnerability (H) are generated in the range [0, 10] during VMs and the server's initialization. Accordingly, the VM threat database reporting traces of attacks on GCD VMs, including attributes { V_{id}^{victim} , S_{id} , V_{id}^{Mal} , V_{i}^{CPU} , $V_i^{BW}, V_i^{memory}, R_i^{score}, L_i, H_i, C_i, N_i, V_i^{status}, \dots, \text{ etc.} \}$ is generated and updated at runtime according to requirement of the proposed model. These datasets do not report user database and hence, we created a user database consisting of $\{U_{id},$ $Attack_{threshold}, U_{class}$ and utilized it for user behaviour analysis based on their previous VM usage. The number of users is considered equals to 30% of the number of VMs (i.e., 1200 VMs), who requested varying number and type of VMs over time. Therefore, different number and types of VMs are mapped with user at run-time and according to the risk scores associated to different VMs, threat is defined. Each user can hold VMs with a constraint that at any instance, the total number of VMs requests must not exceed total number of available VMs at the datacenter. The user database is created and updated during runtime. All the experiments are executed for 100 time-intervals of five minutes to analyse the performance of proposed model dynamically, though this period can be extended as per the availability of traces. The security threats are generated depending upon the threshold values for the four types of risks $\{L_i, H_i, C_i, N_i\}$ associated with a VM and presence of the malicious V^{Mal} . The presence of some malicious user VM (V^{Mal}) on a server and the risk scores corresponding to i^{th} VM (V_i) 'greater than equal to' their respective threshold values indicate the high probability of security threat (i.e., $V_i^{\Xi} > 0$).

C. VM Cyberthreat Prediction

The VM threat prediction is performed for the different population of malicious user U^{Mal} , such as 5%, 20%, 50% and 90%. The prediction accuracy achieved during training, testing, and live phase for 5% and 50% U^{Mal} over period of 500 minutes is shown in Fig. 6 for both the datasets. It can be noticed that prediction accuracy is closer to 98% for all three phases, which is slightly increasing for live cyberthreat detection during each experiment because of the capability of online-learning and re-training of MR-TPM with the passage of time. To provide online training, we perform read/write operation of live VM threats in threat database file dynamically during period of 500 minutes. The Receiver Operator Characteristics (ROC) curve obtained for different experiments using both the datasets are depicted in Fig. 7. ROC curves for the $U^{Mal} = 5\%$ is better than the ROC curves obtained with $U^{Mal} = 50\%$ because of effective learning of true threats in the presence of least number of malicious users. It is observed that the proposed MR-TPM efficiently predicts VM threats for the test as well as live data in all the experiments for both datasets.

Fig. 8 analyses the Actual Threat (AT), Predicted Threat (PT), and Unpredicted Threat (UT) for online VM threat

prediction in the presence of 5%, and 50% U^{Mal} for both datasets. It can be observed that most of the VM threats are predicted correctly where UT is closer to zero and PT is closer to AT, indicating that along with all true threats, some false threats are also predicted. However, the difference between AT and PT is reducing over the time with enhancement of learning capability of VM threat predictor.

The values of precision, recall, F1 measure score, average of mean square error (Avg.MSE), average of mean absolute error (Avg.MAE) observed for the different experimental cases of both the datsets, including GCD and ONeb VM traces are shown in Table IV and Table V which are consistently higher than 0.95 for each case. The Avg.MSE and Avg.MAE values are observed in the range of [0.0001 - 0.0008] and [0.001 - 0.010], respectively, and the accuracy of prediction is higher than 96% reaching up to 99.71% and 99.25% for the GCD and ONeb, respectively. The reason for such an accurate prediction is the incremental learning of MR-TPM with historical and live VM threat databases in real-time. Fig. 9 shows the changes observed in the various risks scores {L, H, N, C} of a randomly selected VM among the 1200 VMs under simulation for a period of 500 minutes.

TABLE IV: Performance metrics for GCD VM traces

U^{Mal}	time	Performance metrics									
(%)	(min.)	Precision	Recall	F1score	Avg.MSE	Avg.MAE	Accuracy				
	100	0.97	0.99	0.98	0.00021	0.0034	98.87				
	200	0.99	0.99	1.00	0.00043	0.0294	98.96				
5	300	1.00	0.99	0.96	0.00036	0.0052	99.42				
	400	0.99	0.97	1.00	0.00071	0.0073	98.87				
	500	1.00	0.99	0.99	0.00191	0.0099	99.71				
	100	1.00	0.98	0.98	0.00062	0.0094	97.07				
	200	0.99	0.99	0.99	0.00031	0.0044	98.66				
20	300	1.00	1.00	1.00	0.00026	0.0025	99.67				
20	400	0.99	0.97	1.00	0.00043	0.0069	99.17				
	500	0.99	0.98	0.99	0.00061	0.0084	99.96				
	100	0.97	0.99	0.98	0.00039	0.0041	98.16				
	200	0.98	0.96	0.99	0.00058	0.0064	97.43				
50	300	1.00	1.00	1.00	0.00016	0.0015	99.69				
	400	0.99	0.98	0.98	0.00037	0.0049	98.25				
	500	0.99	0.99	0.99	0.00031	0.0034	98.17				
	100	1.00	0.99	1.00	0.00028	0.0041	98.76				
	200	0.98	0.96	0.99	0.00058	0.0064	99.74				
5 20 50 90	300	1.00	0.99	1.00	0.00014	0.0020	98.69				
	400	0.99	0.98	0.96	0.00029	0.0039	99.25				
	500	1.00	0.99	0.99	0.00011	0.0014	99.97				

TABLE V: Performance metrics for OpenNebula VM traces

U^{Mal}	time	Performance metrics									
(%)	(min.)	Precision	Recall	F1score	Avg.MSE	Avg.MAE	Accuracy				
	100	0.96	0.98	0.96	0.0007	0.0014	99.10				
	200	0.99	0.99	1.00	0.00023	0.0094	99.06				
U ^{Mat} (%) 5 20 50 90	300	.99	0.99	0.98	0.00045	0.0005	99.10				
	400	1.00	0.97	0.99	0.00091	0.0023	99.07				
	500	0.99	0.98	1.00	0.00071	0.0006	99.11				
	100	0.99	0.97	0.97	0.00062	0.0004	98.16				
	200	1.00	0.99	1.00	0.00051	0.0024	97.96				
20	300	1.00	1.00	0.99	0.00025	0.0015	98.17				
20	400	1.00	0.99	1.00	0.00022	0.0029	99.17				
	500	0.99	0.98	1.00	0.00021	0.0014	99.01				
	100	0.98	0.99	0.99	0.00031	0.0031	98.82				
	200	0.98	0.96	0.99	0.00058	0.0045	98.03				
50	300	1.00	1.00	Performance metrics Recall F1score Avg.MSE Avg.MAE 0.98 0.96 0.0007 0.0014 0.99 1.00 0.00023 0.0094 0.99 0.98 0.00045 0.0005 0.97 0.99 0.00091 0.0023 0.98 1.00 0.00091 0.0023 0.98 1.00 0.00071 0.0006 0.97 0.97 0.00062 0.0004 0.99 1.00 0.00025 0.0014 1.00 0.99 0.00022 0.0024 1.00 0.99 0.00022 0.0024 1.00 0.099 0.00021 0.0014 0.99 0.00 0.00016 0.0021 1.00 1.00 0.00016 0.0021 1.00 0.99 0.00027 0.0030 0.99 0.88 0.00040 0.0032 1.00 1.00 0.00178 0.0042 0.97 0.99 0.00067	99.04						
	400	0.98	1.00	0.99	0.00027	0.0030	99.25				
	500	1.00	0.99	0.98	0.00040	0.0032	98.85				
	100	0.99	1.00	1.00	0.00108	0.0064	97.91				
	200	0.99	0.98	1.00	0.00078	0.0042	98.23				
20 50 90	300	0.99	0.97	0.99	0.00067	0.0038	98.99				
	400	0.99	0.99	0.98	0.00029	0.0029	99.06				
	500	0.99	0.99	1.00	0.00016	0.0016	99.17				

D. Deployment and Comparison

To further analyse the efficiency of MR-TPM, it is deployed with existing state-of-the-art VM placement (VMP) policies, including Previously Selected Server First (PSSF) [11], Secure and Energy Aware Load Balancing (SEA-LB) [16], Security Embedded Dynamic Resource Allocation (SEDRA) [15] and baseline VMP policies, including First-Fit Decreasing (FFD), Best-Fit (BF), and Random- Fit (RF). All the results shown in Section 8-C are derived with FFD VMP policy.

Table VI compares the average number of VM threats realised without and with MR-TPM (results are shown for the Live phase) integrated together with the above mentioned VMP policies. It can be observed that up to 88.5%, 86.5%, 86.2%, 88.9%, 88.5% and 88.1% threats are reduced with proposed approach over PSSF, SEA-LB, SEDRA, RF, BF and FFD, respectively, for $U^{Mal}\% = 90$ at T(min) = 500. The resource utilization of datacenter can be obtained using Eqs. (22), (23), where Z is the number of resources, $\omega_{ji} = \{0, 1\}$ is mapping between server (S_i) and VM (V_j) . Though in formulation, only CPU and Mem are considered, it is extendable to any number of resources.

$$RU_{dc} = \int_{t_1}^{t_2} \left(\frac{RU_{dc}^{CPU} + RU_{dc}^{Mem}}{|Z| \times \sum_{i=1}^{P} \gamma_i} \right) dt$$
(22)

$$RU_{dc}^{r} = \sum_{i=1}^{P} \frac{\sum_{j=1}^{Q} \omega_{ji} \times V_{j}^{r}}{S_{i}^{r}} \quad r \in CPU, Mem, etc.$$
(23)

The resource utilization of different VMP integrated with MR-TPM follows the trend: $FFD \ge SEA - LB \ge SEDRA \ge PSSF \ge BF \ge RF$, as shown in Fig. 10a.

The power consumption for i^{th} server can be formulated as PW_i and total power consumption PW_{dc} during timeinterval $\{t_1, t_2\}$ can be computed by applying Eq. (24), where PW_i^{max} , PW_i^{min} and PW_i^{idle} are maximum, minimum and idle state power consumption of i^{th} server.

$$PW_{dc} = \int_{t_1}^{t_2} \sum_{i=1}^{P} [PW_i^{max} - PW_i^{min}] RU + PW_i^{idle} dt$$
(24)

Fig. 10b shows the comparison of power consumption which is highest (i.e., 109.10 KW) for MR-TPM + PSSF and least (69.29 KW) for MR-TPM + FFD. The average number of active servers is compared in Fig. 10c , where MR-TPM + FFD and MR-TPM + PSSF operates at the lowest (118) and highest (774) number of active servers, respectively. Both the power consumption as well as the number of active servers follow the trend: FFD < BF < SEDRA < SEA - LB <RF < PSSF. The reason for the resultant trend is that VMs are tightly packed onto servers using FFD, while in the case of others, sharing of servers is minimized for the purpose of security at the cost of the larger number of active servers.

9. CONCLUSIONS AND FUTURE WORK

To provide a comprehensive solution for secure workload distribution at cloud datacenter, a novel MR-TPM is proposed which estimates the future threats on user VMs by analysing multiple risk pathways, including VM and hypervisor vulnerabilities, co-residency, network cascading effects and user



Fig. 8: Number of threats (AT:Actual Threats, PT: Predicted Threats, UT: Unpredicted Threats)

TABLE VI: Comparison of number of threats without and with MR-TPM (Live phase) deployed with various VMP approaches

TTMal	T	Percentage of VM security threats (Ξ)											
0						ercentage	Scentage of VIVI security linears (2				EED		
(%)	(min)	PSSF [11]		SEA-LB [10]		SEDKA [15]		KF		БГ		FFI	<u>)</u>
		W-TP	TP	W-TP	TP	W-TP	\mathbb{TP}	W-T₽	\mathbb{TP}	W-T₽	\mathbb{TP}	W-TP	\mathbb{TP}
	100	116	19	206	13	137	17	276	16	283	19	318	58
	200	203	33	187	27	169	19	296	22	258	17	287	22
5	300	270	27	193	19	125	18	226	17	238	16	308	26
	400	216	32	208	18	148	23	298	17	222	18	256	23
	500	223	22	214	21	177	25	196	18	236	17	312	29
	100	365	26	327	54	244	17	474	19	86.7	17.4	678	35
	200	376	23	364	68	314	24	494	17	89.9	15.7	673	42
20	300	399	17	397	39	344	28	478	29	87.5	14.0	579	44
	400	416	31	389	28	297	28	473	26	86.4	17.2	598	58
	500	402	26	428	49	308	37	501	28	89.7	16.9	657	39
	100	537	28	466	37	376	24	638	27	584	27	779	56
	200	556	23	451	39	349	23	627	29	595	24	767	49
50	300	536	15	485	44	339	30	692	27	603	38	797	67
	400	547	37	509	35	388	26	701	28	586	27	745	54
	500	533	41	487	46	373	34	694	25	567	15	779	48
	100	783	57	766	41	676	56	893	65	837	76	958	108
	200	723	49	748	56	621	43	907	84	878	89	997	99
90	300	792	78	687	75	658	49	958	98	889	94	946	83
	400	712	62	673	69	684	57	897	87	847	88	984	94
	500	728	84	678	91	633	87	927	102	837	96	996	118



Fig. 9: Variation of multiple risk values for a VM

behaviour. The proposed model is periodically trained and retrained with historical and live threat data for accurate prediction of threat on VMs. MRTPM deployed with existing VM allocation policies minimizes multiple risks based VM threats and related adversary breaches. The performance evaluation of the proposed VM threat prediction model supports its efficacy in improving cybersecurity and resource efficiency over the compared approaches. In the future, MR-TPM can be extended with transfer learning to enhance its capabilities of analysing unknown/unseen security threats. Additionally, other possible security risk factors can be quantified and included to improve the prediction approach of cyberthreats further.

ACKNOWLEDGMENTS

The authors would like to thank the University of Aizu, Japan and the National Institute of Technology, Kurukshetra, India for financially supporting the research work.

REFERENCES

- S. E. Technology, "Eight top cybersecurity threats for businesses," in Available at: https://www.straightedgetech.com/5-top-cybersecuritythreats-and-their-solutions-for-2020/, 2021.
- [2] C. Point, "2020 cloud security report," in Available at: https://pages.checkpoint.com/2020-cloud-security-report.html, 2020.
- [3] D. Saxena, J. Kumar, A. K. Singh, and S. Schmid, "Performance analysis of machine learning centered workload prediction models for cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 4, pp. 1313–1330, 2023.
- [4] A. K. Singh, D. Saxena, J. Kumar, and V. Gupta, "A quantum approach towards the adaptive prediction of cloud workloads," *IEEE Trans. on Parallel and Distributed Systems*, 2021.
- [5] D. Saxena, A. K. Singh, and R. Buyya, "OP-MLB: An online vm prediction based multi-objective load balancing framework for resource management at cloud datacenter," *IEEE Trans. on Cloud Computing*, 2021.
- [6] D. Saxena and A. K. Singh, "An intelligent traffic entropy learning-based load management model for cloud networks," *IEEE Networking Letters*, vol. 4, no. 2, pp. 59–63, 2022.
- [7] I. Gupta, D. Saxena, A. K. Singh, and C.-N. Lee, "Secom: An outsourced cloud-based secure communication model for advanced privacy preserving data computing and protection," *IEEE Systems Journal*, 2023.
- [8] I. Gupta, R. Gupta, A. K. Singh, and R. Buyya, "Mlpam: A machine learning and probabilistic analysis based model for preserving security and privacy in cloud environment," *IEEE Systems Journal*, pp. 1–12, 2020.
- [9] D. Saxena, A. K. Singh, C.-N. Lee, and R. Buyya, "A sustainable and secure load management model for green cloud data centres," *Scientific Reports*, vol. 13, no. 1, p. 491, 2023.
- [10] T. Y. Win, H. Tianfield, and Q. Mair, "Big data based security analytics for protecting virtualized infrastructures in cloud computing," *IEEE Trans. on Big Data*, vol. 4, no. 1, pp. 11–25, 2017.

- [11] Y. Han, J. Chan, T. Alpcan, and C. Leckie, "Using virtual machine allocation policies to defend against co-resident attacks in cloud computing," *IEEE Trans. on Dependable and Secure Computing*, no. 1, pp. 95–108, 2017.
- [12] D. Saxena and A. K. Singh, "OSC-MC: Online secure communication model for cloud environment," *IEEE Communications Letters*, 2021.
- [13] S. R. Swain, D. Saxena, J. Kumar, A. K. Singh, and C.-N. Lee, "An ai-driven intelligent traffic management model for 6g cloud radio access networks," *IEEE Wireless Communications Letters*, 2023.
- [14] A. K. Singh, S. R. Swain, D. Saxena, and C.-N. Lee, "A bio-inspired virtual machine placement toward sustainable cloud resource management," *IEEE Systems Journal*, 2023.
- [15] D. Saxena and A. Singh, "Security embedded dynamic resource allocation model for cloud data centre," *Electronics Letters*, 2020.
- [16] A. K. Singh and J. Kumar, "Secure and energy aware load balancing framework for cloud data centre networks," *Elec. Lettr.*, vol. 55, no. 9, pp. 540–541, 2019.
- [17] Y. Han, T. Alpcan, J. Chan, C. Leckie, and B. I. Rubinstein, "A game theoretical approach to defend against co-resident attacks in cloud computing: Preventing co-residence using semi-supervised learning," *IEEE Trans. on Information Forensics and Security*, vol. 3, no. 11, pp. 556–570, 2016.
- [18] N. Juma, J. Shahen, K. Bijon, and M. V. Tripunitara, "The overhead from combating side-channels in cloud systems using vm-scheduling," *IEEE Trans. on Dependable and Secure Computing*, 2018.
- [19] X. Liang, X. Gui, A. Jian, and D. Ren, "Mitigating cloud co-resident attacks via grouping-based virtual machine placement strategy," in 2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC). IEEE, 2017, pp. 1–8.
- [20] D. Saxena, I. Gupta, J. Kumar, A. Singh, and W. Xiaoqing, "A secure and multi-objective virtual machine placement framework for cloud data center," *IEEE Systems Journal*, 2021.
- [21] C. Yang, W. Liu, Y. Wang, Q. Tong, and L. Li, "Interference-based vm migration to mitgate cache-based side-channel attacks in cloud," in 2018 IEEE 4th International Conference on Computer and Communications (ICCC). IEEE, 2018, pp. 1188–1192.
- [22] G. Levitin, L. Xing, and Y. Xiang, "Co-residence data theft attacks on nversion programming-based cloud services with task cancelation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 1, pp. 324–333, 2020.
- [23] W. Wu, J. Liu, H. Wang, J. Hao, and M. Xian, "Secure and efficient outsourced k-means clustering using fully homomorphic encryption with ciphertext packing technique," *IEEE Trans. on Know. and Data Engineering*, 2020.
- [24] P. Zhang, M. Zhou, and Y. Kong, "A double-blind anonymous evaluation-based trust model in cloud computing environments," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 3, pp. 1805–1816, 2019.
- [25] F. Miao, L. Wang, and Z. Wu, "A vm placement based approach to proactively mitigate co-resident attacks in cloud," in 2018 IEEE Symposium on Computers and Communications (ISCC). IEEE, 2018, pp. 00 285–00 291.
- [26] X. Qiu, Y. Dai, Y. Xiang, and L. Xing, "A hierarchical correlation model for evaluating reliability, performance, and power consumption of a cloud service," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 3, pp. 401–412, 2015.
- [27] J. Han, W. Zang, S. Chen, and M. Yu, "Reducing security risks of clouds through virtual machine placement," in *IFIP Annual Conference on Data* and Applications Security and Privacy. Springer, 2017, pp. 275–292.
- [28] Y. Zhang, M. Li, K. Bai, M. Yu, and W. Zang, "Incentive compatible moving target defense against vm-colocation attacks in clouds," in *IFIP international information security conference*. Springer, 2012, pp. 388– 399.
- [29] R. C. Chiang, S. Rajasekaran, N. Zhang, and H. H. Huang, "Swiper: Exploiting virtual machine vulnerability in third-party clouds with competition for i/o resources," *IEEE Trans. on Parallel and Distributed Systems*, vol. 26, no. 6, pp. 1732–1742, 2014.
- [30] H. Holm, M. Ekstedt, and D. Andersson, "Empirical analysis of systemlevel vulnerability metrics through actual attacks," *IEEE Trans. on dependable and secure computing*, vol. 9, no. 6, pp. 825–837, 2012.
- [31] D. Saxena and A. K. Singh, "A proactive autoscaling and energy-efficient vm allocation framework using online multi-resource neural network for cloud data center," *Neurocomputing*, 2020.
- [32] D. Saxena, I. Gupta, A. K. Singh, and C.-N. Lee, "A fault tolerant elastic resource management framework towards high availability of cloud services," *IEEE Transactions on Network and Service Management*, 2022.



Fig. 10: Load management metrics



(c) Average number of active servers

- [33] D. Saxena and A. K. Singh, "A high availability management model based on vm significance ranking and resource estimation for cloud applications," *IEEE Trans. on Services Computing*, 2022.
- [34] IBM, "Power model. [online]." https:// www.ibm.com/, 1999
- [35] Dell, "Power model. [online]." https:// www.dell.com/systems/power/hardware/, 1999.
- [36] Amazon, "Amazon ec2 instances. [online]." https:// aws.amazon.com/ec2/instance-types/, 1999.
- [37] P. Purnaye and V. Kulkarni, "Opennebula virtual machine profiling for intrusion detection system," 2020. [Online]. Available: https://dx.doi.org/10.21227/24mb-vt61
- [38] C. Reiss, J. Wilkes, and J. L. Hellerstein, "Google cluster-usage traces: format+ schema," *Google Inc., White Paper*, pp. 1–14, 2011.



Rishabh Gupta received the MCA degree in Computer Science from Guru Jambheshwar University Science and Technology, Hisar, India and Ph.D. from the Department of Computer Applications, National Institute of Technology (NIT), Kurukshetra, India. Currently, he is a Post-doc fellow at The University of Aizu, Japan. He is awarded the Senior Research Fellowship by the University Grants Commission, Government of India. His research interests include cloud computing, machine learning, and information security and privacy.



Deepika Saxena holds the position of an Associate Professor in the Division of Information Systems at the University of Aizu, Japan. Also, she is working as Part-time/Visiting Professor in the University of Economics and Human Sciences, Warsaw, Poland, Europe. She earned her Ph.D. degree in Computer Science from the National Institute of Technology, Kurukshetra, India, and completed her Post Doctorate from the Department of Computer Science at Goethe University, Frankfurt, Germany. She has been honored with the prestigious EUROSIM 2023

Best Ph.D. Thesis Award. Her major research interests include Neural networks, Evolutionary algorithms, Resource management, and Security in Cloud Computing, Internet traffic management, and Quantum machine learning, DataLakes, Dynamic Caching Management. Some of her research findings are published in top cited journals such as IEEE TSC, IEEE TSMC, IEEE TPDS, IEEE TCC, IEEE Communications Letters, IEEE Networking Letters, IEEE Systems Journal, IEEE Wireless Communication Letters, IEEE TNSM, IET Electronics Letters, Neurocomputing, etc.



Ishu Gupta is working as a Ramanujan Faculty Fellow at Computer Science Department, IIIT, Bangalore, India. She completed her Ph.D. from NIT Kurukshetra, India with prestigious UGC-JRF-SRF Fellowships and Postdoc at Cloud Computing Research Center, NSYSU, Kaohsiung, Taiwan. Her major research interests include Cloud Computing, Cybersecurity, Artificial Intelligence, Quantum Machine Learning. She is recipient of Gold Medal for her master's degree and the 'Excellent Paper Award' (Twice).



Ashutosh Kumar Singh is working as a Professor and Director of Indian Institute of Information Technology Bhopal, India. Also, he is working as Adjunct Professor in the University of Economics and Human Sciences, Warsaw, Poland. He received his Ph.D. in Electronics Engineering from Indian Institute of Technology, BHU, India and Post Doc from Department of Computer Science, University of Bristol, UK. He has research and teaching experience in various Universities of the India, UK, and Malaysia. His research area includes Design

and Testing of Digital Circuits, Data Science, Cloud Computing, Machine Learning, Security. He has published more than 370 research papers in different journals and conferences of high repute. Some of his research findings are published in top cited journals such as IEEE TSC, IEEE TC, IEEE TSMC, IEEE TPDS, IEEE TII, IEEE TCC, IEEE Communications Letters, IEEE Networking Letters, IEEE Design & Test, IEEE Systems Journal, IEEE Wireless Communication Letters, IEEE TNSM, IET Electronics Letters, FGCS, Neurocomputing, Information Sciences, Information Processing Letters, etc.



Xiaoqing Wen received the Ph.D. degree from Osaka University, Japan, in 1993. He founded Dependable Integrated Systems Research Center in 2015 and served as its Director until 2017. His research interests include VLSI test, diagnosis, and testable design. He holds 43 U.S. Patents and 14 Japan Patents on VLSI testing. He is a fellow of the IEEE, a senior member of the IPSJ, and a senior member of the IEICE. He is serving as associate editors IEEE Transactions on VLSI and the Journal of Electronic Testing: Theory and Applications.