IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 24 January 2024, accepted 31 March 2024, date of publication 16 April 2024, date of current version 23 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3385863

RESEARCH ARTICLE

Robustness of Workload Forecasting Models in Cloud Data Centers: A White-Box Adversarial Attack Perspective

NOSIN IBNA MAHBUB[1], MD. DELOWAR HOSSAIN[1], SHARMEN AKHTER[1], MD. IMTIAZ HOSSAIN[1], KIMOON JEONG[2], AND EUI-NAM HUH[1], (Member, IEEE)

[1]Department of Computer Science and Engineering, Kyung Hee University, Yongin-si 17104, South Korea
[2]Korea Institute of Science and Technology Information (KISTI)

Corresponding author: Eui-Nam Huh (johnhuh@khu.ac.kr)

This research was partly supported by the Korea Institute of Science and Technology Information (KISTI) (Research of Cloud Based HPC Task Resource Provisioning Prediction Model), the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2024-RS-2023-00258649) supervised by the IITP (Institute for Information Communications Technology Planning Evaluation) and the MSIT (Ministry of Science and ICT), Korea, under the Convergence security core talent training business support program (IITP-2023-RS-2023-00266615) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

ABSTRACT Cloud computing has become the cornerstone of modern technology, propelling industries to unprecedented heights with its remarkable and recent advances. However, the fundamental challenge for cloud service providers is real-time workload prediction and management for optimal resource allocation. Cloud workloads are characterized by their heterogeneous, unpredictable, and fluctuating nature, making this task even more challenging. As a result of the remarkable achievements of deep learning (DL) algorithms across diverse fields, scholars have begun to embrace this approach to addressing such challenges. It has become the defacto standard for cloud workload prediction. Unfortunately, DL algorithms have been widely recognized for their vulnerability to adversarial examples, which poses a significant challenge to DL-based forecasting models. In this study, we utilize established white-box adversarial attack generation methods from the field of computer vision to construct adversarial cloud workload examples for four cutting-edge deep learning regression models, including Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), 1D Convolutional Neural Network (1D-CNN) and attention-based models. We evaluate our study with three widely recognized cloud benchmark datasets: Google trace, Alibaba trace, and Bitbrain. The findings of our analysis unequivocally indicate that DL-based cloud workload forecasting models are highly vulnerable to adversarial attacks. To the best of our knowledge, we are the first to conduct systematic research exploring the vulnerability of DL-based models for workload forecasting in the cloud data center, highlighting the inherent hazards to both security and cost-effectiveness in cloud data centers. By raising awareness of these vulnerabilities, we advocate the urgent development of robust defensive mechanisms to enhance the security of cloud workload forecasting in a constantly evolving technical landscape.

INDEX TERMS Cloud computing, workload prediction, cloud security, deep learning, adversarial attack.

I. INTRODUCTION

The cloud computing [1], [2] offers the potential for users to access computing, storage, and networking resources as needed, accompanied by service level agreements (SLAs) established among providers of cloud services (CSPs) and users. The key objective of the cloud computing strategy is to facilitate the provisioning of resources on-demand, ensuring economic satisfaction for both cloud providers and users [3]. Simultaneous user requests can lead to burst workloads, potentially causing insufficient availability of resources. Conversely, the idle status is characterized by low workloads,

The associate editor coordinating the review of this manuscript and approving it for publication was Frederico Guimarães.

leading to the inefficient utilization of resources. Workload fluctuations result in resource over-provisioning or under-provisioning, leading to high overhead costs or inadequate SLAs [4], [5]. Therefore, it is essential for CSPs to efficiently ascertain resource allocation strategies that ensure SLAs and enhance resource utilization [4]. To attain these goals, cloud computing requires rapid and adaptable workload forecast systems [6]. Proactive resource configuration and allocation strategies enable efficient resource provisioning through precise workload prediction. In recent decades, multiple methods have been developed for predicting cloud workloads. The schemes discussed in this context can be categorized into three primary groups: statistical models, machine learning techniques, and deep learning-based approaches [7]. Analytical forecast approaches assume linear dependence and stationary behavior among time-series samples. Standard statistical techniques like Holt-winter [8], ARIMA (Autoregressive integrated moving average) [9], seasonal ARIMA (SARIMA) [10], and Markov models [11], [12] are widely used. However, these methods have shown limited success in accurately forecasting excessively unstable time series and long-term applications [13]. In contrast, another classification of machine learning methodologies has been utilized to overcome the limitations of traditional approaches [13]. Several approaches, including particle swarm-optimization (PSO), support-vector-regression (SVR), and relevance-vector-machine (RVM), have been employed for predicting the workload in cloud data centers [13]. Although these methodologies are not suitable for managing large datasets, their effectiveness greatly depends on fine-tuning parameters [14]. DL-based forecasting algorithms have recently gained popularity for their superior performance compared to classical machine learning techniques in solving complex problems, including cloud workload forecasting [15], [16]. With the rising popularity of cloud services such as location-based services (LBS) [17], e-health [18] and others, which are managed through containers or virtual machines across multiple clouds, there is an increasing demand to develop intelligent workload prediction models. These models should be trained using trace data that covers various cloud environments. Since the practice of transferring data to cloud servers increases to reduce the burden of local storage and computation, it also intensifies worries about security [19], [20]. The dilemma of data privacy, driven by concerns about sharing trace data between clouds, prevents the use of traditional distributed training methods. To address this difficulty, researchers have lately turned to federated learning [21] as a feasible approach for anticipating workloads in inter-cloud environments [22]. Notably, federated learning algorithms are based on the assumption of trust among users participating in collaborative model training. However, practical circumstances contradict this assumption, as participants in federated learning systems frequently struggle with a lack of mutual trust [23]. This trust deficit results from possible external threats or resource limits, which contribute to the dependability of individual participant activity.

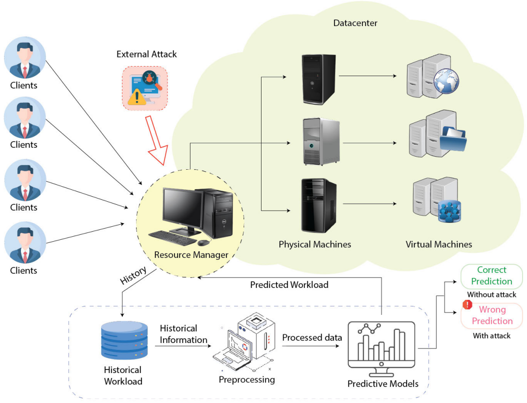


FIGURE 1. Overview of an external attack on workload predictive models in a cloud datacenter.

Additionally, cloud environments are an essential framework that hackers can target, as the data maintained in the cloud is typically sensitive and confidential. External attacks on forecasting frameworks can distract security teams from focusing on actual threats [24]. When data center operators are faced with the consequences of unreliable workload predictions, they might overlook other, potentially more serious security incidents or vulnerabilities that are being exploited at the same time. Figure 1 illustrates the impact of an external attack on predictive models in a cloud datacenter.

Therefore, it is essential to consider not only how accurate cloud workload forecasts are but also how secure they are against attack. Current studies have indicated that DL algorithms are prone to vulnerability when subjected to adversarial attacks [25], [26]. Thus, using DL involves risks and provides attackers with new attack possibilities. Adversarial attacks subtly alter the initial data of Machine Learning (ML) methods in order to produce false forecasts. This particular hazard poses a significant risk to DL models that receive input data from interfaces that are crucial for ensuring safety. Different cloud trace data are frequently used as input features in cloud workload forecasting models. Such information is commonly obtained from publicly accessible data sources, which might be tampered by hackers. Hence, the primary concern of this study lies in the topic of adversarial attacks on cloud workload prediction and it aims to answer the following inquiries: Can adversarial examples be used to attack state-of-the-art DL models for cloud workload prediction? The significant contributions of this study are as follows:

- **Adversarial workload construction:** We introduce two well-established white-box adversarial attack methods named FGSM and PGD from the field of computer vision which can effectively construct adversarial cloud workload samples.

- **Reliability assessment framework:** We propose a novel framework to evaluate the robustness of the state-of-the-art deep learning-based cloud workload prediction models. This assessment is conducted through white-box adversarial attacks to uncover the vulnerabilities of cloud workload forecasting models.
- **Transferability analysis:** We investigate an in-depth analysis of the transferability properties inherent in adversarial samples within deep learning-based workload prediction models. This investigation sheds light on potential cross-model vulnerabilities.
- **Comprehensive experiments:** We conduct extensive experiments with the most well-known real-world cloud workload datasets from Bitbrain, Google trace, and Alibaba trace. The results indicate the effectiveness of our proposed white-box adversarial attack strategy in challenging the robustness of existing workload forecasting methods.
- **Defense Mechanism Discussion:** We carry out a comprehensive discussion about potential defense strategies, which are crucial for future research in this domain. This will contribute to developing more robust cloud workload prediction techniques that are resistant to adversarial attacks.

The remaining parts of this study are structured in the following manner. Section II provides a concise overview of the existing literature. Section III and IV provide the technical background and methodology of this research, respectively. Section V of the research paper presents the experimental setup employed in the study, as well as a thorough analysis of the obtained results. Sections VI and VII of the work present a brief discussion of the potential defense mechanism and conclusion, respectively.

II. RELATED WORK

A. DEEP LEARNING FOR WORKLOAD FORECASTING

In light of the recent achievements of deep learning (DL) in diverse domains, numerous studies have utilized DL methodologies for analyzing and predicting time-series data. Notably, the recurrent neural network (RNNs) offers exceptional capacities for sequential processing. Consequently, the authors of [27], [28], and [29] employed the RNN-based approach to forecast the workload in a cloud environment. However, prior studies have indicated that vanilla RNNs encounter difficulties in capturing long-term dependencies as a result of the vanishing gradient problem [30]. In order to alleviate this difficulty, LSTM [31] and GRU [32] were formed to address long-term dependencies more effectively. Therefore, the authors of [33] used the LSTM network to predict workload, which was an improvement over their RNN-based work. Compared to LSTM, GRU requires significantly less processing power due to its capacity to converge with fewer parameters [32]. Hence, a few studies [34], [35] utilized GRU-based architecture to predict cloud workload. In contemporary research, the reliability of cloud workload forecasting has been substantially improved by the implementation of more advanced deep learning methods. Authors of [36] demonstrate the effectiveness of the BHyPreC architecture, which is a Hybrid Recurrent Neural Network (RNN) that utilizes stacked LSTM and GRU components to predict CPU consumption workloads for cloud virtual machines via Bidirectional Long Short-Term Memory (Bi-LSTM). Hybrid generative adversarial networks emerge victorious in [37], demonstrating remarkable precision in forecasting forthcoming duties and differentiating patterns. In comparison to extant models, authors in [38] and [39], introduce an ensemble architecture incorporating Attention Mechanisms (AM), LSTM, Bidirectional Long Short-Term Memory (BiLSTM), and Convolutional Neural Networks (CNN), resulting in substantial reductions in RMSE and MAE compared to existing models. The MAG-D model, proposed in the article [7], utilizes Multivariate Attention and Gated Recurrent Units to effectively capture long-range dependencies, resulting in superior performance compared to existing techniques. In addition, a Convolutional Neural Network (CNN) effectively retrieves spatial features and incorporates them seamlessly into a Gated Recurrent Unit (GRU) network that has been optimized for temporal correlation with an attention mechanism [40].

B. ADVERSARIAL ATTACK

The majority of adversarial attack strategies were initially developed for image classification in the deep learning domain. Szegedy et al. [25] introduced adversarial examples for image recognition, paving the way to study adversarial attacks across multiple disciplines. The Fast Gradient Sign Method (FGSM) was proposed by Goodfellow et al. [26] as a single-step attack. The initial adversarial attacks on Time Series Classification (TSC) were introduced by Oregi et al. [41]. Fawaz et al. [42] employed established adversarial attack methods, including the Fast Gradient Sign Method (FGSM) and Basic Iterative Method (BIM), to reduce the performance of residual networks in terms of Time Series Classification. Rathore et al. [43] introduce the concept of targeted attacks on time series data. The targeted attacks primarily focus on TSC tasks. Yang et al. [44] introduce a black-box technique referred to TSadv for the TSC task. Additionally, Mode and Hoque et al. [45] investigate the susceptibility of deep learning multi-time series regression methods to adversarial samples in the context of time series forecasting. Additionally, the work concentrates on gradient-based white box attacks on several deep learning methods (including 1D-CNNs, GRUs, and LSTMs).

III. TECHNICAL BACKGROUND

A. PROBLEM DEFINITION

1) CLOUD WORKLOAD FORECASTING

An univariate time series refers to a collection of measurements of a single variable that are recorded over a period of time. We analyze univariate time-series data of workload

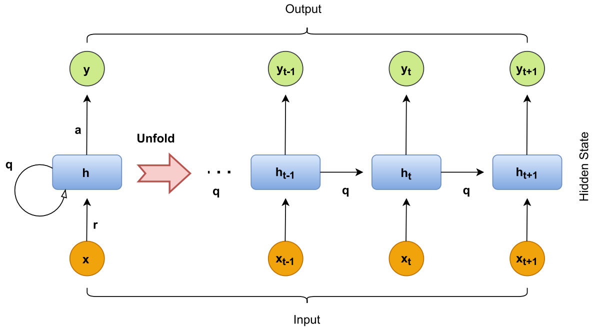


FIGURE 2. Recurrent neural network architecture.

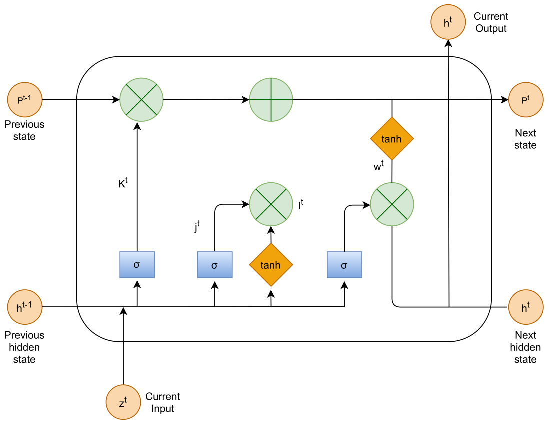


FIGURE 3. Long short term memory architecture.

(CPU usages) obtained at regular intervals from real-world cloud trace datasets.

Given a workload $W = [w_1, w_2, ..w_T]$, a workload forecasting task predicts the value of $w_T + 1$ based on the previous samples $[w_T - L, w_T - L + 1, \ldots w_T]$, where L is the lookback period under consideration. The sample $wT + 1$ corresponds to the forecasted value and is often represented by $\hat{Y}$.

2) ADVERSARIAL WORKLOAD

An adversarial perturbation ϵ, typically superposed on a given workload W, to construct $\hat{W}$ given by $[\hat{w_1}, \hat{w_2}, \ldots, \hat{w_T}]$. The adversarial workload $\hat{W}(W_{ad})$ is intended to significantly worsen the output prediction $\hat{Y}$ of a workload forecasting model.

3) GOAL OF ADVERSARY

The goal of the attacker is to create a targeted output impact on the time series. We consider L∞-bounded perturbation that causes a targeted attack. The definition of white-box access is examined in this context, which refers to the availability of the model's gradients for loss computation. We denote the regressor $f : \mathbb{R}^{(M)} \rightarrow \mathbb{R}^{(N)}$ with parameters θ, the predicted output for input $x \in \mathbb{R}^{(M)}$ is represented as $y = f(x)$.

4) PROPERTIES OF THE PERTURBATION

Usually, the introduction of a perturbation to the input tends to impair the performance of the model's prediction. Additionally, it is also important for the perturbation to satisfy additional requirements including: 1. Small changes to the input to create bigger performance degradation on the output. Larger perturbations to the inputs are also easily detectable by the input plausibility check modules. Notably, it is more expensive to achieve higher input perturbation. 2. Imperceptible perturbations, attributing to reduced risk due to detection of the input perturbation. The perturbation is hence formulated as a constrained optimization problem, where F is the workload forecasting model under consideration and ϵ indicates the strength of the attack.

B. WORKLOAD FORECASTING MODELS

1) RECURRENT NEURAL NETWORK

Recurrent Neural Networks (RNNs) belong to a category of Artificial Neural Network (ANN) algorithms specifically designed to process sequential data, particularly time-series data. An RNN's network architecture uses a feedback loop to handle variable-length input sequences. It uses the output from the previous step (n-1) as input for the current step n, performing an iterative procedure for each successive step. This network architecture is useful for predicting cloud workloads by predicting past load levels to anticipate future workloads [36]. The RNN model, depicted in Figure 2, has a single hidden layer and an extended structure, with input data x_t, hidden state h_t, and output y_t. Historical data on cloud workload, specifically CPU consumption (x_t), will be used for forecasting. According to Figure 2, the calculation of h_t in the RNN is based on the previously hidden state values and the output at the current time step and can be determined by the following equation:

$$h_t = \beta_h(rx_t + qh_t + e_h) \tag{1}$$

The computation of the output state y_t for the input x_t is contingent upon the hidden state h_t at time step t in the following manner:

$$y_t = \beta_y(ah_t + e_y) \tag{2}$$

Here, β_h and β_y represent non-linear activation functions, such as tanh, ReLU, or sigmoid functions. Once again, the parameter matrices and vectors are denoted as r, a, q, and e. An RNN employs a consistent set of parameters, matrices, and vectors across all stages, reducing the number of parameters required for training, unlike a conventional deep neural network.

2) LONG SHORT-TERM MEMORY

Long Short-Term Memory (LSTM) models are commonly favored for addressing the challenge of long-range reliance and have demonstrated exceptional performance in tasks that involve sequences. The incorporation of a hidden layer within the LSTM structure distinguishes it from the RNN

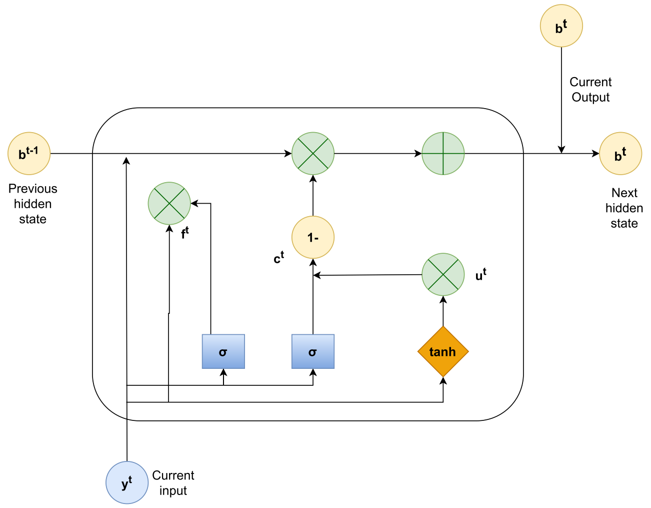


FIGURE 4. **Gated recurrent unit architecture.**

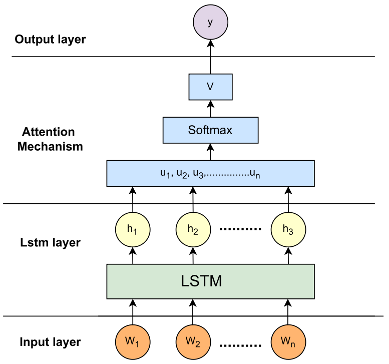


FIGURE 5. **Architecture of the LSTM network incorporating the attention mechanism.**

design. The concealed stratum inside the LSTM model is commonly referred to as the LSTM cell [7]. The architectural representation of the basic LSTM block is illustrated in Figure 3. The LSTM architecture consists of a memory cell, input gate, output gate, and forget gate. The activation functions used are the Sigmoid function and hyperbolic tangent function. The variable x^t represents the input at a specific time, while h^t represents the concealed state. The process of quantifying input data in the cell state involves using K^t as the candidate state. Recurrent and input weights are represented as Ω and α, respectively. The bias associated with the forget gate is denoted by o^K. The sigmoid function determines the permissible range of values that can be transmitted. The forget gate k_t verifies the contents of the cell state and determines what needs removal. Additionally, it can be presented as:

$$k^t = \sigma(x^t \times \alpha^k + o^k + h_{t-1} \times \Omega^k) \tag{3}$$

The subsequent step involves updating the cell state p^t by incorporating the newly acquired information from the preceding cell state p^{t-1}, as well as the input and forget gates.

$$p^t = p^{t-1} \times k^t + l^t \times j^t \tag{4}$$

The tanh activation function is commonly chosen to generate the new candidate values l^t in the construction process. The formulation for this activation function is as follows:

$$l^t = tanh(o^l + x^t \times \alpha^l + h^{t-1} \times \Omega^l) \tag{5}$$
$$j^t = \sigma(o^j + \alpha^j \times x^{t-1} + \Omega^j \times h^{t-1}) \tag{6}$$

The weight matrices, bias vector, and output gate W^t are represented by $(\Omega^l, \alpha^l, o^l)$ and $(\Omega^j, \alpha^j, o^j)$, respectively. The output gate W^t is determined by the cell state content, with the sigmoid activation function identifying the fragment portion to output. The formulation of this concept can be expressed as:

$$w^t = \sigma(o^w + x^t \times \Omega^w + h^{t-1} \times \Omega^w) \tag{7}$$
$$h^t = w^t \times tanh(p^t) \tag{8}$$

The weight metrics assigned to the recurrent and input components are symbolized as (Ω^w, α^w), while the bias of the output gate is represented as oG in the equation. In the provided diagram, the symbol ⊗ denotes the operation of element-wise multiplication.

3) GATED RECURRENT UNIT

The Gated Recurrent Unit (GRU) design, which is recognized as a viable alternative to LSTM, is characterized by its simpler structure and widespread popularity [32]. The update gate in the GRU architecture combines the functions of the forget gate and input gate into a unified element. This update gate operates using a single hidden state. The basic framework of a conventional GRU cell is seen in Figure 4.The input weight matrix, recurrent weight matrix, and bias are represented by P^i, Z^i, and L^i at a specific time-step t. The reset gate, S_t, integrates previous memory and new input. After the function f^t is terminated, associated data becomes inapplicable for the current hidden state, and is disregarded.

$$S^t = \sigma(Z^t b^{t-1} + P^t y^t + L^t) \tag{9}$$

The formulation of the candidate cell v_t is based on the input weight matrix P_n, recurrent weight matrix Z_n, as well as bias L_n. It takes into account the input y_t, the previous hidden state b^{t-1}, and the reset gate S_t.

$$v^t = tanh(Z^n((S^t \oplus b^{t-1} + P^n y^t + L^n))) \tag{10}$$

The gate k^t is updated using the weight matrices (Z^m, P^m) for recurrent and input connections and the bias term L^m. The function of this gate is to regulate the transmission of information from the preceding concealed state to the present concealed state.

$$k^t = \sigma(Z^m b^{t-1} + P^m y^t + L^m) \tag{11}$$

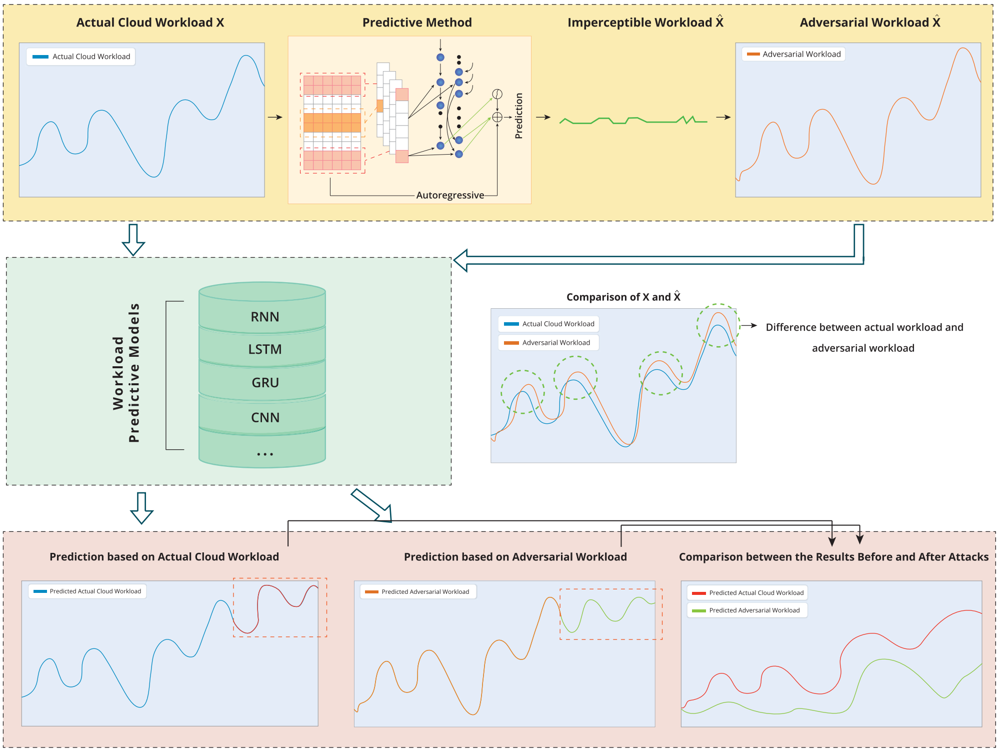

Actual Cloud Workload X
Actual Cloud Workload
Predictive Method
Prediction
Autoregressive
Imperceptible Workload X̂
Adversarial Workload X̂
Adversarial Workload
Workload Predictive Models
RNN
LSTM
GRU
CNN
...
Comparison of X and X̂
Actual Cloud Workload
Adversarial Workload
Difference between actual workload and adversarial workload
Prediction based on Actual Cloud Workload
Predicted Actual Cloud Workload
Prediction based on Adversarial Workload
Predicted Adversarial Workload
Comparison between the Results Before and After Attacks
Predicted Actual Cloud Workload
Predicted Adversarial Workload

FIGURE 6. Overall workflow of this study where the upper section depicts the input of the actual cloud workload into predictive models to generate perturbation, and the lower section illustrates how implementing adversarial attacks disrupts the accurate prediction of future values.

The final hidden state b^t can be computed by adding two composite equations. These equations involve the element-wise product $\otimes$ of $(1 - k^t)$, b^{t-1}, and k^t, resulting in v^t.

$$b^t = (1 - k^t) \otimes b^{t-1} + k^t \otimes v^t \tag{12}$$

The utilization of smaller gates in combination with the GRU results in a more streamlined and effective architecture that requires fewer parameters. This leads to accelerated training times as compared to the LSTM network.

4) 1D CONVOLUTIONAL NEURAL NETWORK

Convolutional Neural Networks (CNNs) are powerful tools for feature extraction and deep mining of information. The two-dimensional nature of image data makes it ideal for processing image data. However, most data used to forecast roller bearings' useful life consists of uni-dimensional vibration signals, which cannot be efficiently handled using a two-dimensional convolution kernel. A 1D-CNN approach is used to extract features from these signals, considering specific sequence data attributes [46]. 1D-CNNs offer a unique method for feature extraction, allowing for autonomous retrieval of data features without requiring extensive operator input or specialized expertise. The horizontal vibration signal undergoes fast Fourier transformation to reduce the sequence length and improve data representation. The positive segment, 0 Hz points, and central symmetry points are selected to minimize computational requirements. The sequence data is inputted into the 1D-CNN model, and feature data is retrieved through iterative movement of the one-dimensional convolution kernel. By analyzing data variations across multiple time points, specific characteristics can be extracted for prediction. The spectral characteristics of bearings vary over their life cycles, with notable differences observed in the middle and later stages of degeneration. Thus, 1D-CNN models enable the extraction of diverse feature information from sequential input at distinct temporal intervals for prediction.

5) ATTENTION MECHANISM

The attention mechanism, introduced in [47], is proposed as a way to comprehensively analyze all input words in a natural language processing (NLP) issue and assign relative priority to each word. Due to the similarities between NLP and time series forecasting, the attention mechanism also attracted the eye of the time series community [48]. The majority of attention models utilize the Encoder-Decoder framework, which initiates by analyzing the input sequence for a set of encoder-generated annotations. The concealed state of the decoder is determined by an operator-defined neural network recurrent architecture (Regarding this study, LSTM blocks are utilized). Figure 5 depicts the structure of the LSTM network integrated with the attention mechanism. The attention mechanism is executed primarily through the subsequent procedures. The LSTM produces the output $[h_1, h_2, h_3, .., h_n]$, which is nonlinearly transformed into $[u_1, u_2, u_3, \ldots, u_n]$. Certain components of the cloud workload forecasting method have a significant impact on the workload prediction; therefore, greater weight needs to be assigned to these portions. The attention weight matrix $[\alpha_1, \alpha_2, \alpha_3, \ldots., \alpha_n]$, which can represent the significance of each intermediate state, is generated by the attention mechanism. As a final step, the input parameter and weight are combined via weighted sum to produce the encoding vector V. Decoding the input y by the encoding vector V yields the desired result y. The following is the detailed equation of the attention mechanism:

$$U_j = tanh(W_j h_j + c_j), \tag{13}$$

$$\alpha_j = \frac{exp(U_{j^T} U_m)}{\sum_{j=1}^{n} exp(U_{j^T} U_m)}, \tag{14}$$

$$V = \sum_{j=1}^{n} \alpha_j h_j, \tag{15}$$

where W_j is the weight matrix, c_j is the value of the offset number, α_j is the attention weight that has been normalized, and U_m is the attention time series matrix that has been randomly initialized.

C. GRADIENT BASED WHITE BOX ADVERSARIAL ATTACK FOR CLOUD WORKLOAD FORECASTING

In white box adversarial attacks, the attacker possesses complete access to the targeted system. This includes comprehensive knowledge of the model, encompassing its gradient, parameters, hyperparameters, as well as the training dataset [49]. According to [50], the attack algorithms that yield the highest success rates primarily utilize optimization techniques based on gradients. The researchers extract a substantial volume of data from the method through the utilization of adversarial assaults generated by the gradients of a loss function. The research literature frequently uses two optimization-based attacks: the Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD). These attacks were initially introduced in [26] and [51] respectively.

1) FAST GRADIENT SIGN METHOD(FGSM)

The Fast Gradient Sign Method (FGSM) was initially introduced in [26], demonstrating its effectiveness in deceiving the GoogLeNet model through the creation of inconspicuous hostile images. FGSM computes the gradient of the cost function concerning the input of the neural network. This form of attack is sometimes referred to as the one-shot approach, as the generation of the adversarial perturbations occurs through a single-step computation. It should be noted that the FGSM is an approximation explanation that relies on a linear premise [52]. Adversarial instances for cloud workload are generated using the subsequent mathematical expression:

$$\begin{aligned} \eta &= \epsilon \cdot sign(\nabla_w Z_f(W, Y)) \\ \hat{W} &= W + \eta \end{aligned} \tag{16}$$

Here, Z_f represents the cost function associated with model f, while ∇ denotes the gradient of the model about the initial cloud workload. In the context of this study, the variable W is appropriately assigned the label Y. The symbol α represents the hyper-parameter that dictates the magnitude of the perturbation, while $\hat{W}$ refers to the adversarial workload.

Algorithm 1 FGSM Attack on Cloud Workload

```
Require: Actual workload $W$; and it's label $Y$; defined model parameters w; loss function $Z$ of the recognition model f; perturbation intensity $\epsilon$
Ensure: Adversarial workload $\hat{W}$
1: Get the loss $Z(W, Y)$ after forward propagation.
2: Get the gradient $\nabla_w Z_f(W, Y)$ of the input workload.
3: Take the sign function for the obtained gradient and get sign($\nabla_w Z_f(W, Y)$).
4: Calculate the adversarial perturbation: $\eta = \epsilon \cdot$ sign($\nabla_w Z_f(W, Y)$).
5: Add the adversarial perturbation to the input workload and obtain the adversarial workload as follows:
       $\hat{W} = W + \eta$
         $= W + \epsilon \cdot \mathrm{sign}(\nabla_w Z_f(W, Y))$
6: return $\hat{W}$
```

2) PROJECTED GRADIENT DESCENT (PGD)

The PGD attack is a variant of the Iterative-FGSM attack, which is commonly referred to as the Basic Iterative Method (BIM) [51]. The initialization of PGD can be performed by randomly selecting any point within the l_∞-norm distance of the original sample, as mentioned in the work of Madry et al. [53]. In contrast to the single-step approach known as the FGSM, the PGD technique involves the execution of many iterations. With each incremental action, the disturbance will be consistently confined within the predetermined range.

$$W_+ t + 1 = \prod_{w+P} (W_t + \alpha \cdot sign(\nabla_w Z_f(W_t, Y, \theta))) \tag{17}$$

Here, α be the length of each step, and $P = r \in R^d$ represents the perturbation set. Furthermore, the perturbation r meets the condition that its infinity norm is less than or equal to epsilon, denoted as $\|r\|_\infty < \epsilon$. On the other hand, the expression $\prod_{w+P}$ denotes the projection onto the ϵ-neighborhood range sphere. If the intensity of the disturbance is excessively large, the surplus portion will be drawn back towards the boundary region.

IV. METHODOLOGY

The objective of adversarial attacks in cloud workload forecasting is to create carefully constructed hostile workload instances to deceive prediction systems. The prediction model demonstrates a high degree of accuracy in forecasting

Algorithm 2 PGD Attack on Cloud Workload

```
Require: Input workload $W$; actual label $Y$; defined model
    parameters $\theta$; loss function $Z$ of the defined model;
    perturbation intensity $\epsilon$; the total number of iterations $T$
Ensure: Adversarial Workload $\hat{W}$
Get the perturbation level $\alpha = \epsilon / T$ of each iteration;
Initialize $\hat{W}_0 = W$;
for epoch $t = 0$ to $T - 1$ do
    Get the loss $Z(W, Y)$ after forward propagation;
    Get the gradient $\nabla_w Z_\theta(W, Y)$ of the input workload;
    Take the sign function for the obtained gradient and
    get $\mathrm{sign}(\nabla_w Z_\theta(W, Y))$;
    Calculate the adversarial perturbation of each iteration
    and get $\alpha \cdot \mathrm{sign}(\nabla_w Z_\theta(W, Y))$;
    Use $\mathrm{Proj}\{\cdot\}$ to project the adversarial workload in
    the $\alpha - l_\infty$ neighbor of the original workload
    after each iteration as: $\hat{W}_{t+1} = \mathrm{Proj}_{t,\alpha}\big(\hat{W}_t + \alpha \cdot \mathrm{sign}(\nabla_w Z_\theta(\hat{W}_t, y))\big)$;
end for
Return $\hat{W} = \hat{W}_{T-1}$
```

future workload direction, particularly when considering the workload patterns observed over a certain time frame. Nevertheless, the utilization of an attack-based methodology can produce adversarial perturbations that can effectively disrupt the initial workload and mislead the prediction techniques. It is desirable for the adversarial task to closely resemble the original workload. This work aims to reduce the distance between the two sequences by controlling the perturbations. The comprehensive structure of this investigation is illustrated in Figure 6. This study encompasses the three primary elements of an adversarial attack: an adversarial cloud workload generator, an adversarial attack, and a transferrable attack. The intricacies of the framework are elaborated upon in the subsequent sections. A brief discussion of the methodology is presented below.

- **Adversarial workload generator.** The key aim of an adversarial cloud workload generator is to produce subtle yet impactful perturbations in the initial workload that may effectively deceive the workload prediction model while remaining undetectable.

  As illustrated in the upper portion of Figure 6, the core idea is to obtain the actual cloud workload W and feed it to workload forecasting models to construct the perturbation $W + \eta$, leveraging the adversarial example generator (Algorithms 1 or 2, named FGSM and PGD, respectively). In FGSM, as demonstrated in Algorithm 1, the process starts by evaluating the loss experienced by the model with the real workload. It calculates the gradient of the loss with respect to the input workload, determining the direction with the highest increase in loss. Through analyzing the sign of the gradient (positive or negative), the algorithm crafts the adversarial perturbation. Finally, this perturbation is applied to the input workload to generate an adversarial workload. In contrast to the single-step perturbation employed by FGSM, PGD adopts an iterative technique to refine the adversarial workload, as shown in Algorithm 2. The approach calculates the loss and gradient for the current workload, projects the resultant perturbation into an appropriate range, and uses it to produce the workload for the next iteration. This repeated refining process helps to generate a more robust adversarial workload, which increases the potential efficacy of the adversarial assault. Such repeated modification is consistent with the larger goal of improving adversarial resilience in the target model.

  Considering the premise that the attacker possesses knowledge of the loss function employed in the workload prediction method, it becomes feasible for the attacker to acquire gradient information by performing partial derivatives on the loss value.
- **Workload forecasting adversarial attack.** The bottom section of Figure 6 demonstrates that the implementation of adversarial attacks by the adversarial workload hinders the correct prediction of future values. The primary aim of an adversarial attack is to significantly compromise the accuracy of workload prediction algorithms by exploiting the high cost associated with data tampering and the ability of adversarial perturbations to go undetected.
- **Transferable attack.** Numerous modern deep neural networks have been utilized in addressing the challenge of workload prediction. A "transferable adversarial attack" refers to a scenario where adversarial examples crafted to deceive a certain prediction model are effective in fooling that model, while also causing other prediction approaches to fail. In this study, we investigate the potential for portable adversarial assaults and carry out extensive tests to confirm this hypothesis.

V. EXPERIMENTS

This section provides a thorough analysis to examine the robustness of five contemporary deep learning approaches in the context of workload forecasting. The evaluation of these models is conducted using two popular evaluation metrics and three real-world cloud trace datasets.

A. EXPERIMENTS SETUP

1) DATASETS

This research employs three freely accessible cloud trace datasets to conduct a performance evaluation. During the training and testing of forecasting models, the dataset sequence is partitioned into two sub-datasets. The initial subset of the dataset contains 70% of the total data and is employed to train the model. The remaining 30% of the data is allocated to evaluate the performance of the model. The

TABLE 1. Summary of Bitbrain dataset.

Metrics	Description
CPU cores	Number of virtual CPU cores
CPU capacity provisioned	The capacity of the CPUs
CPU usage	CPU usages in MHZ
CPU usage	CPU usages in percentage
Memory provisioned	The capacity of VM memory
Memory usage	The memory that is actively used
Disk read throughput	Data read speed
Disk write throughput	Data write speed
Network received throughput	Data received speed by the network
Network transmitted throughput	Data transmitted speed by the network

workload data was partitioned into multiple time windows of size 30 to predict future workload.

a: BITBRAIN

Bitbrain is a widely recognized distributed data center that specializes in the collection of extensive and enduring traces of authentic data [54]. The dataset comprises performance metrics for 1,750 virtual machines obtained from Bitbrains' distributed data center. Bitbrains is a corporate entity that focuses on offering regulated hosting and business computing services tailored to meet the needs of various enterprises. The management of computing capacity is facilitated through the utilization of generic VMware provisioning frameworks, such as Dynamic Resource Scheduling and Storage Dynamic Resource Scheduling. Each file contains the performance metrics of the virtual machine. The files are organized into two categories: fastStorage and Rnd. FastStorage is a system including 1250 virtual machines (VMs) that are associated with storage devices known as Storage Area Network (SAN) gadgets. On the other hand, Rnd is a system consisting of 500 VMs that are associated with either faster SAN gadgets or somewhat slower Network Attached Storage (NAS) gadgets. The arrangement of each file follows a row-based structure, where every row represents an analysis of performance metrics consisting of 11 columns. The dataset encompasses a total of 5,446,811 CPU hours, 23,214 GB of memory, and 5,501 cores. Following the pre-processing steps, the original dataset undergoes a conversion procedure resulting in the formation of a DataFrame. Table 1 presents a concise overview of the Bitbrain trace dataset. In this study, we utilize the 'CPU Usage' data from the provided DataFrame.

b: GOOGLE TRACE

Google collected and stored trace data during the entirety of May 2019 [55]. This trace offers a comprehensive understanding of the actual cloud data center infrastructure. Typically, the workload is received by the cluster in the form of jobs. The dataset contains running-time traces for over 650,000 real-time jobs that have undergone various scheduling methods. These traces include the start time, end time, and execution time of the jobs over 29 days. The employment positions are allocated on disparate physical computers that possess varying quantities of cores and RAM. Every job within the cluster trace is linked to a collection of resource use metrics that have been gathered at various intervals. The resource tables for all the traces are categorized into three distinct categories: Jobs & Tasks, Machines, and Resource Utilisation. For this study, we considered the initial seven-day period CPU utilization data from the resource utilization table. A concise overview of the metrics about resource utilization is presented in Table 2.

TABLE 2. Summary of resource utilization in Google trace dataset.

Metrics	Description
ACPU	Aggregated CPU usages
AMEM	Aggregate memory Usages
MMEM	Maximum memory usages
AVM	Allocated virtual memory
MCPU	Maximum CPU usages
CPC	Combined Page cache usages
CPI	Cycles per instruction across all nodes
TPUC	Total unmapped page cache usage
RMAI	Rate of memory access per instruction
TDSP	Total disk capacity space utilization
LOXD	Longest duration of disk I/O detected
TDIO	Total disk I/O time access all disks

c: ALIBABA TRACE

Alibaba cluster trace [56] includes a higher number of machines and a longer duration. It is composed of 4K machines, 9K online services, and 4M batch jobs that save static and runtime data for 8 days. The dataset consists of three distinct traces: servers, online services, and batch jobs. Over a period of 8 days, we randomly choose 1,000 machines from the server trace, each with around 7,000 traces. Then, we extract various significant metrics relevant to workload prediction, such as the machine ID, timestamp, CPU utilization, memory utilization, memory bandwidth, and disc I/O consumption of each trace. As with the previous two datasets, we only consider CPU utilization from this trace in our experimental investigation.

2) FORECASTING MODELS AND ADVERSARIAL ATTACK METHODS

In order to assess the robustness of DL models against adversarial attacks for the purpose of cloud workload forecasting and to determine whether the adversarial workload generated can be distributed across multiple models, we looked at five popular methods for cloud workload forecasting: 1D-CNN, LSTM, GRU, RNN and attention based LSTM along with two white box adversarial attack methods, FGSM and PGD, which are presented in details in III-B and III-C respectively.

3) EVALUATION METRICS

This research study has employed two commonly used performance indicators, specifically the root mean squared error (RMSE) and the empirical correlation coefficient (CORR). The details of these specified measurements are given as follows:

- RMSE is a statistical metric used to assess the accuracy and reliability of predictions by quantifying the standard deviation of the prediction errors. Mathematically, the formulation is expressed as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(\hat{W}_t - W_t)^2} \tag{18}$$

  where T symbolizes the quantity of samples to be taken into account. The real values and expected values at a specific time t are represented by the variables W and $\hat{W}$ correspondingly.
- CORR determines the strength and direction of the linear association between two variables. This statement elucidates the extent to which the fluctuations in one variable can be accounted for by the fluctuations in another one. Mathematically, the equation can be represented as follows:

$$CORR = \frac{\sum(W - mean(W))(\hat{W} - mean(\hat{W}))}{\sqrt{\sum(W - mean(W))^2 \sum(\hat{W} - mean(\hat{W}))^2}} \tag{19}$$

  here, W and $\hat{W}$ represents the actual and foretasted values respectively.

4) HYPERPARAMETERS

The hyperparameters for RNN, LSTM, and GRU models followed a conventional architectural design as described in previous studies [36], while the attention-based LSTM model adhered to a standard architectural design as mentioned in [38]. This configuration consisted of two hidden layers with a size of 64, utilizing a tanh activation function and including a dropout rate of 0.15 to mitigate overfitting. 1D-CNN model included two filters, each with a size of 64, and the kernel size is set to 3. The L1 loss function is employed for training all the models due to its inherent robustness in handling anomalies present in real-time series data [57]. For the FGSM attack, as outlined in Algorithm 1, the perturbation levels were set at 0.01, 0.05, 0.1, 0.15, and 0.2, respectively. For the PGD attack, we maintained a consistent perturbation range while keeping the number of iterations fixed at 50 and employing a step size of 0.01. The perturbation values were carefully chosen to assess the resilience of workload forecasting models when subjected to different levels of perturbation. The selected perturbation values were determined by a systematic investigation of the parameter space. The experiment begins by employing lower perturbation values to investigate the sensitivity of the model. The perturbation levels are then gradually increased to evaluate their influence on the model's resilience. The purpose of this strategy was to replicate many possible hostile situations, ensuring that the study encompasses both nuanced and more forceful attacks.

TABLE 3. RMSE loss after FGSM adversarial attack against forecasting models on Bitbrain, Google trace, and Alibaba trace dataset.

Datasets	Models	ϵ				
		0.01	0.05	0.10	0.15	0.20
Bitbrain	RNN	0.013	0.057	0.113	0.167	0.221
	LSTM	0.013	0.054	0.107	0.159	0.213
	GRU	0.012	0.051	0.100	0.149	0.197
	1D-CNN	0.006	0.011	0.013	0.015	0.017
	Attention based LSTM	0.001	0.004	0.011	0.016	0.017
Goolge Trace	RNN	0.022	0.056	0.097	0.137	0.174
	LSTM	0.023	0.068	0.138	0.166	0.197
	GRU	0.022	0.074	0.133	0.163	0.189
	1D-CNN	0.022	0.034	0.040	0.043	0.046
	Attention based LSTM	0.002	0.007	0.018	0.041	0.084
Alibaba Trace	RNN	0.091	0.111	0.139	0.166	0.189
	LSTM	0.091	0.110	0.131	0.166	0.180
	GRU	0.106	0.148	0.168	0.181	0.192
	1D-CNN	0.107	0.151	0.170	0.176	0.183
	Attention-based LSTM	0.015	0.026	0.034	0.037	0.038

TABLE 4. RMSE loss after PGD adversarial attack against forecasting models on Bitbrain, Google trace, and Alibaba trace dataset.

Datasets	Models	ϵ				
		0.01	0.05	0.10	0.15	0.20
Bitbrain	RNN	0.008	0.027	0.049	0.071	0.093
	LSTM	0.005	0.012	0.020	0.029	0.037
	GRU	0.007	0.023	0.040	0.057	0.074
	1D-CNN	0.010	0.030	0.052	0.081	0.101
	Attention-based LSTM	0.001	0.002	0.004	0.009	0.016
Goolge Trace	RNN	0.015	0.020	0.025	0.030	0.035
	LSTM	0.017	0.035	0.062	0.090	0.101
	GRU	0.017	0.035	0.063	0.085	0.099
	1D-CNN	0.016	0.025	0.029	0.034	0.038
	Attention-based LSTM	0.002	0.003	0.004	0.005	0.006
Alibaba Trace	RNN	0.089	0.098	0.113	0.129	0.146
	LSTM	0.079	0.090	0.109	0.119	0.136
	GRU	0.097	0.136	0.182	0.229	0.275
	1D-CNN	0.098	0.138	0.189	0.242	0.295
	Attention-based LSTM	0.013	0.020	0.029	0.039	0.050

B. RESULT ANALYSIS

1) IMPACT OF ADVERSARIAL ATTACK

Tables 3 and 4 illustrate the results of our research on the Bitbrain, Google trace, and Alibaba trace datasets, which demonstrate the impact of FGSM and PGD adversarial attacks on DL-based cloud workload forecasting models. As the perturbation parameter ϵ is steadily increased, we observe a consistent rise in RMSE values for all forecasting models. Notably, lower ϵ values can also result in a substantial decline in the performance of forecasting models. The deliberate addition of noise, as indicated by the perturbation parameter ϵ, provides valuable insights into the resilience of the models against various attack scenarios. As the value of ϵ increases in the Bitbrain dataset, we consistently find an increase in RMSE values across all forecasting models. For instance, the RMSE loss for RNN, LSTM, GRU, 1D-CNN, and attention-based LSTM on the Bitbrain dataset increases by 338.46%, 315.38%, 325%, 83.33%, and 300%, respectively, when ϵ increases from 0.01 to

TABLE 5. CORR between actual values and the forecasted values after FGSM adversarial attack against forecasting models on Bitbrain, Google trace, and Alibaba trace dataset.

Datasets	Models	ϵ				
		0.01	0.05	0.10	0.15	0.20
Bitbrain	RNN	-0.182	-0.229	-0.242	- 0.247	-0.251
	LSTM	-0.184	-0.231	-0.244	-0.249	-0.253
	GRU	-0.139	-0.183	-0.195	-0.201	-0.205
	1D-CNN	-0.080	-0.079	-0.073	-0.073	-0.073
	Attention-based LSTM	-0.114	-0.165	-0.180	-0.179	-0.161
Goolge Trace	RNN	-0.068	-0.169	-0.201	-0.213	-0.219
	LSTM	-0.357	-0.422	-0.428	-0.408	-0.392
	GRU	-0.385	-0.445	-0.436	-0.415	-0.392
	1D-CNN	-0.062	-0.079	-0.077	-0.075	-0.072
	Attention-based LSTM	-0.142	-0.225	-0.229	-0.222	-0.215
Alibaba Trace	RNN	- 0.019	-0.383	-0.585	-0.660	-0.694
	LSTM	- 0.022	-0.389	-0.594	-0.667	-0.695
	GRU	-0.034	-0.371	-0.448	-0.462	-0.449
	1D-CNN	-0.057	-0.382	-0.456	-0.454	-0.410
	Attention based LSTM	-0.116	-0.519	-0.599	-0.604	-0.576

TABLE 6. CORR between actual values and the forecasted values after PGD adversarial attack against forecasting models on Bitbrain, Google trace, and Alibaba trace dataset.

Datasets	Models	ϵ				
		0.01	0.05	0.10	0.15	0.20
Bitbrain	RNN	-0.154	-0.212	-0.230	-0.238	-0.244
	LSTM	-0.157	-0.211	-0.229	-0.237	-0.242
	GRU	-0.112	-0.166	-0.183	-0.190	-0.196
	1D-CNN	-0.141	-0.183	-0.190	-0.203	-0.204
	Attention-based LSTM	-0.078	-0.143	-0.163	-0.170	-0.174
Goolge Trace	RNN	-0.031	-0.115	-0.164	-0.189	-0.203
	LSTM	-0.272	-0.406	-0.441	-0.450	-0.449
	GRU	-0.287	-0.441	-0.473	-0.480	-0.480
	1D-CNN	-0.067	-0.113	-0.122	-0.123	-0.124
	Attention-based LSTM	-0.069	-0.186	-0.221	-0.232	-0.238
Alibaba Trace	RNN	-0.087	-0.169	-0.401	-0.533	-0.607
	LSTM	-0.089	-0.169	-0.502	-0.549	-0.701
	GRU	-0.083	-0.292	-0.464	-0.537	-0.578
	1D-CNN	-0.064	-0.300	-0.448	-0.511	-0.548
	Attention-based LSTM	-0.028	-0.399	-0.572	-0.639	-0.674

0.05 under the FGSM attack (Table 3). This emphasizes the vulnerability of the models to small perturbations under the FGSM attack. However, Table 3 illustrates that 1D-CNN is marginally more resilient than other models against FGSM attacks. And, in terms of PGD attack, 1D-CNN is more vulnerable than other models where the attention-based LSTM model shows more robustness under this attack as indicated in Table 4. Similar patterns have been observed with the Google trace and Alibaba trace datasets, where increasing ϵ leads to a rise in RMSE values, demonstrating the models' vulnerability to adversarial attacks. Figure 7 shows the visualized representation of the performance of five DL-based workload forecasting models on the Bitbrain dataset against both adversarial attacks. Furthermore, the correlation values between the prediction results derived from the adversarial workloads and the results obtained from the actual workload are presented in Table 5 and 6. These correlation values are calculated for the four state-of-the-art prediction models and the attention-based LSTM model. As the value of the disturbance ϵ is increased the data undergo perturbation, resulting in a decrease in correlation. Based on the findings of the aforementioned experiment, it can be inferred that cloud workload forecasting models that rely on deep learning techniques exhibit vulnerability to adversarial attacks.

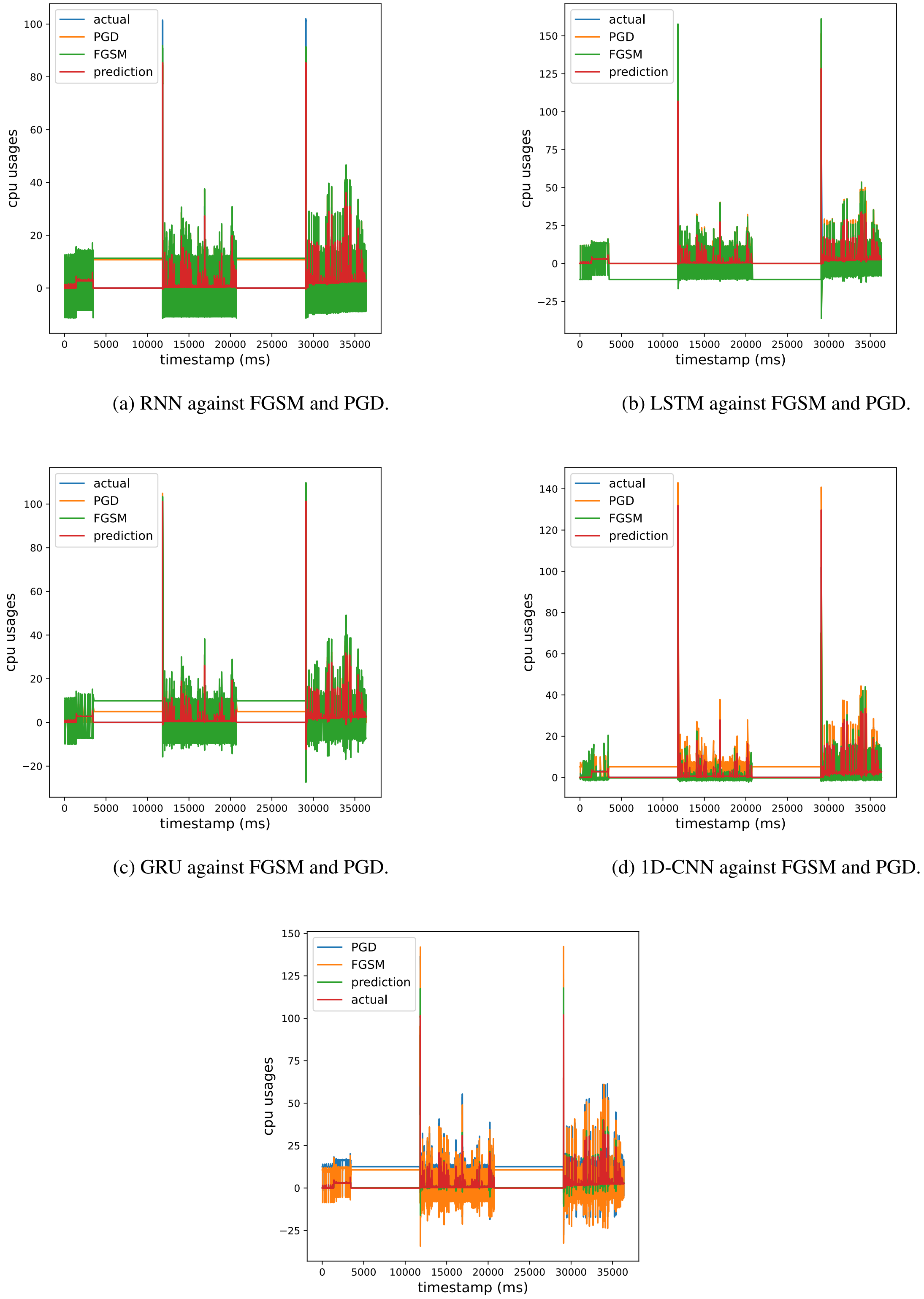
2) PERFORMANCE VARIATION VS THE AMOUNT OF PERTURBATION

In Figure 8, the performance of the model is assessed in relation to the varying amount of perturbations permitted for generating the adversarial workload samples. Overall, across all five cutting-edge forecasting models, RMSE values of the prediction approaches exhibit an upward trend as the level of perturbations increases. This observation highlights the susceptibility of workload forecasting models to adversarial attacks. Based on the findings presented in Figure 8, it can be observed that the PGD attack poses a greater threat to the accuracy of predictive models compared to the FGSM attack, particularly when considering larger values of ϵ. When comparing different values of ϵ, it is observed that the use of a greater ϵ in the PGD method results in a generation of more potent adversarial instances that are capable of deceiving all prediction models. This discrepancy arises from the observation that PGD introduces a minor disturbance during each iteration, while FGSM introduces a fixed amount of noise, epsilon, for each data point. The reason for this discrepancy [58] is that PGD introduces a minor perturbation at each iteration, while FGSM introduces an ϵ amount of disturbance for each point of data.

3) TRANSFERABILITY OF ADVERSARIAL WORKLOADS

A transferable attack is capable of generating adversarial examples that are specifically designed to deceive a workload load prediction model. However, it is worth noting that this attack has the potential to also mislead other forecasting algorithms. Here, on the Bitbrain and Google trace datasets, we investigate the transferability of both adversarial attack methods among the four workload prediction models. The outcomes of transferable attacks, wherein adversarial cloud workloads are created for one model and subsequently employed as input for the other models at ϵ values of 0.0 and 0.1, are presented in Table 7 and 8.

We observe that in terms of PGD attack for both datasets, the adversarial workload examples crafted for GRU are the most transferable. In terms of the FGSM attack for the Bitbrain dataset, the adversarial workload generated for GRU again is the most transferable but for the Google trace dataset adversarial workload generated for LSTM is the most transferable. This means most cases a higher RMSE is observed when the adversarial workload examples crafted for the GRU model are transferred to other models. After a thorough investigation of the experimental outcomes presented in Table 7 and 8, it becomes evident that both adversarial attacks are capable of transferring between the



(a) RNN against FGSM and PGD.

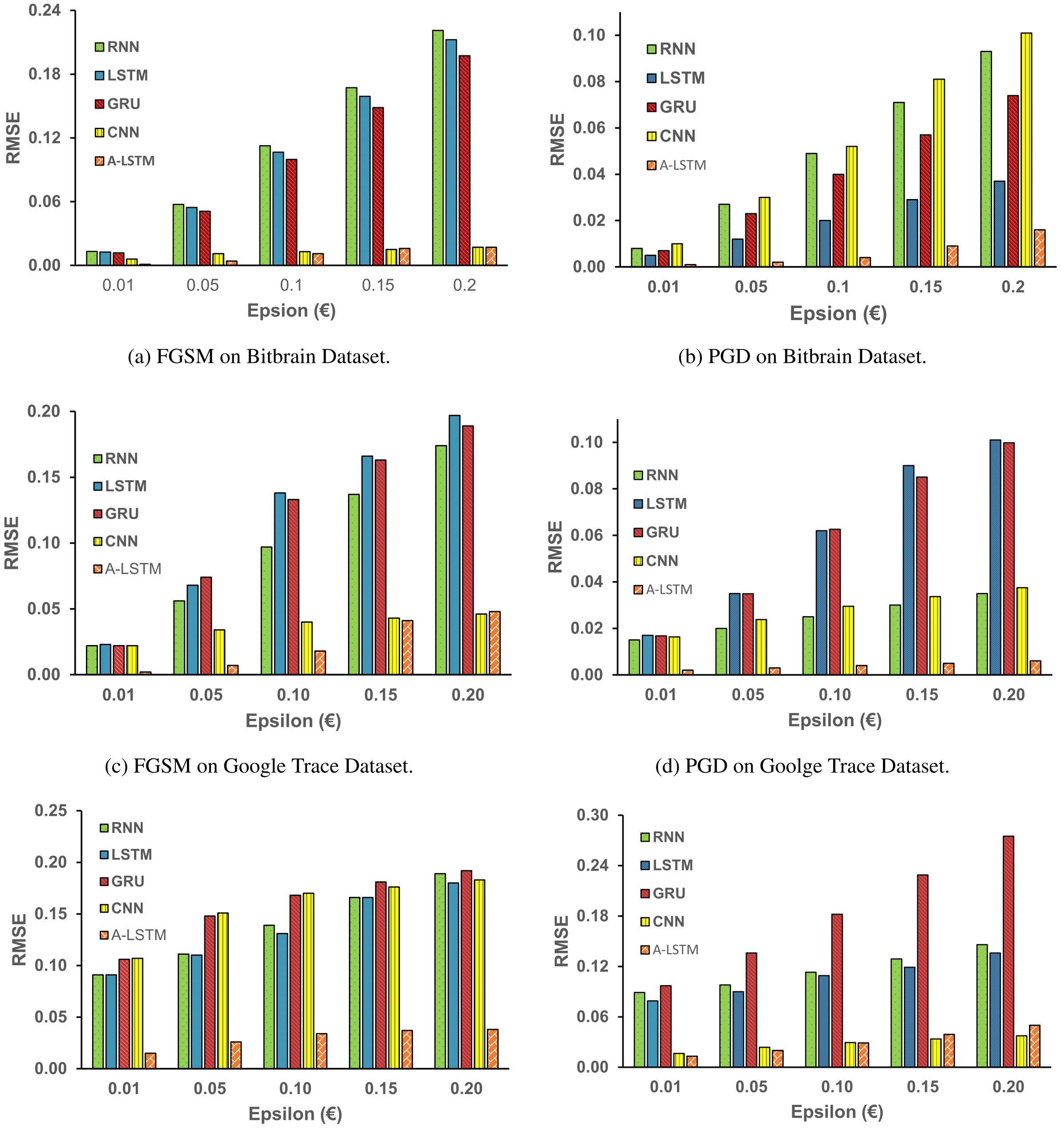
(b) LSTM against FGSM and PGD.

(c) GRU against FGSM and PGD.

(d) 1D-CNN against FGSM and PGD.

(e) LSTM with attention against FGSM and PGD.

FIGURE 7. The impact of FGSM and PGD adversarial attack against DL-based forecasting models in terms of CPU usage of the Birbrain dataset. (a), (b), (c), (d), and (e) show the effects of adversarial attacks on RNN, LSTM, GRU, 1D-CNN, and attention-based LSTM, respectively.



(a) FGSM on Bitbrain Dataset.

(b) PGD on Bitbrain Dataset.

(c) FGSM on Google Trace Dataset.

(d) PGD on Goolge Trace Dataset.

(e) FGSM on Alibaba Trace Dataset.

(f) PGD on Alibaba Trace Dataset.

FIGURE 8. RMSE values after adversarial attack against workload forecasting methods in terms of different perturbation amounts on Bitbrain, Google Trace, and Alibaba Trace dataset. (a)-(b) shows results after FGSM and PGD attacks with different perturbation amounts on the Bitbrain Dataset respectively. (c)-(d) shows results after FGSM and PGD attacks with different perturbation amounts on Google Trace Dataset respectively. (e)-(f) shows results after FGSM and PGD attacks with different perturbation amounts on the Alibaba Trace Dataset respectively.

TABLE 7. Transfer attack with FGSM.

Dataset	Epsilon	From RNN			From LSTM			From GRU			From 1D-CNN		
		LSTM	GRU	1D-CNN	RNN	GRU	1D-CNN	RNN	LSTM	1D-CNN	RNN	LSTM	GRU
Bitbrain	0.01	0.012	0.011	0.006	0.011	0.009	0.005	0.012	0.012	0.011	0.006	0.006	0.006
	0.05	0.052	0.051	0.017	0.047	0.039	0.014	0.054	0.052	0.039	0.022	0.025	0.019
Google Trace	0.01	0.019	0.019	0.016	0.021	0.019	0.039	0.021	0.021	0.021	0.016	0.015	0.015
	0.05	0.036	0.048	0.029	0.051	0.068	0.117	0.051	0.063	0.037	0.025	0.022	0.028

models. Therefore, the adversarial burden generated against other forecasting models is also able to deceive the target prediction method, even if the particular target model is unknown.

TABLE 8. Transfer attack with PGD.

Dataset	Epsilon	From RNN			From LSTM			From GRU			From 1D-CNN		
		LSTM	GRU	1D-CNN	RNN	GRU	1D-CNN	RNN	LSTM	1D-CNN	RNN	LSTM	GRU
Bitbrain	0.01	0.002	0.002	0.002	0.005	0.004	0.004	0.007	0.007	0.007	0.006	0.005	0.005
	0.05	0.004	0.004	0.003	0.011	0.011	0.009	0.024	0.024	0.022	0.024	0.022	0.021
Google Trace	0.01	0.013	0.013	0.013	0.017	0.016	0.015	0.017	0.016	0.016	0.014	0.013	0.013
	0.05	0.015	0.015	0.019	0.029	0.030	0.033	0.029	0.033	0.039	0.016	0.015	0.015

VI. DEFENSE AGAINST ADVERSARIAL ATTACKS

Several defense strategies against adversarial attacks have been proposed by scholars [59], with a primary emphasis on the visual area. There are three distinct categories in which tactics for defending against adversarial attacks can be classified: data change, model transformation, and the use of additional tools. The act of modifying data involves making alterations to the training dataset during the training phase or adjusting the input data during the testing phase. Additionally, the techniques encompass adversarial training [26], transferability restriction [60], compression of data [61], gradient concealment [62], and data randomization [63]. On the other hand, the term ''modifying models'' pertains to the alteration of DL models, which includes techniques such as defensive distillation [64], feature compressing [65], deep contractive network [66], and mask defense [67]. The incorporation of auxiliary techniques into deep learning models is commonly known as using additional tools. These tools encompass defense-GAN [68], MagNet [69], and high-level representation guided denoiser [70]. Regrettably, a significant drawback of the majority of these detectors lies in their susceptibility to adversarial assaults. This vulnerability arises from the deliberate design of these attacks, which aims to deceive the aforementioned detectors [71]. Therefore, it is imperative for researchers in the fields of time series analysis, data mining, and machine learning to give particular consideration to this domain. This is due to the increasing popularity of deep learning models in cloud computing domains that prioritize safety and cost-effectiveness. One possible approach to identify adversarial cases in cloud workload forecasting is to employ an inductive conformal anomaly detection technique [72]. Another possible approach involves utilizing the wide range of research on non-probabilistic classifiers, which are the combination of nearest neighbor algorithms with dynamic time warping [71].

VII. CONCLUSION AND FUTURE WORK

This research addresses the potential risks associated with adversarial attacks targeting cloud workload prediction methods. To the best of our understanding, there is a lack of research conducted on adversarial attacks against cloud workload forecasting models. Existing research has solely concentrated on forecasting the future workload in cloud data centers. To investigate the robustness and security of predictive models, we therefore focus on the adversarial attack on workload prediction. The experimental results of this study demonstrate that all state-of-the-art workload forecasting models are susceptible to adversarial attacks, which can have disastrous security implications for cloud data centers. In the future, we will focus on the development of enhanced adversarial attack methods specifically tailored for cloud workload data. This is due to the fact that changes in cloud workload data are more perceptible to human observation compared to alterations in picture data. The notion that a change in picture usage is imperceptible to the human eye remains inaccurate in the context of cloud workload data. Furthermore, our research will encompass an investigation of defense mechanisms aimed at identifying and mitigating hostile risks inside deep-learning forecasting models.

REFERENCES

[1] T. Dillon, C. Wu, and E. Chang, ''Cloud computing: Issues and challenges,'' in *Proc. 24th IEEE Int. Conf. Adv. Inf. Netw. Appl.*, Apr. 2010, pp. 27–33.

[2] C. Gong, J. Liu, Q. Zhang, H. Chen, and Z. Gong, ''The characteristics of cloud computing,'' in *Proc. 39th Int. Conf. Parallel Process. Workshops*, Sep. 2010, pp. 275–279.

[3] W. Voorsluys, J. Broberg, and R. Buyya, ''Introduction to cloud computing,'' in *Cloud Computing: Principles and Paradigms*, 2011, pp. 1–41.

[4] Z. Wang, M. M. Hayat, N. Ghani, and K. B. Shaban, ''Optimizing cloud-service performance: Efficient resource provisioning via optimal workload allocation,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 6, pp. 1689–1702, Jun. 2017.

[5] Z. Chen, J. Hu, and G. Min, ''Learning-based resource allocation in cloud data center using advantage actor-critic,'' in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[6] F. Xu, H. Zheng, H. Jiang, W. Shao, H. Liu, and Z. Zhou, ''Cost-effective cloud server provisioning for predictable performance of big data analytics,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 5, pp. 1036–1051, May 2019.

[7] Y. S. Patel and J. Bedi, ''MAG-D: A multivariate attention network based approach for cloud workload forecasting,'' *Future Gener. Comput. Syst.*, vol. 142, pp. 376–392, May 2023.

[8] S. Subramanian and A. Kannammal, ''Real time non-linear cloud workload forecasting using the holt-winter model,'' in *Proc. 10th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2019, pp. 1–6.

[9] Q. Zhang, M. F. Zhani, S. Zhang, Q. Zhu, R. Boutaba, and J. L. Hellerstein, ''Dynamic energy-aware capacity provisioning for cloud computing environments,'' in *Proc. 9th Int. Conf. Autonomic Comput.*, Sep. 2012, pp. 145–154.

[10] V. Podolskiy, A. Jindal, M. Gerndt, and Y. Oleynik, ''Forecasting models for self-adaptive cloud applications: A comparative study,'' in *Proc. IEEE 12th Int. Conf. Self-Adapt. Self-Organizing Syst. (SASO)*, Sep. 2018, pp. 40–49.

[11] Z. Gong, X. Gu, and J. Wilkes, ''PRESS: PRedictive elastic ReSource scaling for cloud systems,'' in *Proc. Int. Conf. Netw. Service Manage.*, Oct. 2010, pp. 9–16.

[12] H. Nguyen, Z. Shen, X. Gu, S. Subbiah, and J. Wilkes, ''AGILE: Elastic distributed resource scaling for infrastructure-as-a-service,'' in *Proc. 10th Int. Conf. Autonomic Comput.*, 2013, pp. 69–82.

[13] J. Kumar, R. Goomer, and A. K. Singh, "Long short term memory recurrent neural network (LSTM-RNN) based workload forecasting model for cloud datacenters," *Proc. Comput. Sci.*, vol. 125, pp. 676–682, Jan. 2018.

[14] N. Zaini, L. W. Ean, A. N. Ahmed, and M. A. Malek, "A systematic literature review of deep learning neural network for time series air quality forecasting," *Environ. Sci. Pollut. Res.*, vol. 29, no. 4, pp. 4958–4990, Jan. 2022.

[15] M. Xu, C. Song, H. Wu, S. S. Gill, K. Ye, and C. Xu, "EsDNN: Deep neural network based multivariate workload prediction in cloud computing environments," *ACM Trans. Internet Technol.*, vol. 22, no. 3, pp. 1–24, Aug. 2022.

[16] Y. S. Patel and R. Misra, "Performance comparison of deep vm workload prediction approaches for cloud," in *Proc. ICCAN*. Cham, Switzerland: Springer, 2017, pp. 149–160.

[17] R. Li, A. X. Liu, A. L. Wang, and B. Bruhadeshwar, "Fast and scalable range query processing with strong privacy protection for cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 4, pp. 2305–2318, Aug. 2016.

[18] G. Aceto, V. Persico, and A. Pescapé, "Industry 4.0 and health: Internet of Things, big data, and cloud computing for healthcare 4.0," *J. Ind. Inf. Integr.*, vol. 18, Jun. 2020, Art. no. 100129.

[19] Y. Miao, Y. Yang, X. Li, L. Wei, Z. Liu, and R. H. Deng, "Efficient privacy-preserving spatial data query in cloud computing," *IEEE Trans. Knowl. Data Eng.*, 2023.

[20] Y. Miao, F. Li, X. Li, Z. Liu, J. Ning, H. Li, K. R. Choo, and R. H. Deng, "Time-controllable keyword search scheme with efficient revocation in mobile E-health cloud," *IEEE Trans. Mobile Comput.*, 2023.

[21] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Comput. Ind. Eng.*, vol. 149, Jan. 2020, Art. no. 106854.

[22] D. Xiao, B. Cao, and W. Wu, "EFL-WP: Federated learning-based workload prediction in inter-cloud environments," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2022, pp. 1–10.

[23] J. Guo, Z. Liu, S. Tian, F. Huang, J. Li, X. Li, K. K. Igorevich, and J. Ma, "TFL-DT: A trust evaluation scheme for federated learning in digital twin for mobile networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 11, pp. 3548–3560, Nov. 2023.

[24] R. Kumar and R. Goyal, "On cloud security requirements, threats, vulnerabilities and countermeasures: A survey," *Comput. Sci. Rev.*, vol. 33, pp. 1–48, Aug. 2019.

[25] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*.

[26] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.

[27] M. Duggan, K. Mason, J. Duggan, E. Howley, and E. Barrett, "Predicting host CPU utilization in cloud computing using recurrent neural networks," in *Proc. 12th Int. Conf. Internet Technol. Secured Trans. (ICITST)*, Dec. 2017, pp. 67–72.

[28] Z. Huang, J. Peng, H. Lian, J. Guo, and W. Qiu, "Deep recurrent model for server load and performance prediction in data center," *Complexity*, vol. 2017, pp. 1–10, Jan. 2017.

[29] W. Zhang, B. Li, D. Zhao, F. Gong, and Q. Lu, "Workload prediction for cloud cluster using a recurrent neural network," in *Proc. Int. Conf. Identificat., Inf. Knowl. Internet Things (IIKI)*, Oct. 2016, pp. 104–109.

[30] J. Bi, S. Li, H. Yuan, Z. Zhao, and H. Liu, "Deep neural networks for predicting task time series in cloud computing systems," in *Proc. IEEE 16th Int. Conf. Netw., Sens. Control (ICNSC)*, May 2019, pp. 86–91.

[31] R. C. Staudemeyer and E. Rothstein Morris, "Understanding LSTM—A tutorial into long short-term memory recurrent neural networks," 2019, *arXiv:1909.09586*.

[32] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

[33] B. Song, Y. Yu, Y. Zhou, Z. Wang, and S. Du, "Host load prediction with long short-term memory in cloud computing," *J. Supercomput.*, vol. 74, no. 12, pp. 6554–6568, Dec. 2018.

[34] Z. Chen, J. Hu, G. Min, A. Y. Zomaya, and T. El-Ghazawi, "Towards accurate prediction for high-dimensional and highly-variable cloud workloads with deep learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 4, pp. 923–934, Apr. 2020.

[35] S. Tajalizadehkhoob, M. Korczynski, A. Noroozian, C. Gañán, and M. van Eeten, "Apples, oranges and hosting providers: Heterogeneity and security in the hosting market," in *Proc. NOMS-IEEE/IFIP Netw. Oper. Manage. Symp.*, Apr. 2016, pp. 289–297.

[36] M. E. Karim, M. M. S. Maswood, S. Das, and A. G. Alharbi, "BHyPreC: A novel bi-LSTM based hybrid recurrent neural network model to predict the CPU workload of cloud virtual machine," *IEEE Access*, vol. 9, pp. 131476–131495, 2021.

[37] A. I. Maiyza, N. O. Korany, K. Banawan, H. A. Hassan, and W. M. Sheta, "VTGAN: Hybrid generative adversarial networks for cloud workload prediction," *J. Cloud Comput.*, vol. 12, no. 1, p. 97, Jun. 2023.

[38] Y. Zhu, W. Zhang, Y. Chen, and H. Gao, "A novel approach to workload prediction using attention-based LSTM encoder–decoder network in cloud environment," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, pp. 1–18, Dec. 2019.

[39] A. Kaim, S. Singh, and Y. S. Patel, "Ensemble CNN attention-based BiLSTM deep learning architecture for multivariate cloud workload prediction," in *Proc. 24th Int. Conf. Distrib. Comput. Netw.*, Jan. 2023, pp. 342–348.

[40] J. Dogani, F. Khunjush, M. R. Mahmoudi, and M. Seydali, "Multivariate workload and resource prediction in cloud computing using CNN and GRU by attention mechanism," *J. Supercomput.*, vol. 79, no. 3, pp. 3437–3470, Feb. 2023.

[41] I. Oregi, J. Del Ser, A. Perez, and J. A. Lozano, "Adversarial sample crafting for time series classification with elastic similarity measures," in *Intelligent Distributed Computing XII*. Springer, 2018, pp. 26–39.

[42] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Müller, "Adversarial attacks on deep neural networks for time series classification," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[43] P. Rathore, A. Basak, S. H. Nistala, and V. Runkana, "Untargeted, targeted and universal adversarial attacks and defenses on time series," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8.

[44] W. Yang, J. Yuan, X. Wang, and P. Zhao, "TSadv: Black-box adversarial attack on time series with local perturbations," *Eng. Appl. Artif. Intell.*, vol. 114, Sep. 2022, Art. no. 105218.

[45] G. R. Mode and K. A. Hoque, "Adversarial examples in deep learning for multivariate time series regression," in *Proc. IEEE Appl. Imag. Pattern Recognit. Workshop (AIPR)*, Oct. 2020, pp. 1–10.

[46] D. Yao, B. Li, H. Liu, J. Yang, and L. Jia, "Remaining useful life prediction of roller bearings based on improved 1D-CNN and simple recurrent unit," *Measurement*, vol. 175, Apr. 2021, Art. no. 109166. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263224121001895

[47] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*.

[48] S. Du, T. Li, Y. Yang, and S.-J. Horng, "Multivariate time series forecasting via attention-based encoder–decoder framework," *Neurocomputing*, vol. 388, pp. 269–279, May 2020.

[49] A. Athalye and N. Carlini, "On the robustness of the CVPR 2018 white-box adversarial example defenses," 2018, *arXiv:1804.03286*.

[50] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin, "On evaluating adversarial robustness," 2019, *arXiv:1902.06705*.

[51] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Artificial Intelligence Safety and Security*. Boca Raton, FL, USA: CRC Press, 2018, pp. 99–112.

[52] N. Akhtar, A. Mian, N. Kardan, and M. Shah, "Advances in adversarial attacks and defenses in computer vision: A survey," *IEEE Access*, vol. 9, pp. 155161–155196, 2021.

[53] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, *arXiv:1706.06083*.

[54] (2018). *Bitbrains Cluster Log*. [Online]. Available: http://gwa.ewi.tudelft.nl/datasets/gwa-t-12-bitbrains

[55] C. Reiss, J. Wilkes, and J. L. Hellerstein, "Google cluster-usage traces: Format + schema," Google Inc., White Paper, 2011, vol. 1, pp. 1–14.

[56] J. Guo, Z. Chang, S. Wang, H. Ding, Y. Feng, L. Mao, and Y. Bao, "Who limits the resource efficiency of my datacenter: An analysis of Alibaba datacenter traces," in *Proc. IEEE/ACM 27th Int. Symp. Quality Service (IWQoS)*, Jun. 2019, pp. 1–10.

[57] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long- and short-term temporal patterns with deep neural networks," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jun. 2018, pp. 95–104.

[58] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14410–14430, 2018.

[59] A. Muhammad and S.-H. Bae, "A survey on efficient methods for adversarial robustness," *IEEE Access*, vol. 10, pp. 118815–118830, 2022.

[60] H. Hosseini, Y. Chen, S. Kannan, B. Zhang, and R. Poovendran, "Blocking transferability of adversarial examples in black-box learning systems," 2017, *arXiv:1703.04318*.

[61] N. Das, M. Shanbhogue, S.-T. Chen, F. Hohman, L. Chen, M. E. Kounavis, and D. H. Chau, "Keeping the bad guys out: Protecting and vaccinating deep learning with JPEG compression," 2017, *arXiv:1705.02900*.

[62] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, Apr. 2017, pp. 506–519.

[63] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, "Adversarial examples for semantic segmentation and object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1378–1387.

[64] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 582–597.

[65] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," 2017, *arXiv:1704.01155*.

[66] S. Gu and L. Rigazio, "Towards deep neural network architectures robust to adversarial examples," 2014, *arXiv:1412.5068*.

[67] J. Gao, B. Wang, Z. Lin, W. Xu, and Y. Qi, "DeepCloak: Masking deep neural network models for robustness against adversarial samples," 2017, *arXiv:1702.06763*.

[68] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: Protecting classifiers against adversarial attacks using generative models," 2018, *arXiv:1805.06605*.

[69] D. Meng and H. Chen, "MagNet: A two-pronged defense against adversarial examples," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 135–147.

[70] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, "Defense against adversarial attacks using high-level representation guided denoiser," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1778–1787.

[71] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2805–2824, Sep. 2019.

[72] D. Volkhonskiy, E. Burnaev, I. Nouretdinov, A. Gammerman, and V. Vovk, "Inductive conformal martingales for change-point detection," in *Conformal and Probabilistic Prediction With Applications*, 2017, pp. 132–153.

NOSIN IBNA MAHBUB received the B.Sc. degree from the Department of Information and Communication Technology (ICT), Islamic University, Bangladesh, in 2021. He is currently pursuing the combined Ph.D. degree with the Department of Computer Science and Engineering, Kyung Hee University, Global Campus, South Korea. He has been a Research Student with the Intelligent Computing and Security Laboratory (ICNS Lab), Kyung Hee University, since March 2023. His research interests include cloud computing, adversarial machine learning, edge artificial intelligence (AI), and explainable AI.

MD. DELOWAR HOSSAIN received the B.Sc. and M.Sc. degrees from the Department of Information and Communication Engineering (ICE), Islamic University, Bangladesh, in 2004 and 2005, respectively, and the Ph.D. degree from the Department of Computer Science and Engineering, Kyung Hee University, Republic of Korea. He was a Visiting Scholar with Infosys, Bengaluru, India. He is currently a Professor with the Department of Computer Science and Engineering, Hajee Mohammad Danesh Science and Technology University, Bangladesh, where he was the Chairman, from 2011 to 2013. He is also a Postdoctoral Researcher with the Department of Computer Science and Engineering, Kyung Hee University. His current research interests include cloud/edge/fog computing, vehicular edge computing, big data, machine learning, and the Internet of Things. He was a recipient of the Best Paper Award from KSC 2018, KSC 2019, and KCC 2021, South Korea.

SHARMEN AKHTER received the B.Sc. and M.Sc. degrees from the Department of Information and Communication Technology (ICT), Islamic University, Bangladesh, in 2016 and 2018, respectively. She is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Kyung Hee University, Global Campus, South Korea. She has been a Research Student with the Intelligent Computing and Security Laboratory (ICNS Lab), Kyung Hee University, since 2021. Her research interests include deep learning, image processing, object detection and recognition, visual-based human action recognition in videos, and video surveillance systems.

MD. IMTIAZ HOSSAIN received the B.Sc. and M.Sc. degrees from the Department of Information and Communication Technology (ICT), Islamic University, Bangladesh, in 2016 and 2018, respectively, and the M.S. degree from the Department of Computer Science and Engineering, Kyung Hee University, Global Campus, Yongin-si, South Korea, in 2021, where he is currently pursuing the Ph.D. degree. He has been a Research Student with the Intelligent Computing and Security Laboratory (ICNS Lab), Kyung Hee University, since 2019. His research interests include transfer learning, knowledge distillation, visual-based human action recognition, 3D pose estimation, 3D motion modeling, and metaverse.

KIMOON JEONG received the B.S., M.S., and Ph.D. degrees in computer science from Chonnam National University, in 1999, 2001, and 2009, respectively. From 2001 to 2005, he was a Security Researcher with KISA and NIS. Since 2005, he has been a Network Security and Vulnerability Analyst with Korea Institute of Science and Technology Information (KISTI). His research interests include HPC cloud, supercomputing, and cloud security.

EUI-NAM HUH (Member, IEEE) received the B.S. degree from Busan National University, South Korea, the master's degree in computer science from The University of Texas, USA, in 1995, and the Ph.D. degree from Ohio University, USA, in 2002. He is currently a Professor with the Department of Computer Science and Engineering, Kyung Hee University, South Korea. His research interests include cloud computing, the Internet of Things, future internet, distributed real-time systems, mobile computing, big data, and security. He is on the reviewer board of the National Research Foundation of Korea. He has served many community services for ICCSA, WPDRTS/IPDPS, APAN Sensor Network Group, ICUIMC, ICONI, APIC-IST, ICUFN, and SoICT as various types of chairs. He is the Vice-Chairman of the Cloud/Bigdata Special Technical Group of TTA and an Editor of ITU-T SG13 Q17.