IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 July 2023, accepted 24 July 2023, date of publication 27 July 2023, date of current version 2 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3299331

RESEARCH ARTICLE

Privacy-Preserving On-Screen Activity Tracking and Classification in E-Learning Using Federated Learning

DURJOY MISTRY[1], M. F. MRIDHA[2], (Senior Member, IEEE), MEJDL SAFRAN[3], SULTAN ALFARHOOD[3], ALOKE KUMAR SAHA[1], AND DUNREN CHE[4]

[1]Department of Computer Science and Engineering, University of Asia Pacific, Dhaka 1215, Bangladesh
[2]Department of Computer Science and Engineering, American International University of Bangladesh, Dhaka 1216, Bangladesh
[3]Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia
[4]School of Computing, Southern Illinois University, Carbondale, IL 62901, USA

Corresponding authors: M. F. Mridha (firoz.mridha@aiub.edu) and Mejdl Safran (mejdl@ksu.edu.sa)

The authors extend their appreciation to the Deputyship for Research and Innovation, ''Ministry of Education'' in Saudi Arabia for funding this research (IFKSUOR3–013-1).

ABSTRACT E-learning, a modern method of education that utilizes electronic technologies such as computers, mobile devices, and the internet, has experienced a significant surge in adoption and usage in recent years. While it has the potential to reach every corner of the world, it also creates an opportunity for time and resource wastage. In almost all cases students use the same device for studying and for entertainment purposes. Being one click away from ever-addicting social media, it is very difficult for students to stay focused on studying using digital devices and not waste time on it. The issue is quite significant as online education will be practised more and more in the future. In spite of that, detecting the on-screen activity of students is an underexplored region of research, and to our best knowledge, no research takes protecting their privacy into consideration. Therefore in this research, a privacy-preserving architecture is proposed to detect whether students are utilizing their time on their computer or wasting it while the user's privacy is protected with federated learning. A dataset containing over 4000 screenshots of different activities of students is used to classify them into categories using several pre-trained models where our proposed FedInceptionV3 achieves a state-of-the-art test accuracy of 99.75%.

INDEX TERMS Deep learning, E-learning, federated learning, machine learning, on screen activity detection, transfer learning.

I. INTRODUCTION

As the fourth industrial revolution is taking place, the internet is becoming faster and more accessible to everyone [1]. From education to professional work, digital devices have become essential tools. In recent years, online education has experienced exponential growth, breaking down geographical barriers and offering a flexible learning environment [2]. A study conducted in 2017 revealed a consistent increase in the number of US students enrolling in at least one online course for 14 consecutive years, with over 6 million students taking online courses in 2016. This surge in popularity reflects a growing demand for accessible and flexible education [3]. As e-learning continues to gain traction, its popularity is expected to further soar in the upcoming years.

The growth of online learning is predicted to excel in upcoming years, and it is expected that the global e-learning market is projected to reach $325 billion by 2025 [4]. However, ensuring students' engagement in online learning poses unique challenges. One of the challenges associated with online learning is the potential for unwanted time and resource wastage. Because students often use the same device for both studying and entertainment purposes, this makes it difficult to maintain focus on academic tasks, as social media and other distractions are only a click away. Even unintentionally, students tend to open social media while studying using a digital device. As a result, they end up wasting valuable time instead of dedicating themselves to

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

their studies. S. Khan et al. found that social media use had a negative impact on academic performance, particularly for students who spent more time on social media [5]. This finding is consistent with research by Geot [6]. Another study, conducted by u. et al. in the field of education, uses a cross-sectional survey design to gather data from a sample of 379 students from four universities in the Khyber Pakhtunkhwa region [7]. The results of the study indicate that the majority of the students perceived social media to have a negative impact on their academic performance. The study also found that the amount of time spent on social media was positively correlated with negative perceptions of its impact on academic performance.

However, several studies show that online learning has seen significant growth, driven by factors such as advances in technology, increasing internet penetration, and a global shift towards remote education [8], [9]. As the e-learning ecosystem continues to expand, it becomes increasingly important to prioritize the quality of the learning experience and ensure that students are effectively utilizing their time instead of getting distracted by social media while engaging in online learning activities. While social media site blockers are often seen as a potential solution to mitigate distractions and enhance student focus in online learning environments, they may not always be effective. One of the reasons is that social media site blockers primarily function at the device or browser level, relying on the student's self-discipline to activate and enforce them. However, students may find workarounds or disable these blockers when faced with the temptation to access social media platforms. Furthermore, it is important to acknowledge that social websites like YouTube can serve both educational and entertainment purposes. Therefore, employing a simplistic approach of site blocking, solely based on URLs, may not be intuitive or beneficial. In light of this, we have proposed a more effective approach that involves analyzing the actual content displayed on the screen, rather than relying on URL-based site tracking. It is essential to acknowledge that the utilization of traditional machine learning models for computer screen content analysis raises valid privacy concerns, as the process necessitates the uploading of raw data to a centralized server [10], [11]. To address these challenges, we propose a federated learning-based architecture for privacy-preserving computer activity detection and classification in an e-learning scenario [11], [12].

This paper presents the design, implementation, and evaluation of our proposed architecture, which leverages the strengths of federated learning and transfer learning to provide an effective and privacy-preserving solution [13]. By distributing the learning process across multiple devices and ensuring that sensitive data remains local, our approach ensures the privacy of the end user. Additionally, the use of transfer learning enables the adaptation of pre-trained models to different learning environments, reducing computational and training time costs. The paper also discusses the ethical implications, practical considerations, and potential applications of our architecture in the broader context of online learning [14], [15]. With this research, we aim to contribute to the development of a more robust, secure, and privacy-centric on-screen activity detection system.

The overall contribution of the paper includes:

- A user-friendly privacy-preserving architecture is proposed that can detect students' productivity in online learning.
- Created Federated Learning Architecture to implement six different popular deep learning models for on-screen activity classification.
- The proposed system with FedInceptionV3 predicts 798 samples correctly out of 800 samples in the test sample and achieves a remarkable test accuracy of 99.75%.
- A comprehensive performance analysis of six different models is provided including metrics like Accuracy, Precision, Recall, F1-Score, Loss, and Confusion Matrix.

The paper is organized in the following manner: In Section II, previous research in this area is discussed. The proposed methodology is presented in Section III. Section IV is dedicated to discussing the experimental results. Wrong predictions are discussed in Section V. Section VI is where the limitations and potential future research avenues are addressed along with the conclusion.

II. RELATED WORK

Computer screen activity detection has gained considerable attention in recent years, particularly in the context of user behaviour analysis, employee monitoring, and remote learning. With the development of computer vision techniques, machine learning, and deep learning models, researchers have made significant advancements in detecting and analyzing screen activities. Privacy-preserving vision-based computer activity detection remains an underexplored area of research, while the increasing concern for privacy is evident in the contemporary world. This literature review aims to discuss the state-of-the-art approaches in privacy-preserving techniques and computer screen activity detection. Table 1 represents the state-of-the-art approaches in computer activity tracking and encompasses an exploration of cutting-edge approaches utilizing computer vision, machine learning, and traditional software methods.

A. PRIVACY-PRESERVING TECHNIQUES

Many different privacy-preserving techniques Privacy-preserving techniques have gained significant attention in research and industry. For instance, Differential Privacy is a framework that provides mathematical guarantees for protecting the privacy of individual data by adding noise to the query results, ensuring that the presence or absence of any individual's data has a limited impact on the overall outcome [19]. On the other hand, Homomorphic Encryption is a cryptographic technique that allows computations to be performed on encrypted data without decrypting it, preserving the privacy and security of the data throughout the

TABLE 1. State-of-the-art technologies in activity monitoring.

Paper / Software	Method	Dataset	Accuracy	Limitations
Software [25-31]	These software solutions track computer activity through methods such as capturing screenshots, tracking usage, recording data, etc.	N/A	N/A	Requires designated monitor; Cannot distinguish between productive and unproductive use of social media; Has privacy issues and expensive.
Paper [16]	Video Screen Capture technology to analyze user behaviour	N/A	N/A	Video Screen Capture is resource expensive; Does not offer any automatic detection.
Paper [17]	Computer-vision based Machine Learning to track activity of students from Mobile Screen Recordings	118 Videos	N/A	Continius mobile screen recording is not practical as it slows down the performance by a significant margin, additionally it raises privacy concern.
Paper [18]	Deep Learning based activity tracking from computer screenshots	4000 Screenshots	95.4%	Computationally less expensive, but does not take privacy into consideration.

computation process [20]. Secure Multi-Party Computation (MPC) is another cryptographic protocol that enables multiple parties to jointly compute a function on their private inputs without revealing the individual inputs to each other. It ensures privacy and confidentiality while allowing collaborative computation on sensitive data [21]. Instead of utilizing the above-mentioned techniques, we opted for Federated Learning, which is a decentralized approach to machine learning. In this method, model training occurs locally on individual devices or edge servers, and the resulting model updates are shared and merged to enhance the overall model [22]. We prepared Table 2 which is a comparison among privacy-preserving techniques based on [22], [23], and [24] and we can observe that federated learning offers the best trade-off between privacy and utility for image classification. It is more privacy-preserving than differential privacy, while also being more accurate than homomorphic encryption. It is also less computationally complex than secure multi-party computation, and it has lower communication complexity than all of the other models. Therefore federated learning was chosen to protect the privacy of the user as our proposed model works on vision-based image processing.

B. SOFTWARE BASED APPROACH

Screen capture technology has been widely used in the field of user research to study user behaviour. There are various third-party software options for screen monitoring such as Hubstaff [25], Teramind [26], Workpuls [27], DeskTime [28], Time Doctor [29], InterGuard [30], and ActivTrak [31], which utilize features like screenshot capture, email and chat monitoring, website tracking, and file monitoring to determine the active and idle time of users. These systems rely on a designated monitor to oversee user activity. Although they are capable of monitoring employee social media use during work hours, they lack the ability to distinguish between the use of social media for work-related or personal purposes. Moreover, these software options are not open source and require a subscription fee for access to their features as well as they require self-supervision to track activities and don't notify the user about potential time wastage. Importantly, the study conducted by Hankerson et al. talks about the privacy concern of students using monitoring software on their devices [32]. The study highlights the potential privacy concerns associated with such technology, including unauthorized access to student information and the risk of data breaches. The authors emphasize the need for greater transparency and accountability in the use of these devices, recommending the implementation of clear policies and procedures for data collection and use, as well as regular privacy audits to protect student privacy.

C. MACHINE-LEARNING BASED APPROACH

Computer screen activity classification using vision-based machine learning is a relatively under-explored region of research. Imler and Eichelberger discuss the use of screen capture technology in the field of user research [16]. The authors argue that screen capture allows for a more naturalistic and unobtrusive observation of user behaviour, providing valuable insights into user preferences and needs. The benefits of using screen capture in user research include its ability to provide a more accurate representation of user behaviour, the ability to analyze user interactions in their natural environment, and the ability to gain a deeper understanding of user decision-making processes and problem-solving strategies. The authors conclude that screen capture technology is a valuable tool for studying user behaviour and improving the design and development of digital products and services. One study by Krieter and Breiter involved collecting screen recordings of students using a mobile app for learning [17]. The recordings were then analyzed to generate log files that captured various aspects of the student's behaviour, such as the duration of their interactions with the app and the types of activities they engaged in. The log files were then used to generate visualizations and reports that provided insights into the student's learning behaviour. However, the performance of the device can be significantly hindered by recording screens for longer periods or analyzing video files to identify user activity. The issue is addressed by a study by Ferdosi et al. and they propose an architecture that works with screenshots to detect the activity of students on digital devices [18]. While their suggested approach is effective and user-friendly, it fails to address privacy concerns. Since every screenshot must

TABLE 2. Comparison of privacy preserving techniques.

Privacy Preserving Techniques	Privacy	Utility	Accuracy	Computational Complexity	Communication Complexity
Differential privacy	High	Low	Low	Low	Low
Homomorphic encryption	Medium	High	High	High	High
Secure multi-party computation	High	Medium	Medium	High	High
Federated learning	High	Medium	High	Low	Low

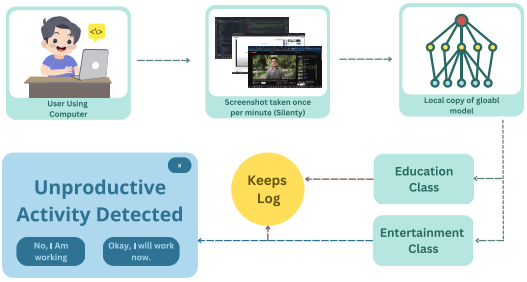


FIGURE 1. Systematic overview of proposed framework.

be sent to a central server for classification, it poses a risk to the user's privacy. The user might have sensitive content on their computer, like personal photos, exposed passwords, or bank details, and they would understandably be unwilling to upload a screenshot containing such information to a third-party server.

Our research is the first one to take the privacy issue into consideration in vision-based computer activity detection and we implemented a solution for on-screen activity detection to protect the user's privacy. Our approach ensures that user data is not collected during model development, which is especially beneficial for students who can rest assured about their privacy. To achieve this, we have incorporated federated learning, enabling us to identify and analyze on-screen activities without compromising user privacy. Importantly, the entire detection process takes place solely on the user's device, with no transmission or uploading of local data to the cloud. This meticulous approach guarantees that the user's private information remains confined to their device, preserving their privacy with the utmost care and diligence.

III. PROPOSED METHODOLOGY

In this section we will first explain the dataset and then the workflow of the proposed framework is explained and after that, we explain how privacy is being preserved with federated learning.

A. PROPOSED FRAMEWORK

In a usual scenario, the users may want to use their computers for entertainment or work. With our proposed system we will be able to detect that in a privacy-preserving manner. The workflow of our proposed framework is explained step by step with the diagram drawn in Figure 1.

Framework Explanation:

- Step 1: Once our services are started the system will silently take a screenshot of the entire screen once every minute.

TABLE 3. Log format of activity detection.

Time Stamp	Activity	Class
08/05/2023 03:05:15 PM	Productive	Education Coursera
08/05/2023 03:06:15 PM	Productive	Education YouTube
08/05/2023 03:07:15 PM	Unproductive	Entertainment YouTube

- Step 2: After that, the screenshot will be analyzed by the local model and it will either belong to the education class or entertainment class.
  - The program stays silent if the screenshot belongs to education classes and registers the log as productive. The logs are stored in the following formats shown in table 3.
  - If back-to-back two screenshots belong to the entertainment class the user will be notified with a pop-up "Unproductive Activity Detected" which has the following options with the following consequences.
    - No, I Am Working: Clicking this option counts the prediction as a failure and updates the model parameters and registers the log as productive.
    - Okay, I Will Work Now: Clicking this option counts the prediction as a success thus it keeps the model parameters, and registers the log as entertainment.
    - X: Clicking the X option will stop the service and the last screenshot will have no logs.

Following the registration of the log, it is ensured that all screenshots are permanently deleted from the local device, thereby mitigating any potential burden on the system resources. In order to minimize the potential distractions caused by our system, a decision was made to set the time duration for detecting unproductive activity to one minute. This approach was chosen because it was deemed that a brief period of unproductive activity, such as entertainment use on the computer, would not significantly impact the user's overall productivity. However, a system that constantly generates notifications for every instance of unproductive activity would be counter-intuitive and may discourage the user from utilizing the system. In instances where educational YouTube content is being viewed, it is possible for entertaining YouTube advertisements to appear. In such situations, the system may classify the occurrence as an unproductive activity. To address this issue, the system is designed to capture a silent screenshot once per minute and only alert the user if two consecutive screenshots indicate unproductive activity. As the entertaining YouTube ad would have been skipped or ended within a one-minute period, the user is not likely to receive a false alarm, thus preventing unnecessary alerts from being triggered.

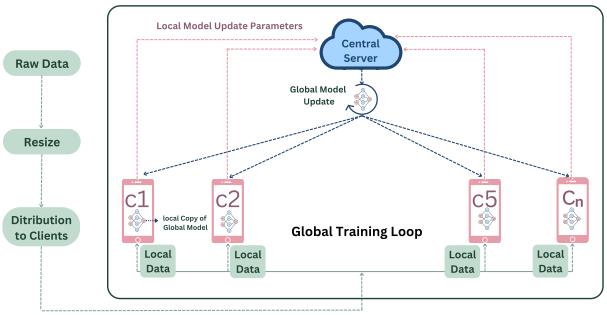


FIGURE 2. An abstract view of federated learning architecture.

By implementing this approach, the system aims to maintain its effectiveness in identifying unproductive activity while minimizing disruptions to the user. And thus our system can detect if a user is utilizing his time on his computer and or not in a user-friendly manner. But how is privacy protected? That is explained in the section III-B.

B. PRIVACY PRESERVATION WITH FEDERATED LEARNING

The privacy of a user in our system is protected with federated learning. If our system was deployed on a single centralized server in traditional machine learning architecture the screenshots would have needed to be uploaded on the central cloud to train it. This creates a great opportunity for a security breach and also the privacy of the user gets exposed. But Federated learning provides privacy by enabling machine learning models to be trained on decentralized data sources, without requiring the data to be centralized in one location.

In federated learning, the data remains on the individual devices or servers that generate it, and the model is trained on each of these decentralized data sources in a collaborative manner. The model itself is not shared with the individual devices or servers, and only the updated model weights are transmitted back and forth between the devices and the central server.

This means that individual data sources do not have to be shared with the central server, and the model itself does not need to know the specific data points that it is being trained on. Therefore, federated learning enables privacy-preserving machine learning, as the data remains private and secure on the devices or servers that generate it, and the model can still be trained effectively without compromising the privacy of the data. Our proposed federated architecture is explained in figure 2.

In our proposed architecture for the decentralized training of a model, data is first read at the input layer. However, as the current architecture only supports identical data, the images are resized to a fixed shape with specific pixel values to ensure that all images are identical. The dataset is then randomly distributed among clients. A global training loop is created where a copy of the global model is sent to each client, and the local model weight is set to the weight of the global model. The local model is trained, tested and validated with local data. Then the local weights are then collected, and the average of all weights is set to the new weight of the global model and sent back to the local clients. Thus the global training loop is repeated according to the number of epochs. Figure 2 illustrates the overall process of our proposed strategy for training the model in a decentralized manner while ensuring user privacy is protected. Algorithm 1 shows a pseudo-code explaining how we implemented Federated Transfer learning in our system.

Algorithm 1 Federated Learning Algorithm With FedAvg

Require: Pre-trained model M_0
Require: Client dataset $\mathcal{D} = \{D_1, D_2, \ldots, D_n\}$
Require: Categorical cross-entropy loss function L
Require: Stochastic Gradient Descent (SGD) optimizer with learning rate η
Require: Number of communication rounds T
Require: Number of local epochs E
Require: Local mini-batch size B
Ensure: Global model M_{global}

$M_{global} \leftarrow M_0$
for $t = 1$ to T **do**
  for each client $i = 1, \ldots, n$ **in parallel do**
    $M_i \leftarrow M_{global}$
    for $e = 1$ to E **do**
      Shuffle D_i
      for each mini-batch $(x_1, y_1), \ldots, (x_B, y_B)$ in D_i **do**
        $g \leftarrow \frac{1}{B} \sum_{j=1}^{B} \nabla L(M_i(x_j), y_j)$
        Update M_i using SGD: $M_i \leftarrow M_i - \eta g$
      end for
    end for
  end for
  Aggregate models using FedAvg: $M_{global} \leftarrow \frac{1}{n} \sum_{i=1}^{n} M_i^t$
end for

1) ALGORITHM EXPLANATION

The federated learning algorithm we describe here leverages a pre-trained model M_0, a collection of client datasets $\mathcal{D}$, and a categorical cross-entropy loss function L. The optimization is performed using stochastic gradient descent (SGD) with a specified learning rate η. The algorithm proceeds for a fixed number of communication rounds T. In each communication round, clients update their local models using their respective datasets for a specified number of local epochs E and a local mini-batch size B. Here's a step-by-step explanation of the algorithm;

1) Initialize the global model M_{global} with a pre-trained model M_0.
2) Perform a fixed number of communication rounds T to enable collaboration between the server and the clients.
3) For each communication round t, iterate over each client i in parallel.
4) Assign the current global model M_{global} to the local model M_i of each client.

5) Within each client, perform a specified number of local epochs E to update the local model based on its respective dataset D_i.
6) Shuffle the client's dataset D_i to ensure randomness in the mini-batch selection.
7) For each mini-batch in D_i, compute the average gradient g of the categorical cross-entropy loss function L by evaluating the model's predictions $M_i(x_j)$ against the corresponding labels y_j.
8) Update the local model M_i using stochastic gradient descent (SGD) with a learning rate η and the computed gradient g.
9) Repeat steps 4-8 for the specified number of local epochs E to allow the client to optimize its local model using its own dataset.
10) Once all clients have completed their local updates, aggregate their models using Federated Averaging (FedAvg).
11) Apply FedAvg by taking the weighted average of the model parameters across all clients, giving more weight to clients with larger datasets.
12) Update the global model M_{global} with the aggregated model obtained from the FedAvg step.
13) Repeat steps 3-12 for the specified number of communication rounds T to progressively refine the global model.

At the conclusion of the federated learning algorithm, the final global model M_{global} emerges as the culmination of the collaborative knowledge extraction from all participating clients' datasets. By employing this approach, the global model is able to leverage the diverse and expansive information contained within each client's dataset, resulting in a more comprehensive and robust model. Importantly, this collaborative process upholds the crucial principle of data privacy by ensuring that raw data remains confidential and is not shared among clients or with the central server. Thus, the final global model obtained through federated learning offers both enhanced performance and a steadfast commitment to preserving data privacy, making it an appealing and convincing solution for distributed machine learning scenarios.

IV. EXPERIMENT AND RESULT

To attain a better result, we conducted different tests with different Deep Learning models. We trained five different pre-trained models along with CNN to classify the computer activity of students. The dataset used in this experiment contains 4000 screenshots of five categories that are most common in an e-learning scenario. The following experiments are depicted:

A. DATASET

In this research, the dataset by Ferdousi et al. has been used that contains over 4000 screenshots of five categories [18]. The dataset has been created keeping in mind to determine the productivity of students mainly. So the following categories shown in table 4 are chosen which are mostly used by students in particular:

TABLE 4. Description of dataset.

Type of Class	Class	Size
Education Class	Education Coursera	800
	Education Google Classroom	799
	Education Programming	800
	Education YouTube	812
Entertainment Class	Entertainment YouTube	802

These classes are particularly chosen because, in an online learning scenario, students may use their computers mostly in google classroom, doing programming, learning using online learning platforms like Coursera [33], or learning from YouTube tutorials. While learning from YouTube tutorials, it is very easy to get distracted by suggested videos and start watching entertaining videos which is counterproductive and trigger a wastage of valuable time. For that reason, all classes except "Entertainment YouTube" count as productive, therefore if the captured screenshot is predicted as any of these four classes the time will be registered as productive in the log, and if the captured screenshot is predicted as Entertainment YouTube class the time will be registered as unproductive in the log. An example screenshot of each class from the dataset is described in figure 3.

Figure 3(a) Education Coursera: The figure showcases an image from a Coursera lesson, which is an online platform providing web-based courses to enhance skills.

Figure 3(b) Education Google Classroom: The figure displays a screenshot taken from Google Classroom, which is a regular activity in an e-learning scenario.

Figure 3(c) Education Programming: The figure shows an image of programming, which is a common learning activity of computer science students.

Figure 3(d) Educational YouTube: The figure highlights an image captured from an Educational YouTube lesson, demonstrating how YouTube can be utilized for learning purposes.

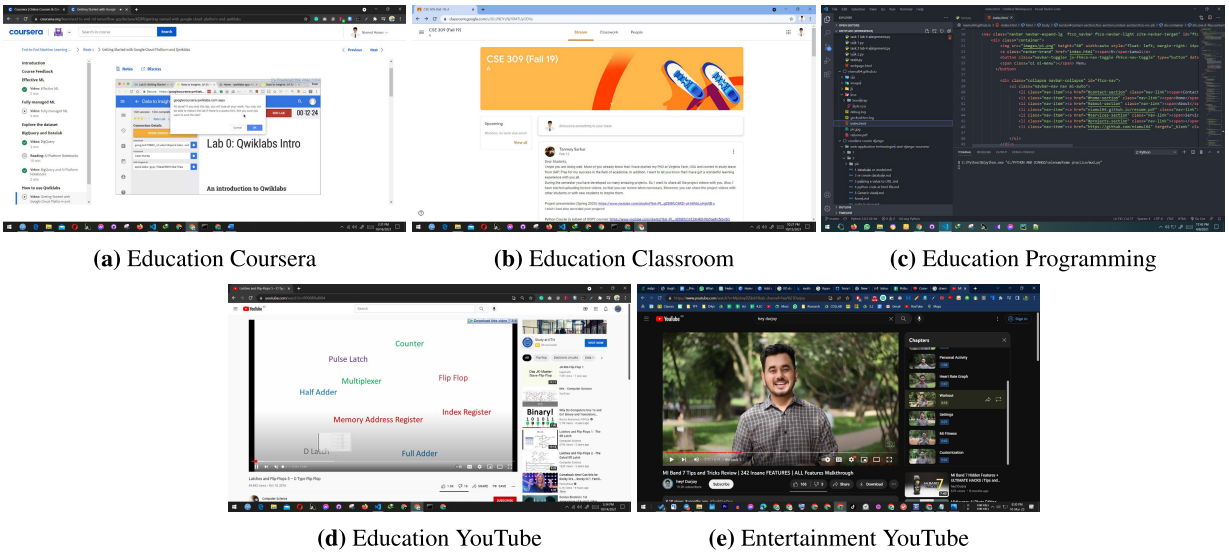
Figure 3(e) Entertainment YouTube: The figure illustrates a snapshot from an Entertainment YouTube video, indicating how YouTube can also be used for leisure activities like watching news, cartoons, or music videos.

B. EXPERIMENT SETUP

The models were implemented using the no-cost version of Google Colaboratory, a cloud-based platform that grants free access to GPU and TPU resources for training machine learning models. The models were implemented utilizing popular machine learning frameworks such as TensorFlow, Keras, and Scikit-learn, with Python being the primary programming language used.

C. EVALUATION METRICS

We consider a number of performance metrics to assess our model. We tested the dataset against several pre-trained deep



(a) Education Coursera **(b)** Education Classroom **(c)** Education Programming

(d) Education YouTube **(e)** Entertainment YouTube

FIGURE 3. Example screenshot of each class from the dataset.

learning models inside a federated environment. To evaluate pre-trained models, we randomly selected 80% samples for training and 20% samples for testing purposes. The model suggested is assessed by measuring its performance across several evaluation metrics, including accuracy (A), precision (P), recall (R), and F1-score (F).

1) ACCURACY

The accuracy score in classification, also commonly referred to as the classification accuracy rating, is determined by dividing the number of correct predictions by the total number of predictions made. This can be represented mathematically by the formula provided in Equation (1).

$$A = \frac{TruePositive + TrueNegative}{TotalNumberofPredictions} \tag{1}$$

2) PRECISION

Precision is a metric that calculates the proportion of true positive results over the total number of positive outcomes, which includes both correctly and incorrectly identified cases (P). To compute precision, Equation (2) is employed.

$$P = \frac{TruePositive}{TruePositive + FalsePositive} \tag{2}$$

3) RECALL

Recall is a metric that quantifies the ratio of true positive results to the total number of positive cases that should have been identified. To calculate the recall score, Equation (3) is utilized.

$$R = \frac{TruePositive}{TruePositive + FalseNegative} \tag{3}$$

4) F1-SCORE

The F1-score is a performance measure that evaluates the accuracy of a model in each class. It is often used in datasets with imbalanced class distributions. To demonstrate the effectiveness of our proposed method, we adopt the F1-score as an evaluation metric. The F1-score is computed using Equation (4).

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{4}$$

D. PERFORMANCE EVALUATION

In this study, the performance of six different federated models for detecting the on-screen activity of students was evaluated using a dataset of over 4000 screenshots divided into five classes. The performance is evaluated with a number of criteria which include the analysis of performance matrices in sub-section IV-D1, loss and accuracy are discussed in sub-section IV-D2, hyper-parameter tuning is analyzed in sub-section IV-D3 and the discussion ends with confusion matrix analysis at subsection IV-D5.

1) PERFORMANCE MATRICES ANALYSIS

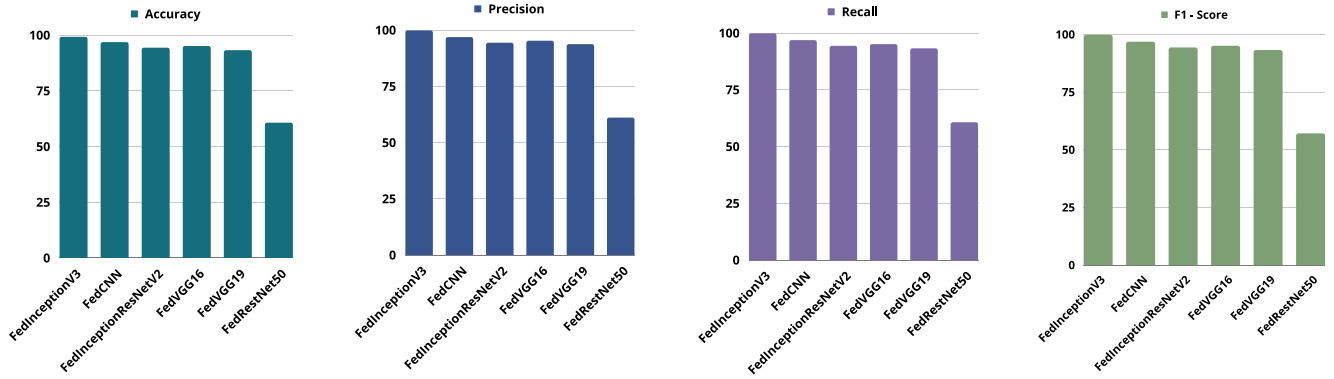
The evaluated models include InceptionV3 [34], CNN [35], InceptionResNetV2 [36], VGG16 [37], VGG19 [38], and RestNet50 [39]. The evaluation matrices of these models are shown in table 5.

Table 5 represents the performance metrics of all six federated models implemented in this research. Among all the evaluated models, FedInceptionV3 achieved the highest test accuracy of 99.75%, followed by FedCNN with 96.76%, FedVGG16 with 95.01%, FedVGG19 with 93.14%, FedInceptionResNetV2 with 94.26%. Although the achieved accuracy scores are acceptable and surpass those of the current research, FedRestNet50 exhibited a disappointing performance with an accuracy score of only 60.6%.

Regarding the precision metric, we used the macro precision metric, which is calculated by taking the average precision score for each class. The macro precision calculates

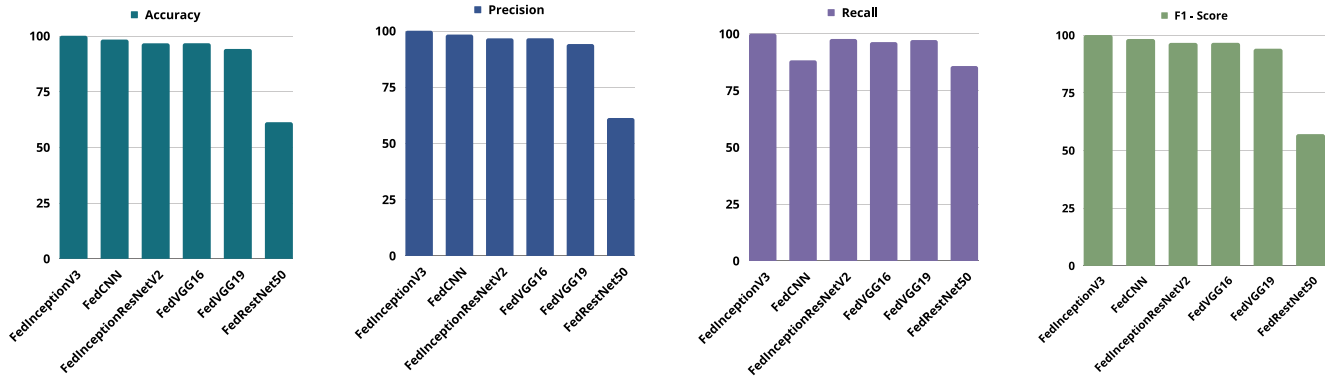
TABLE 5. Federated models' performance matrices on test and train data.

Federated Model	Test Data				Train data			
	Accuracy	Precisoin	Recall	F1- Score	Accuracy	Precisoin	Recall	F1- Score
FedInceptionV3	99.75	99.75	99.75	99.75	100	100	100	100
FedCNN	96.76	96.79	96.76	96.77	98.29	98.31	98.29	98.3
FedInceptionResNetV2	94.26	94.29	94.26	94.27	96.61	96.64	96.61	96.62
FedVGG16	95.01	95.17	95.01	95.09	96.64	96.69	96.64	96.66
FedVGG19	93.14	93.65	93.14	93.39	94.08	94.3	94.08	94.19
FedRestNet50	60.6	61	60.6	60.8	61.2	61.28	61.2	61.24



(a) Accuracy (b) Precision (c) Recall (d) F1-Score

FIGURE 4. Graphical representation of performance matrices on test dataset.



(a) Accuracy (b) Precision (c) Recall (d) F1-Score

FIGURE 5. Graphical representation of performance matrices on train dataset.

the precision independently for each class and then takes the unweighted average of these scores. Similar trade can be observed here, all models performed well except FedResNet50 with values ranging from 61.00% for FedRestNet50 to 99.75% for FedInceptionV3. Similarly, the weighted recall metric, matched the precision trend with FedInceptionV3 achieving the highest value of 99.75% and FedRestNet50 achieving the lowest value of 60.60%.

The F1-score metric, which considers both precision and recall, showed that FedInceptionV3 achieved the highest value of 99.75%, followed by FedCNN with 96.76%, FedVGG16 with 95.01%, FedInceptionResNetV2 with 94.26%, FedVGG19 with 93.14%, and FedRestNet50 with 57.00%. An identical pattern can be observed at the performance matrices on the training dataset with slightly higher scores compared to matrices on the test dataset. Graphical representation of these statistics are represented with bar charts in figure 4 and 5.

2) LOSS AND ACCURACY ANALYSIS

Loss and accuracy graphs are important tools for evaluating model performance in machine learning research. Loss graphs track the error between predicted and actual values during training, while accuracy graphs measure the proportion of correctly predicted values. Analyzing these graphs helps to assess model learning progress, identify overfitting or underfitting issues, and understand the trade-off between model complexity and generalization. Comparison of loss and accuracy graphs can guide model selection, hyperparameter tuning, and model improvement strategies.

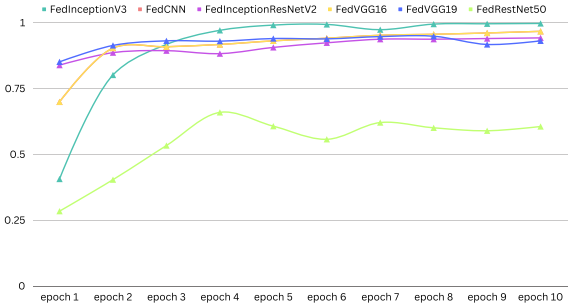


FIGURE 6. Test accuracy graph in 10 epochs.

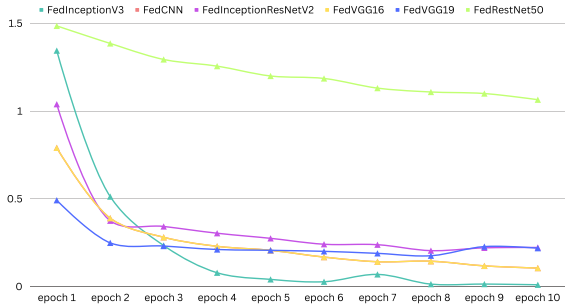


FIGURE 7. Test loss graph in 10 epoch.

These graphs provide visual representations of model performance and play a crucial role in assessing and improving the effectiveness of machine-learning models in research settings.

Figure 6 shows the accuracy values of different machine learning models over epochs during training. The x-axis represents the epochs, ranging from 1 to 10, and the y-axis represents the accuracy values, ranging from 0 to 1. The lines on the graph represent the accuracy trends of each model, with different colours indicating different models. Overall, the graph shows that the models FedInceptionV3 consistently perform better in terms of accuracy compared to the other models, reaching near-perfect accuracy (close to 1) by the later epochs. FedCNN and FedVGG19 also show relatively high accuracy values but with some fluctuations during training. Every model except FedResNet50 performs exceptionally well in terms of accuracy.

Similarly, figure 7 shows the loss values of different machine learning models over epochs during training. The x-axis denotes the epochs, ranging from 1 to 10, while the y-axis represents the corresponding loss values, ranging from 0 to 1. The graph portrays different lines with distinct colours, indicating the performance trends of various models. Notably, FedInceptionV3 exhibits superior performance in terms of loss, consistently achieving lower values (close to 0) in later epochs. It is noteworthy that all models, except FedResNet50, achieve a commendable loss score.

3) HYPER-PARAMETER TURNING

Hyperparameter tuning is an essential process in machine learning that involves selecting the optimal combination of hyperparameters for a given algorithm or model. We have

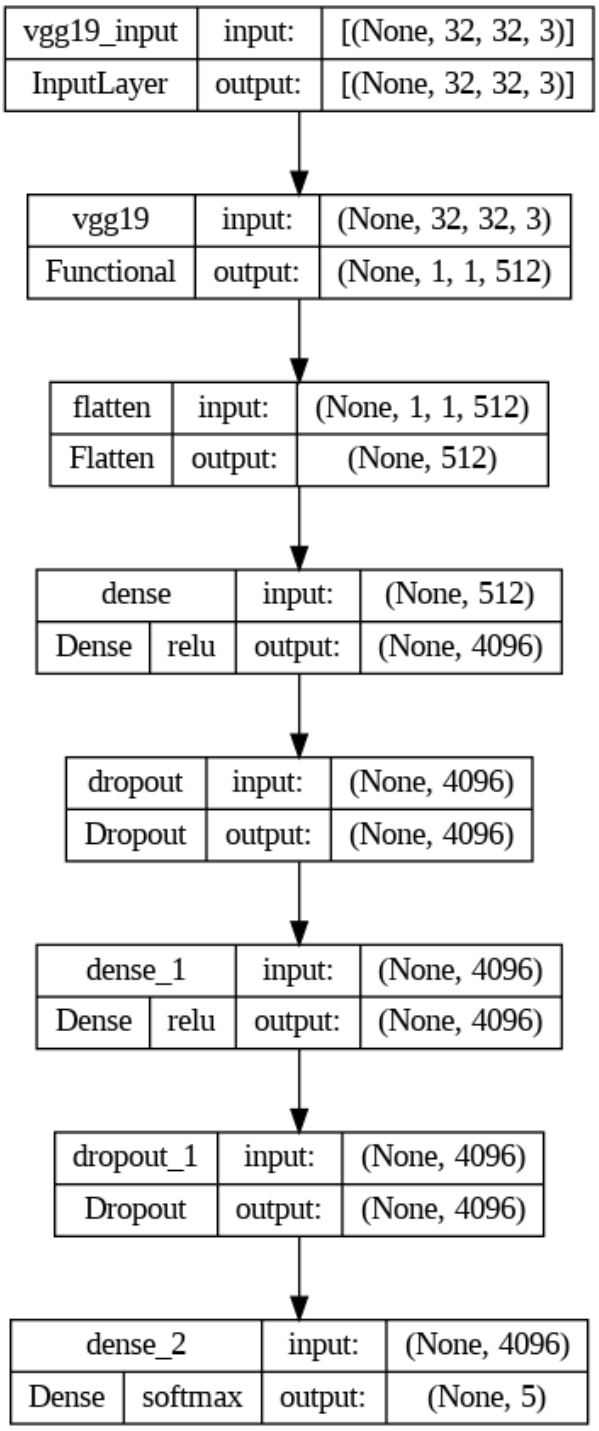


FIGURE 8. Layer by layer model description of FedVGG19.

adjusted a range of hyperparameters to achieve optimal performance for the models. Table 6 includes the name of the federated model, learning rate, optimizer, number of epochs, batch size, number of clients, dropout rate, activation functions, loss function, and image size.

In this study, various hyperparameters were carefully selected to optimize the performance of federated models. The learning rate is an important hyper-parameter that

TABLE 6. Hyper-parameters of evaluated models.

Federated Model	Learning Rate	Optimizer	Number of Epochs	Batch Size	Number of Clients	Dropout Rate	Activation Functions	Loss Function	Input Image Shape
FedInceptionV3	0.001	SGD	10	32	3	0.5	Softmax	CE	75,75
FedCNN	0.001	SGD	10	32	3	0.5	Softmax	CE	32,32
FedInceptionResNetV2	0.001	SGD	10	32	3	0.5	Softmax	CE	75.75
FedVGG16	0.001	Adam	10	32	3	0.5	Softmax	CE	32,32
FedVGG19	0.001	Adam	10	32	3	0.5	Softmax	CE	32,32
FedRestNet50	0.001	SGD	10	32	3	0.5	Softmax	CE	32,32

TABLE 7. Comparison of computational complexity.

Model	Image Size (Pixel)	User Time	Sys Time	Wall Time	Total Time	Epochs	Avg/Sample
FedInceptionV3	75x75	4h 46min 2s	37min 15s	3h 58min 15s	5h 23min 18s	10	0.4850s
FedCNN	32x32	24min 46s	2min 22s	20min 6s	27min 9s	10	0.0407s
FedInceptionResNetV2	75x75	3h 29min 13s	22min 47s	3h 25min 33s	3h 52min 1s	10	0.3480s
FedVGG16	32x32	2min 22s	51.3s	3min 30s	3min 14s	10	0.0049s
FedVGG19	32x32	2min 50s	34s	3min 44s	3min 24s	10	0.0051s
FedRestNet50	32x32	34min 19s	11min	42min 18s	45min 20s	10	0.0680s

TABLE 8. Compression of performance on test dataset regarding accuracy, precision, recall and F1-score.

Ref	Method	Dataset	Privacy	Accuracy	Precision	Recall	F1 Score
Ferdousi et al. [18]	ResNet50	Ferdousi et al. [18]	x	95.40%	95%	95%	95%
Ferdousi et al. [18]	VGG16	Ferdousi et al. [18]	x	94%	94%	94%	94%
Ferdousi et al. [18]	InceptionV3	Ferdousi et al. [18]	x	82%	85%	82%	82%
Proposed	**FedInceptionV3**	**Ferdousi et al. [18]**	✓	**99.75%**	**99.75%**	**99.75%**	**99.75%**

controls how quickly the model adjusts its parameters in response to the estimated error gradient. A learning rate of 0.001 was chosen, as it has been found to be effective in many deep-learning applications. Additionally, to ensure a fair comparison between models, the same number of epochs, batch size, and number of clients were used for all models. The number of epochs refers to the number of times the entire training dataset is shown to the model. The batch size is the number of training examples used in one iteration. The number of clients refers to the number of devices participating in the federated learning process. The dropout rate, a regularization technique used to prevent overfitting is set to 0.5 for each model.

Both Adam and SGD optimizers were found to perform well with all models. The optimizer is responsible for adjusting the model's parameters to minimize the loss function during training. Adam optimizer, which is a variant of Stochastic Gradient Descent (SGD), is known for its adaptive learning rate and momentum. Adam was found to be particularly effective in the FedVGG16 and FedVGG19 models, while SGD was more successful in other models.

Activation functions play a critical role in deep learning models by introducing non-linearity into the model. In this study, the ReLU activation function was applied to the hidden layers, which has been shown to be effective in many deep learning applications. The output layer was assigned the Softmax activation function, as the dataset was multi-class, and Softmax is an effective activation function for multi-class classification problems.

Categorical Cross-entropy (CE) was selected as the loss function for all models. Categorical Cross-entropy is a popular loss function for multi-class classification problems, and it was chosen in this study due to the multi-class nature of the dataset. Finally, the input image shape was adjusted according to resources and model parameters to achieve faster training speed and higher accuracy.

To enhance the replicability of the proposed method, in addition to providing the hyperparameters, we have included a detailed layer-by-layer representation of FedVGG19 in Figure 8. This depiction provides a comprehensive understanding of the architecture, allowing for easier replication and implementation of the model.

4) COMPUTATIONAL COMPLEXITY

In the realm of analyzing computational complexity and comparing the performance of models, training time serves as a crucial metric. In our case, where models were trained using the freely available resources of Google Colaboratory, a thoughtful decision was made to strike a balance between the number of epochs and resource limitations. To ensure an accurate assessment, we relied on the total training time, which encompasses User time, Systime, and Wall time.

- User time represents the CPU time consumed by a program or process during the execution of its instructions in user mode. It captures the core essence of the application's computational workload.
- Systime, on the other hand, accounts for the CPU time used in executing code in system or kernel

mode, covering system calls and other privileged operations.
- Wall time, or real-time, accurately measures the complete elapsed time, encompassing all aspects from execution time to system calls, I/O operations, and any potential delays.

As shown in Table 7 our analysis reveals that models such as FedInceptionV3 and FedInceptionResNetV2 exhibit comparatively higher training times due to the restricted image size they are designed to handle. However, it is crucial to emphasize that even the highest average/sample training time, at a mere 0.4850 seconds, unequivocally confirms that our proposed system is exceedingly well-suited for real-life scenarios. One of the key strengths of our proposed system lies in the fact that the detection process occurs entirely on the local device itself, eliminating the need for sample uploads to a remote server. Moreover, considering that screenshots are captured at intervals of one minute, the system places an insignificantly negligible burden on the end device. These findings collectively solidify the convincing argument that our proposed system not only delivers reliable and accurate results but does so efficiently, without imposing any noticeable strain on the user's device.

5) CONFUSION MATRIX ANALYSIS

The confusion matrix, alternatively referred to as the error matrix, is a fundamental tool in machine learning for assessing the efficacy of classification models. It offers a holistic overview of a model's predictions in relation to the actual labels of the data. By revealing the various types of errors made by the model, including false positives, false negatives, true positives, and true negatives, the confusion matrix serves as a crucial resource for evaluating the accuracy and performance of the classification model.

In Figure 9, the confusion matrix is depicted, providing insights into the performance of various models on the test data. It is evident that FedInceptionV3 exhibits exceptional performance, achieving near-perfect accuracy in correctly classifying the samples. On the other hand, the other models also demonstrate commendable performance in terms of classification accuracy, with some variations observed. Notably, FedResNet50 appears to have comparatively lower performance compared to the other models. These findings highlight the effectiveness of FedInceptionV3 and the other models in accurately classifying the test data, with FedResNet50 being the least-performing model among them.

E. BENCHMARK

Table 8 presents a benchmark analysis of the performance on the test dataset, comparing the proposed method with three state-of-the-art approaches in terms of accuracy, precision, recall, and F1-score. The evaluation was conducted considering privacy concerns.

Ferdousi et al. conducted a comprehensive analysis on their dataset, evaluating the performance of different models [18]. ResNet50 achieved the highest accuracy of 95.40%, with consistent precision, recall, and F1-score values of 0.95. VGG16 achieved an accuracy of 94%, while InceptionV3 obtained an accuracy of 82% with precision, recall, and F1-score values of 0.85, 0.82, and 0.82, respectively. However, our proposed method, FedInceptionV3, achieved superior performance with an accuracy of 99.75%. Furthermore, it demonstrated exceptional precision, recall, and F1-score values, all measuring at 99.75%.

Based on this benchmark analysis, it is evident that the proposed method outperformed the three state-of-the-art approaches in terms of accuracy, precision, recall, and F1-score, showcasing its potential in the given dataset while considering privacy concerns.

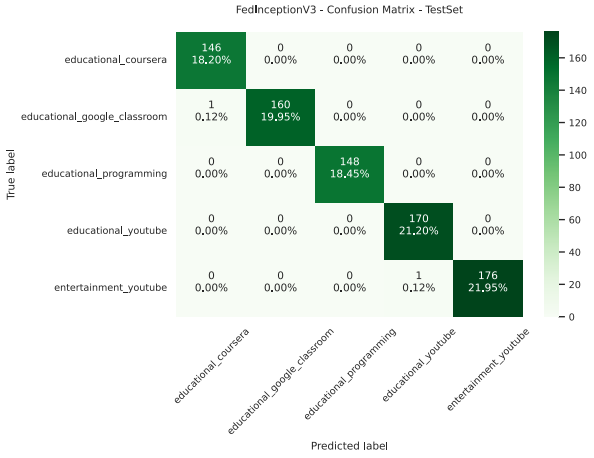
V. DISCUSSION AND PREDICTION ANALYSIS

In this section, we examine and classify the wrong predictions made by our model, while providing insights into the underlying reasons behind these misclassifications. As FedInceptionV3 achieved near-perfect test accuracy we will utilize the FedVGG16 to discuss about the wrong predictions. FedVGG16 in our system processes images in RGB format with a fixed dimension of 32×32 pixels. Within the subsequent subsections (namely subsections V-A, V-B, and V-C), we delve into three distinct types of wrong prediction scenarios, presenting relevant examples to illustrate each case. By thoroughly analyzing these misclassifications, we gain a valuable understanding of the system's limitations and challenges.

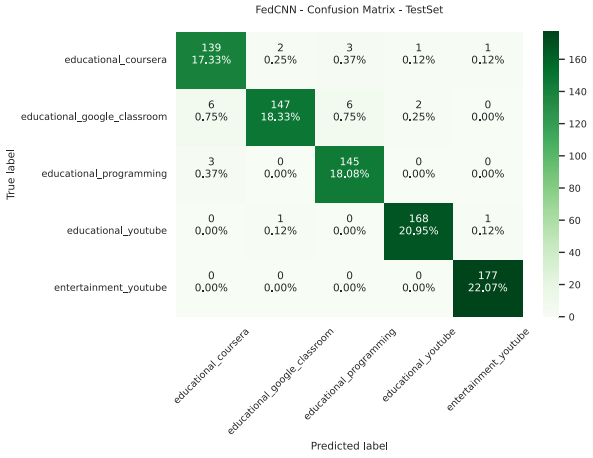
A. HARMLESS WRONG PREDICTION

As stated in subsection IV-A of the paper, all classes, except for "Entertainment YouTube," are considered productive. Therefore, if any of the productive classes, namely "Education Coursera," "Education Google Classroom," "Education Programming," or "Education YouTube," are predicted incorrectly but not classified as "Entertainment YouTube," there will be no adverse effects on the user. In such cases, the system will remain silent, and the time will still be accounted for as productive.

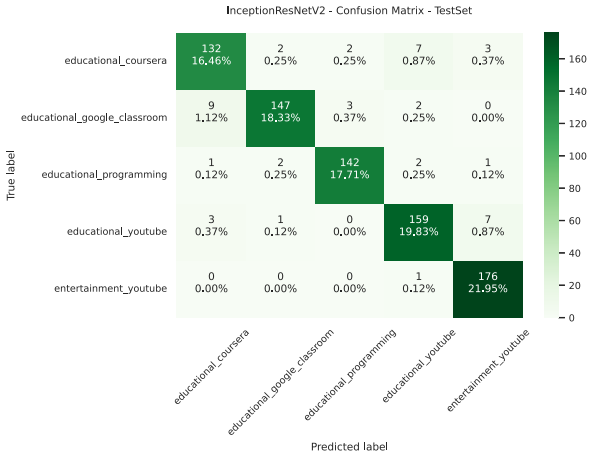
In Figure 10, we present an example that demonstrates a typical misclassification scenario. The image, originally belonging to the "Education Coursera" class, has been incorrectly predicted as "Education Google Classroom." This misclassification can be attributed to the striking similarity in the user interface of both websites, as they predominantly feature white content. The example closely resembles the homepage of Google Classroom, where the black shapes resemble different classes within a classroom homepage shown in Figure 10(c). The deceptive visual resemblance between the two interfaces can be observed in Figure 10(b) and Figure 10(d) contributing to this misclassification, highlighting the challenges in accurately distinguishing between them.



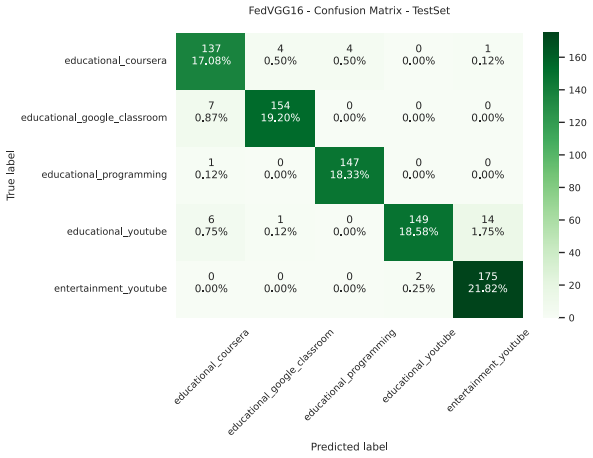
(a) FedInceptionV3



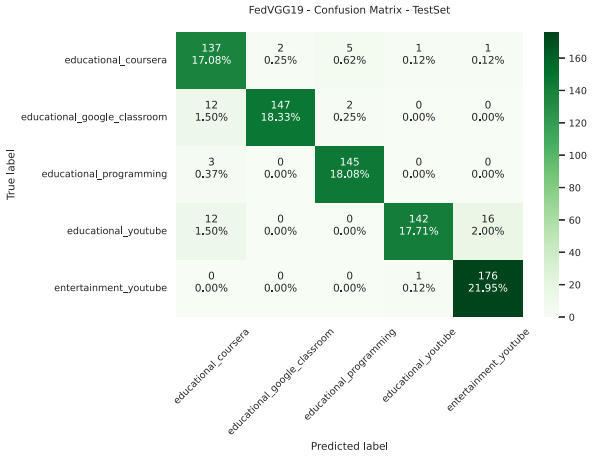
(b) FedCNN



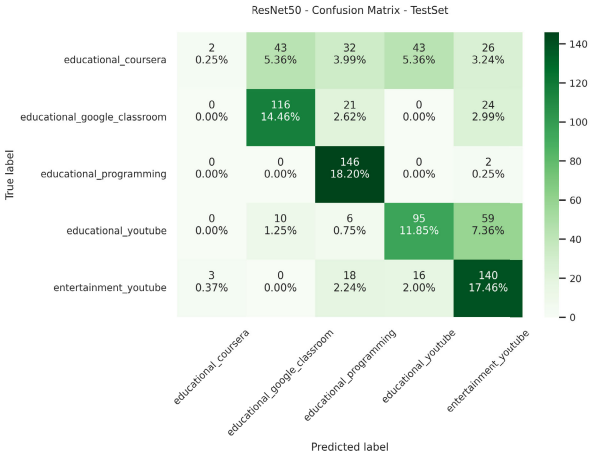
(e) FedVGG19



(e) FedVGG19



(e) FedVGG19

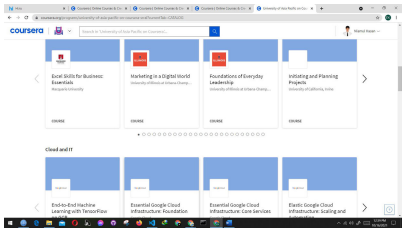


(f) FedResNet50

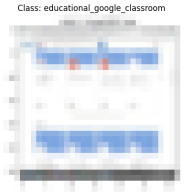
FIGURE 9. **Confusion matrix of all models using test dataset.**

B. MODERATE-IMPACT WRONG PREDICTION

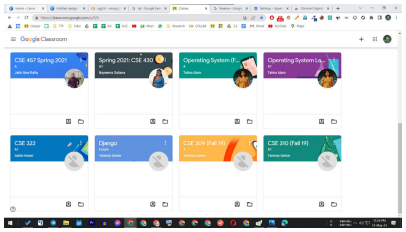
In contrast to harmless wrong predictions, if any of the productive classes are mistakenly classified as unproductive, the system will raise a false alarm. Although this does not inherently pose any issues to the user, it can slightly disrupt the user's concentration. However, it is important to note that the overall impact on the user's productivity remains minimal, and the primary consequence is a temporary interruption in focus caused by the false alarm notification.



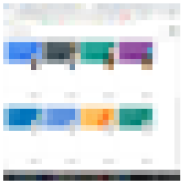
(a) True Class: Education Coursera(Original Image)



(b) Predicted Class: Education Google Classroom(32x32 image)

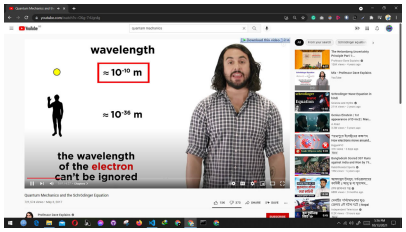


(c) Example Image of Education Google Classroom class(Original Image)

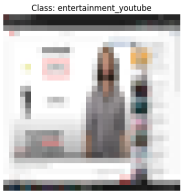


(d) Resized Image of Education Google Classroom Class(32x32 image)

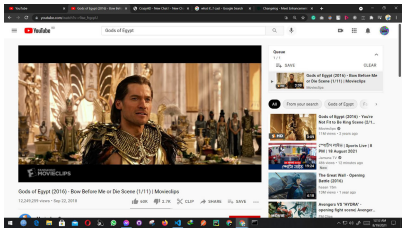
FIGURE 10. Harmless wrong prediction example.



(a) True Class: Education YouTube(Original Image)



(b) Predicted Class: Entertainment YouTube(32x32 image)

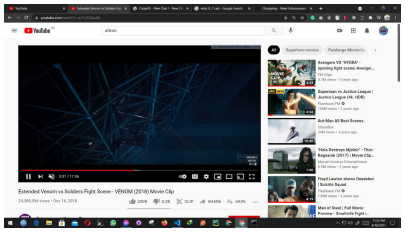


(c) Example Image of Entertainment YouTube Class (Original Image)

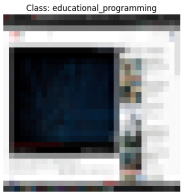
(d) Resized Image of Entertainment YouTube Class (32x32 image)

FIGURE 11. Moderate impact wrong prediction example.

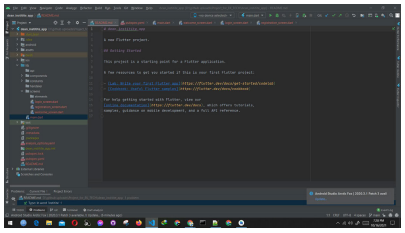
In Figure 11, we observe an example of a moderate impact wrong prediction. In Figure 11(b) educational YouTube tutorial has been misclassified as belonging to the "Entertainment YouTube" class. The misclassification can be attributed to the characteristic trend of entertaining frames, which typically exhibit a wider range of colors compared to predominantly black and white frames. In this specific instance, the inclusion of human skin, the red box, along with the vivid hues present in the first recommended video, combine to produce a more diverse color range within the frame. This feature is reminiscent of the resized depiction of Figure 11(c), which is a characteristic illustration of the "Entertainment YouTube" class. The similarity can further be observed at the 32 × 32 version of both the images in Figure 11(b) and 11(d) which were originally used to train the model. Consequently, the model erroneously classifies the image as "Entertainment YouTube" due to the abundance of colors, rather than correctly identifying it as an educational tutorial.



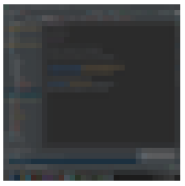
(a) True Class: Entertainment YouTube(Original Image)



(b) Predicted Class: Education Programming(32x32 image)



(c) Example Image of Education Programming Class(Original Image)



(d) Resized Image of Education Programming Class(32x32 image)

FIGURE 12. High impact wrong prediction example.

C. HIGH-IMPACT WRONG PREDICTION

The high-impact wrong prediction scenario of particular concern arises when the "Entertainment YouTube" class is mistakenly classified as a productive class. This presents a significant problem as it undermines the system's primary objective. In this scenario, unproductive activities go undetected by the system, resulting in a failure to accurately track and manage productive time. Consequently, users may engage in unproductive tasks without receiving the necessary notifications or interventions.

In Figure 12, we can observe an example of a high-impact wrong prediction. An entertaining YouTube video has been misclassified as belonging to the "Education YouTube" class. This misclassification can be attributed to the typical trend of productive frames, which often exhibit a black-and-white colour palette, in contrast to the wide range of colors seen in entertaining frames. In this specific example, the dark scene depicted in Figure 12(a) contributes to a predominantly black colour palette within the frame, resembling programming screenshots in dark mode as displayed in 12(c). A stronger similarity between the images used in model training can be observed in their 32 × 32 versions, as shown in Figure 12(b) and 12(d). Therefore, the model erroneously classifies the image as "Educational Programming" due to the absence of vibrant colours, rather than correctly identifying it as an "Entertainment YouTube" frame.

VI. CONCLUSION AND FUTURE WORK

The proposed architecture for detecting the on-screen activity of students shows promising results and opens avenues for further improvement. To enhance its capabilities, there are certain aspects that need attention. For instance, expanding the dataset used in the study to include a more diverse range of devices such as laptops, tablets, and smartphones will improve the architecture's performance and generalizability [40]. Additionally, accommodating screens of different sizes, including multi-screen setups and mobile devices, will enhance its applicability to various real-world scenarios. Furthermore, ensuring secure communication between local and global models through the implementation of measures like secure encryption and authentication protocols will bolster the architecture's overall robustness. Moreover, the proposed architecture holds potential beyond online education, with possible applications in fields like monitoring employee productivity in organizations. Overall, the proposed architecture presents a significant step forward in addressing the challenges of online education. Additionally, the proposed architecture has potential applications beyond online education, such as monitoring employee productivity in organizations. In the future, we plan to expand the architecture to work according to screens of different sizes and explore its applicability in different contexts beyond online education.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deputyship for Research and Innovation, ''Ministry of Education'' in Saudi Arabia for funding this research (IFKSUOR3–013-1).

REFERENCES

[1] M. Xu, J. M. David, and S. H. Kim, ''The fourth industrial revolution: Opportunities and challenges,'' *Int. J. Financial Res.*, vol. 9, no. 2, pp. 90–95, 2018.
[2] A. Sun and X. Chen, ''Online education and its effective practice: A research review,'' *J. Inf. Technol. Educ., Res.*, vol. 15, pp. 157–190, May 2016.
[3] I. E. Allen and J. Seaman, ''Digital learning compass: Distance education enrollment report,'' Babson Survey Res. Group, Wellesley, MA, USA, Tech. Rep. 2017-1, 2017.
[4] S. I. U. Rehman, H. S. Ullah, and A. Akhtar, ''Consumption of social media and academic performance: A cross-sectional survey of perception of students in KP universities,'' *Global Mass Commun. Rev.*, vol. 4, pp. 57–71, 2020. [Online]. Available: https://www.researchgate.net/publication/349955120_Consumption_of_Social_Media_and_Academic_Performance_A_Cross-Sectional_Survey_of_Perception_of_Students_in_KP_universities
[5] I. K. Sohail and N. A. Nabaz, ''The influence of social media on student's academic performance: A case study of Lebanese French University,'' *Mod. Manag. Theory Pract.*, vol. 25, no. 2, pp. 117–127, 2019.
[6] J. Goet, ''Impact of social media on academic performance of students,'' *KIC Int. J. Social Sci. Manage.*, vol. 1, no. 1, pp. 35–42, Dec. 2022.
[7] S. I. U. Rehman, H. S. Ullah, and A. Akhtar, ''Consumption of social media and academic performance: A cross-sectional survey of perception of students in KP universities,'' *Global Mass Commun. Rev.*, vol. 5, no. 4, pp. 57–71, Dec. 2020.
[8] C. Li and F. Lalani, *The COVID-19 Pandemic has Changed Education Forever. This is How*, vol. 29. Geneva, Switzerland: World Economic Forum, 2020.
[9] A. Aristovnik, D. Keržič, D. Ravšelj, N. Tomaževič, and L. Umek, ''Impacts of the COVID-19 pandemic on life of higher education students: A global perspective,'' *Sustainability*, vol. 12, no. 20, p. 8438, Oct. 2020.
[10] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, ''Deep learning with differential privacy,'' in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.* New York, NY, USA: ACM, 2016, pp. 308–318.
[11] W. Dai, S. Wang, H. Xiong, and X. Jiang, ''Privacy preserving federated big data analysis,'' in *Guide to Big Data Applications*. 2018, pp. 49–82.
[12] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, ''Communication-efficient learning of deep networks from decentralized data,'' 2016, *arXiv:1602.05629*.
[13] S. J. Pan and Q. Yang, ''A survey on transfer learning,'' *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2009.
[14] K. R. Varshney, ''Trustworthy machine learning and artificial intelligence,'' *XRDS, Crossroads, ACM Mag. Students*, vol. 25, no. 3, pp. 26–29, Apr. 2019.
[15] E. Vayena, A. Blasimme, and I. G. Cohen, ''Machine learning in medicine: Addressing ethical challenges,'' *PLOS Med.*, vol. 15, no. 11, Nov. 2018, Art. no. e1002689.
[16] B. Imler and M. Eichelberger, ''Using screen capture to study user research behavior,'' *Library Hi Tech*, vol. 29, no. 3, pp. 446–454, 2011.
[17] P. Krieter and A. Breiter, ''Track every move of your students: Log files for Learning Analytics from mobile screen recordings,'' in *Proc. Die 16. E-Learning Fachtagung Informatik (DeLFI)*, 2018, pp. 1–12.
[18] B. J. Ferdosi, M. Sadi, N. Hasan, and M. A. Rahman, ''Tracking digital device utilization from screenshot analysis using deep learning,'' in *Proc. Int. Conf. Data Sci. Appl. (ICDSA)*, vol. 1. Singapore: Springer, 2023, pp. 661–670.
[19] C. Dwork, ''Differential privacy,'' in *Proc. Int. Colloq. Automata, Lang., Program.* Berlin, Germany: Springer, 2006, pp. 1–12.
[20] X. Yi, R. Paulet, E. Bertino, X. Yi, R. Paulet, and E. Bertino, ''Homomorphic encryption,'' in *Tutorials on the Foundations of Cryptography*. Cham, Switzerland: Springer, 2014.
[21] O. Goldreich, ''Secure multi-party computation,'' *Manuscript. Preliminary version*, vol. 78, no. 110, pp. 1–108, 1998.
[22] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, ''Federated Learning: Strategies for improving communication efficiency,'' 2016, *arXiv:1610.05492*.
[23] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, ''Practical secure aggregation for privacy-preserving machine learning,'' in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 1175–1191.
[24] E. Hesamifard, H. Takabi, M. Ghasemi, and R. N. Wright, ''Privacy-preserving machine learning as a service,'' *Proc. Privacy Enhancing Technol.*, vol. 2018, no. 3, pp. 123–142, Jun. 2018.
[25] Hubstaff. *Hubstaff: Time Tracking Software for Remote Teams*. Accessed: May 3, 2023. [Online]. Available: https://hubstaff.com/
[26] Teramind. *Teramind: Employee Monitoring Software*. Accessed: May 3, 2023. [Online]. Available: https://www.teramind.co/
[27] Workpuls. *Workpuls: Time Tracking and Productivity Software*. Accessed: May 3, 2023. [Online]. Available: https://www.workpuls.com/
[28] DeskTime. *DeskTime: Productivity and Time Tracking Software*. Accessed: May 3, 2023. [Online]. Available: https://desktime.com/
[29] Time Doctor. *Time Doctor: Time Tracking and Productivity Software*. Accessed: May 3, 2023. [Online]. Available: https://www.timedoctor.com/
[30] Awareness Technologies. (2023). *InterGuard*. Accessed: May 3, 2023.
[31] ActivTrak. (2023). *ActivTrak: Workforce Analytics and Productivity Software*. Accessed: May 3, 2023. [Online]. Available: https://www.activtrak.com/
[32] D. L. Hankerson, C. Venzke, E. Laird, H. Grant-Chapman, and D. Thakur, ''Online and observed: Student privacy implications of school-issued devices and student activity monitoring software,'' OSF Preprints, Center Democracy Technol. (CDT), USA, Tech. Rep., 2022.
[33] Coursera Inc. (2012). *Coursera*. Accessed: May 8, 2023. [Online]. Available: https://www.coursera.org
[34] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, ''Rethinking the inception architecture for computer vision,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.
[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ''ImageNet classification with deep convolutional neural networks,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[36] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, ''Inception-v4, Inception-ResNet and the impact of residual connections on learning,'' in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2017, vol. 31, no. 1, pp. 4278–4284.
[37] K. Simonyan and A. Zisserman, ''Very deep convolutional networks for large-scale image recognition,'' 2014, *arXiv:1409.1556*.
[38] K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[39] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.
[40] R. L. Brennan, "Generalizability theory," *Educ. Meas., Issues Pract.*, vol. 11, no. 4, pp. 27–34, Dec. 1992.

DURJOY MISTRY received the bachelor's degree in computer science and engineering from the University of Asia Pacific, in 2021. He is currently a Lecturer with the Department of Computer Science, University of Asia Pacific. With a strong research background, his focus lies in the realms of artificial intelligence (AI), machine learning, deep learning, federated learning (FL), and natural language processing (NLP). Previously, he was a Researcher with the Institute of Automation Research and Engineering (IARE). During his tenure at IARE, he actively engaged in pioneering research endeavors, collaborating with fellow experts to explore novel solutions and push the boundaries of AI technology. His dedication to the pursuit of knowledge is reflected in his involvement in various research projects, where he contributed extensively by conducting experiments, analyzing data, and disseminating research findings through publications in reputable conferences and journals. He secured Bronze Award from the International Blockchain Olympiad in 2022. Now, as a Lecturer, he brings his wealth of expertise and passion for AI to the classroom, inspiring and guiding students on their own journeys of intellectual growth and understanding in the dynamic field of computer science.

M. F. MRIDHA (Senior Member, IEEE) received the Ph.D. degree in AI/ML from Jahangirnagar University, in 2017. He is currently an Associate Professor with the Department of Computer Science, American International University-Bangladesh (AIUB). Before that, he was an Associate Professor and the Chairperson of the Department of CSE, Bangladesh University of Business and Technology. He was a Faculty Member with the CSE Department, University of Asia Pacific, and the Graduate Head, from 2012 to 2019. His research experience within both academia and industry, which results in over 120 journals and conference publications. His research work contributed to the reputed journals of *Scientific Reports* (Nature), *Knowledge-Based Systems*, *Artificial Intelligence Review*, IEEE Access, *Sensors*, *Cancers*, and *Applied Sciences*. For more than ten years, he has been with the master's and bachelor's students as a supervisor of their thesis work. His research interests include artificial intelligence (AI), machine learning, deep learning, natural language processing (NLP), and big data analysis. He was a program committee member of several international conferences/workshops. He served as an Associate Editor for several journals, including *PLOS One* journal. He also served as a Reviewer for reputed journals and IEEE conferences, such as HONET, ICIEV, ICCIT, IJCCI, ICAEE, ICCAIE, ICSIPA, SCORED, ISIEA, APACE, ICOS, ISCAIE, BEIAC, ISWTA, IC3e, ISWTA, CoAST, icIVPR, ICSCT, 3ICT, and DATA21.

MEJDL SAFRAN received the M.Sc. and Ph.D. degrees in computer science from Southern Illinois University, Carbondale, in 2013 and 2018, respectively. He has been a Faculty Member, since 2007. He is an Assistant Professor of computer science with King Saud University. His research interests include computational intelligence, artificial intelligence, deep learning, pattern recognition, and predictive analytics. He has published articles in refereed journals and conference proceedings, such as *ACM Transactions on Information Systems*, *Applied Computing and Informatics*, *Biomedicine* (MDPI), *Sensors* (MDPI), IEEE International Conference on Cluster, IEEE International Conference on Computer and Information Science, International Conference on Database Systems for Advanced Applications, and International Conference on Computational Science and Computational Intelligence. His current research focus includes developing efficient recommendation algorithms for large-scale systems, predictive models for online human activities, machine learning algorithms for performance management, and modeling and analyzing user behavior. Since 2018, he has been providing part-time consulting services in the field of artificial intelligence to private and public organizations and firms.

SULTAN ALFARHOOD received the Ph.D. degree in computer science from the University of Arkansas. He is an Assistant Professor with the Department of Computer Science, King Saud University (KSU). Since joining KSU in 2007, he has made several contributions to the field of computer science through his research and publications. He has published several research articles on cutting-edge topics, such as machine learning, recommender systems, linked open data, and text mining. His work includes proposing innovative approaches and techniques to enhance the accuracy and effectiveness of recommender systems and sentiment analysis.

ALOKE KUMAR SAHA received the B.Sc. degree (Hons.) in applied physics amp and electronics and the M.Sc. degee (thesis) in computer science from the University of Dhaka, in 1995 and 1997, respectively, and the Ph.D. degree in computer science and engineering from Jahangirnagar University, Savar, Dhaka, Bangladesh. He was a Lecturer with Queens University, from June 1997 to March 1999. In March 1999, he joined the University of Asia Pacific (UAP), Dhaka, as a Lecturer. He is a Professor with the Computer Science and Engineering (CSE) Department and the Director of the Institutional Quality Assurance Cell (IQAC), UAP. He has authored or coauthored 32 journal articles and 25 conference papers. He usually teaches courses on digital logic amp, system design, numerical methods, data structures, discrete mathematics, and computer graphics. His current research interests are algorithms, artificial intelligence (AI), machine learning, and natural language processing (NLP). For more than 26 years, he is working with undergraduate and graduate students as a course teacher as well as supervisor of their research works. He was the Head of the CSE Department, UAP, from 2008 to 2018. He was the Chair of the Organizing Committee of the International Conference on Computer and Information Technology (ICCIT), in 2017. He was the Contest Director of the National Collegiate Programming Contest (NCPC), in 2016. Under his leadership, UAP host the International Collegiate Programming Contest (ICPC), in 2016 and 2017. He is the Chief of the Organizing Committee of *International Journal of Computer and Information Technology* (IJCIT) (Department of CSE, UAP). He is a reviewer of different conferences and journals.

DUNREN CHE received the B.S. degree in electronic engineering from the Harbin University of Commerce, China, in 1985, the M.S. degree in computer science from the National University of Defense Technology, in 1988, and the Ph.D. degree in computer science from the Beijing University of Aeronautics and Astronautics, in 1994. In 2001, he was a Postdoctoral Research Fellow with Tsinghua University, the German National Research Center for Information Technology, and Johns Hopkins University. He is currently a Professor of computer science with Southern Illinois University (SIU), Carbondale. He was the Director of the Undergraduate Computer Science Programs with the School of Computing, SIU, from 2013 to 2022. His main research interests are database, data mining, machine learning (collectively data science), cloud computing, and scientific workflow. He has authored/coauthored more than 120 peer-reviewed articles published in various venues, such as *VLDB Journal*, *Future Generation Computer Systems*, and various ACM/IEEE transactions and associated conferences.