A publication of ISCA*:
International Society for Computers
and Their Applications

INTERNATIONAL JOURNAL OF COMPUTERS AND THEIR APPLICATIONS

TABLE OF CONTENTS

***"International Journal of Computers and Their Applications is Peer Reviewed".**

Volume 29, No. 3, Sept 2022 **ISSN 1076-5204**

International Journal of Computers and Their Applications

A publication of the International Society for Computers and Their Applications

ISCA Headquaters • 278 Mankato Ave, #220, Winona, Minnesota 55987, USA
Phone 507-458-4517 • Email isca@isca-hq.org • URL: http://www.isca-hq.org

Guest Editorial

This issue of the IJCA is a collection of eight referred papers. Four of which are selected from the 37th International Conference on Computer and Their Applications (CATA 2022). The other four papers in this issue are regular submissions to IJCA.

Each paper submitted to this issue was reviewed judging the originality, technical contribution, significance and quality of presentation. The papers in this issue cover a wide range of research interests in the community of computers and applications. The topics and main contributions of the papers are briey summarized below.

SRUTHI RACHAMALLA and HENRY HEXMOOR of Southern Illinois University, Carbondale, Illinois, USA, wrote their paper "Improving Road Safety by Blockchain-based Monetization of Driver Behavior." In this work, the authors presented a driver incentive model that ranks and rewards the driver's daily behavior to increase the safety of drivers and other road users. The authors also examined the cooperative driving (or platooning) scenario. In this paper, the authors claimed that road safety is improved by connecting two or more cars together by utilizing vehicular communication technologies. The authors proposed that the rewards will come in the form of cryptocurrency tokens and be accomplished by secure transactions using blockchain.

JEFF MCCANN of Dell Technologies, Limerick, IRELAND (also affiliated with University of Limerick, IRELAND), LIAM QUINN of Dell Technologies, Austin, Texas, USA, and SEAN MCGRATH of University of Limerick, IRELAND wrote their paper titled "Video Surveillance architecture from the cloud to the Edge." In the paper, the authors examined the use of digital video in public safety and surveillance systems to determine how human operators are challenged to monitor these data feeds in real time and how the emergence of AI and computer vision solutions can process this data. The authors concluded that using edge computing capabilities, combined with modern coding and management capabilities, can overcome challenges with network latency and enable real-time, preventative surveillance solutions for law enforcement.

MOHAMMAD HOSSAIN of the University of Minnesota Crookston, Minnesota, USA, and HONGKAI CHEN of the University of California, San Diego, USA, wrote a survey paper titled "Application of Machine Learning on Software Quality Assurance and Testing: A Chronological Survey." In the paper, the authors studied and digested papers since 1995 that applied machine learning techniques to solve an issue in software quality assurance and testing. The authors told a story in a chronological manner and pointed out the transition/highlight between each era. This paper would serve as a very good guideline for software engineers who would like to extend the problems involved in SDLC with machine learning techniques.

NARAYAN C. DEBNATH, SHREYA BANERJEE, GIAU UNG VAN, PHAT TAT QUANG, AND DAI NGUYEN THANH from the Department of Software Engineering, Eastern International University, Binh Duong, Vietnam, authored the paper titled "Semantic Reasoning to support End User Development in Intelligent Environment." The authors proposed a semantic-based reasoning framework in this paper to support end users of Internet of Thing (IoT) objects in Intelligent Environments (IE). The proposed framework is based on an upper-level ontology specification named Trigger Action Ontology (TAO). This framework includes a rule-based reasoner implemented in Apache Jena. The authors claimed their proposed framework would assist end users of IE applications on various domains to represent triggers, actions, and their respective combinations.

TIRTHANKAR GHOSH, SIKHA BAGUI, SUBHASH BAGUI, MARTIN KADZIS, LOGAN DAY, and JACKSON BARE of the University of West Florida wrote their paper titled "Univariate and Bivariate Entropy Analysis for Modbus Traffic over TCP/IP in Industrial Control Systems." This paper presented an entropy analysis on an industrial control system network using selected features with datasets obtained from an HVAC system. The authors claimed that their initial analysis showed some promising results using univariate entropy and divergence.

BEVAN CHRISTIAN and TRIANGGORO WIRADINATA of the Universitas Ciputra Surabaya wrote their paper titled "The Implementation of Content Planner Application with MobileNetV2 Classification for Hashtag Automation." This study aimed to design and develop a content planner application with the feature of making Automatic hashtag creation. The authors created an application that helps content developers manage social media posts with helpful features such as implementing image classification using MobileNetV2 for hashtag automation with 72% accuracy.

SOHA ABD EL-MOAMEN MOHAMED, MARGHANY HASSAN MOHAMED, and MOHAMMED F. FARGHALLY of Assiut University, Assiut, Egypt, authored a paper titled "Covid-19 Detection Based on Cascade-Correlation Growing Deep Learning Neural Network Algorithm." The goal of this paper was to detect the covid-19 using a new algorithm called "Cascade-Correlation

Growing Deep Learning Neural Network Algorithm (CCGDLNN)" from Computed tomography (CT) scan images of patients' chests. The authors applied the algorithm over 48,260 Computed tomography scan images. As the authors claimed, their model achieved 98.8% accuracy on more than 7996 test images.

IMAN AL-KINDI AND ZUHOOR AL-KHANJARI of Sultan Qaboos University, Muscat, Sultanate of Oman, wrote the paper titled "A comparative study of Classification Algorithms of Moodle Course Logfile using Weka Tool." The goal of this study was to conduct a thorough theoretical and experimental examination of classification data mining techniques, as well as a comparison study, to determine which methodology is the best for identifying student performance with the support of their engagement, behavior, and personality during different activities of the course. The author claimed that in their study of the comparison results, the classification algorithm with the best accuracy is the Random Forest, with 97.36 % correctly predicted instances.

We would like to express our sincere appreciation to all the authors and the reviewers for their contributions to this special issue. We hope you will enjoy the special issue and look forward to seeing you at future ISCA conferences. More information about ISCA society can be found at http://www.isca-hq.org.

Guest Editors:
Bidyut Gupta, University of Southern Illinois Carbondale, USA
Ajay Bandi, Northwest Missouri State University, USA
Mohammad Hossain, University of Minnesota at Crookston, USA

Improving Road Safety by Blockchain-based Monetization of Driver Behavior

Sruthi Rachamalla*
Southern Illinois University, Carbondale, Illinois 62901, USA.

Henry Hexmoor†
Southern Illinois University, Carbondale, Illinois 62901, USA.

Abstract

The transportation system places a top priority on driving safety. Most drivers on the road and their actions determine how safe it is to drive. Speed, hard braking, abrupt accelerations, and other aggressive driving behaviors are some of the main safety-compromising elements that could jeopardize human life in the event of a fatality. We presented a driver incentive model that ranks and rewards the driver's daily behavior in order to increase the safety of drivers and other road users. These rewards will come in the form of cryptocurrency tokens. We also examined the cooperative driving (or platooning) scenario. Road safety is improved by connecting two or more cars together by utilizing vehicular communication technologies. The leader is crucial as it manages the platoon, establishes communication between vehicles, and performs platoon maneuvers namely Join, Merge, Leave, and Split. As the leader of the platoon has multiple responsibilities than followers, our model rewards more incentives to the leader than followers. This digital monetization method is accomplished by secure transactions using blockchain.

Key Words: Cooperative driving, platooning, ranking, monetization, blockchain technology.

1 Introduction

Aggressive driving has become a global issue as per World Health Organization (WHO), Global status report on the road safety 2018 [3] and Centers for Disease Control and Prevention (CDC), Global Road Safety 2020 [18]. As per WHO [5] nearly 1.3 million people die each year on the world's roads. According to the United States, the National Highway Traffic Safety Administration (NHTSA) [1], aggressive driving is defined as "an individual commits a combination of moving traffic offenses so as to endanger other persons or property". Aggressive driving is a factor in 49% of all fatal motor vehicle crashes, according to the NHTSA, Traffic Safety Facts 2019 [49].

In addition, as stated by NHTSA [4] speeding is the leading aggressive driving behavior which accounts for 17.2% fatal crashes in 2019. It also defines that accidental difficulties to others on road are caused by aggressive driving [1]. The aggressive driving includes but not limited to speeding [6, 48], rapid acceleration or deceleration [15], sudden lane change [28] etc.

For some drivers, aggressive driving is a dysfunctional habit which can jeopardize other drivers on the road. This behavior can be fun on the spur of the moment for the driver but is uncomfortable for other road users which can cause fatal situations. To mitigate unsafe driving, various law enforcement strategies have been placed. When combined with public awareness outlets, it's been shown to be effective in reducing unsafe driving patterns.

Cooperative driving uses vehicle-to-vehicle and infrastructure-to-vehicle wireless communication system and [40] emphasizes the technology aids in the interchange of data gathered from other cars that is impossible to obtain via on-board sensors. The Advanced Transportation Technology (PATH) project in California [45] first proposed the idea of cars traveling together on the road in 1980. Cooperative driving can improve the driving experience on the road by relieving the driver from some of the driving obligations. Traditional sensor based Adaptive Cruise Control (ACC) isn't enough for cooperative platooning, instead Cooperative Adaptive Cruise Control (CACC) should be considered. CACC broadcasts information such as speed, acceleration, and distance through wireless communication. By allowing CACC, the distance between vehicles can be minimized by following closely, improving both safety and fuel efficiency . The focus on cooperative driving or platooning has increased globally in recent years because of the potential it holds in road transportation mainly focusing on automated and mixed traffic. Truck platooning [8, 29], and CACC [46] were prominent examples of cooperative driving, which focused on minimizing inter-vehicular distance by obeying the "Three Second Rule" safety rule [52].

Having a good leader for a platoon is really crucial in forming, maintaining, and improving safety. There are a lot of methods in electing a platoon leader. [47] proposed an incentive based strategy using blockchain to elect a leader who is the best for the safety of the platoon. The other way is through voting [10] to elect the platoon leader. Some other methods may be through scoring and ranking the drivers based on the everyday driving and the driver with the best rank can only initiate platoon formation. Our inspiration is drawn from the ranking method. We added an incentive or monetization

*Email: sruthirachamalla@siu.edu.
†Email: henry.hexmoor@cs.siu.edu.

factor for selecting a platoon leader based on rank. When a driver of a vehicle drives everyday there will be a rank assigned to the driver performance and the driver with the best rank can become a platoon leader and other drivers will be followers. To encourage more drivers to be platoon leaders, a monetization system is required that is fair for all members. This digital monetization is implemented by the usage of a smart contract in blockchain technology that holds users to a higher level of behavior, which is promising in this regard. This smart contract establishes what constitutes acceptable behavior and prevents users from breaking that standard.

Congested roadways are linked to longer commute, lower fuel economy, and a higher risk of motor vehicle collisions. We could save a lot of travel time if we could reduce commuting times by a fraction of a second. One solution to the traffic problem is cooperative driving, also known as platooning [14]. Vehicles in a platoon communicate using an ad-hoc network or other communication protocols. These communication channels allow platoons to drive close to each other while maintaining a safe distance. A platoon of vehicles will have a leader who will interact with the platoon followers while managing the platoon and overseeing maneuvers. The platoon leader is in charge of speed, lane changes, braking, and so on, while the follower vehicles are in charge of following the leader vehicle.

To help improve the driving behavior, Demerit points system [35] and Driver Feedback systems [34] were used extensively. In recent times, driver credit or scoring systems were also introduced to enhance the road behavior. To further enhance the behavior of the drivers, we put forward a ranking framework along with monetization using the blockchain. The inspiration for ranking is drawn from credit bureaus [37, 32]. These credit bureaus put together the credit reports and credit scores of potential individuals that can borrow from lenders or government. The credit bureaus decide the creditworthiness of an individual just from a score. In the proposed system, a simulated driving dataset is generated. Significant features are extracted from the dataset and driver behavior is analyzed for assigning a rank for the driver. Based on the driving rank, the driver is monetized with Driver Safety Reward (DSR) tokens and the transactions are stored on a blockchain network along with driver attributes.

The contributions of this paper are organized as follows: Literature Survey on Driver behavior, Ranking Systems, Cooperative Platooning and Blockchain are discussed in Section 2. Section 3 presents the Monetizing of Driving Behavior including Platooning. Implementation of the simple test network to simulate the storing in Blockchain is elaborated in Section 4. Finally, Section 5 talks about Conclusion and future directions.

2 Literature Review

An *Aggressive* driver is defined as an individual who commits road traffic offenses and put others at risk. The attributes that contribute to driving aggressiveness are speeding, acceleration, braking etc. The authors in [25] proposed a method to detect driver aggressiveness on a vehicle based on visual and sensor features. A Support Vector Machine (SVM) classifier is used to classify those feature vectors in order to detect aggressiveness. This paper [21] states that hurriedness is the primary cause for speeding. They conducted a driving simulated study recruiting thirty-six drivers. The drivers in a hurry drove with higher speeds, accelerated faster, decelerated faster, made tight turns, accelerated faster after red lights, left smaller gaps between vehicles, were more likely to pass a slow vehicle.

Driving behavior plays a major role in improving the road safety. Lately, with the advancements in smart devices and Internet of Things (IoT) [11, 12], the sensors generate huge amounts of data. The data that can be extracted but not limited to speed, braking, accelerations, trip distance, accelerometer, magnetometer, gyroscope information etc. This rich data can be used to analyze and classify driving behavior patterns. In [19] they monitored the driving behavior from the collected set of experimental data to detect the accelerations, brakings and lane-changing behavior while providing constructive audio feedback to the driver. In paper [7] they exploited the demerit point system in Denmark. They introduced a point-recording scheme to record the drivers behavior to a non-monetary penalty method. Based on the driving behavior the responses are stored and demerit points are assigned to their driving licenses. Depending on the number of demerit points piled, drivers with more demerit points reduced the frequency of committing traffic offenses by 9–34%. The authors in [17] proposed a driving profile platform called SenseFleet, to detect risky driving events using smartphone sensors to identify driving maneuvers. A representative score for a driver was accurately detected using real-time information by applying the Fuzzy Logic Systems.

Ranking is a way to give credit for a safe driving behavior. The authors in [51] proposed a Driving Safety Credit system inspired by credit scoring in financial security field, and designed a scoring method using driver's trajectory data and violation records. Initially they extracted driving habits, aggressive driving behaviors and traffic violation behaviors from driver's trajectories and traffic violation records. Later, they trained a classification model to filter out irrelevant features and scored each driver with selected features. In order to accurately identify abnormal driving behavior, the authors in the paper [30] proposed different abnormal driving behavior recognition algorithms. They obtained the data from OBD terminal that combines acceleration changes and behavior. The model combines the driving data of the driver, takes the proportion of abnormal driving behavior as the evaluation index, and uses the entropy weight method and the analytic hierarchy process to obtain the index weight. The model can analyze and evaluate the driving behavior of the driver and give a score for driver's behavior.

This model can also be extended for cooperative platooning. The first platooning simulator that was developed was Hestia [22]. This simulator is used for simulating various scenarios using sensors but its drawback is that it doesn't execute the platooning maneuvers and does not simulate traffic scenarios. In paper [20], the researchers tried to simulate the mixed traffic scenarios using SUMO [31] and were unsuccessful. They implemented a car following model based on CACC to simulate inter-vehicle communication.

The researchers in [27] mainly focused on analyzing the communication effects using the platooning simulator. In [50] paper, they manually simulated the mixed traffic scenarios and tested the simulator to study the consequences of CACC on traffic.

There are different simulation tools that are available in platooning discipline. The PLEXE [44] simulation tool is a platooning extension for SUMO [31] which is open-source and is available to the community. This simulation tool has different CACC car-following models to experiment. The wireless communication protocols are available for simulating the formation and platoon management. Mixed traffic scenarios are available to use and implement platooning maneuvers. The authors in paper [24], developed a simulator to implement platooning maneuvers such as to join an existing platoon and merge two platoons. In paper [10], the authors developed a state-of-the-art simulator based on VENTOS [9] which uses SUMO. PERMIT [36] is a tool which simulates platooning maneuvers like join, merge, leave, and split which is built on Plexe [44].

One way to store the details of driver behavior digitally and securely is through *Blockchain Technology* [39]. It can be defined as decentralized, distributed, encrypted, immutable, trust-free, digital ledger system. Bitcoin [38] and Ethereum [2] are the decentralized peer-to-peer digital currencies that are the most popular examples that rely on blockchain technology. In the paper [41], a new blockchain model was implemented to regulate the traffic offense using demerit points. Smart contracts were used as a conditional filter. These smart contracts store driver offense's demerit points,fines collected, and penalty information including revocation of driver license. A user interface was provided for a traffic officer to input the driver's offense and drivers can check the offense. The evaluation shows that the smart contracts are executed properly as compared to real regulations.

Currently, to our knowledge only few researchers are working on effectively combining the benefits of blockchain technology with the platooning technology to better understand the usefulness. The authors in paper [43] used blockchain as a medium in transportation. They achieved the communication between vehicles in platoon and blockchain public key infrastructure by securely using hardware-based side channels. In [16], they decreased the blockchain transaction validation time, and verified the vehicle identity. The authors in [23], emphasized on using blockchain with platooning to share information securely and rapidly. In [33] paper, the authors build a reputation system based on blockchain for anonymous vehicles. The authors in [26] implemented a leadership incentives mechanism based on blockchain technology for heterogeneous and dynamic platoons.

In this study, existing research such as demerit point system, feedback models, and scoring methods for driving aggressiveness can be further extended by introducing monetization. The driver is offered test crypto tokens and these transaction details along with driver attributes are stored on a blockchain network.

3 Our Approach

This methodology consists of two variations for monetizing the driving behavior one without cooperative platooning and other with it. The main difference between these two forms are what simulation tool is used for fetching the required data and earnings.

3.1 Simulation of Urban Mobility (SUMO)

Using ***Simulation of Urban Mobility (SUMO)*** [13], a road network is built and random traffic is generated. Many characteristics need to be taken into consideration to represent the road network environment. However, we mainly focused on the few performance factors such as Speed, Acceleration, Braking, and Over Speed Limit (OSL). An XML formatted dataset is extracted by running the simulator. From this raw XML dataset, necessary features like Driver ID, and Speed are extracted. Different driver behavioral attributes like the number of sharp decelerations are calculated based on speed. A driving regulatory rule stating that reducing speed of 6mph ($2.5m/s$) in one second is considered as a sharp brake is used as a baseline for calculating the count of the sharp braking. Similarly, the number of sharp accelerations is also determined. Another law states that representing the posted speed limit (max speed limit + 7mph) is being used to compute the over speed feature. By repeating the above process, random driver behaviors are collected each day. Once the features are being prepared in the features extraction phase, a rank is assigned to the driver considering aforementioned features. The driver behavior is captured for 17 drivers over 4 days. The generated data is used for analysis of driver behavior detailed more in [42]. The entire architecture of the above process is detailed in Figure 1.

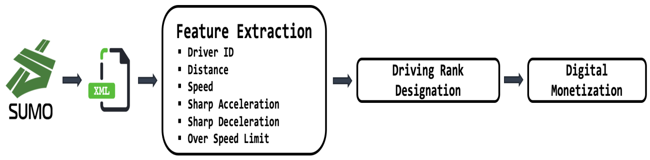


Figure 1: SUMO methodology

3.2 Driver Scoring Model

The driver scoring model takes the parameters from section 3 to rank the driver. The following are the steps that need to be followed to score the driver.

- Calculate the total trip distance and scale using min-max normalization.
- Over speed limit (OSL) percentage is computed using total trip distance and Over Speed Limit (OSL) count.
- The Acceleration Percentage (AP) and Deceleration Percentage (DP) are computed.
- Define the weights for above three parameters as 60% for Deceleration, 30% for Acceleration and 10% for OSL.

- Compute the score using weighted average.

$$Score = \frac{0.3*AP+0.6*DP+0.1*OSL}{100}$$

- The weighted average is scaled between 1-5 as referred in Table 1.

Table 1: Score rating

Rank	Rating
5	Excellent
4	Very Good
3	Good
2	Bad
1	Very Bad

3.3 Monetization Heuristic

An earning rate scale is defined depending on the rank calculated from subsection 3.2. The Table 2 shows the earning rates.

Table 2: Monetization

Score	Earning Rate
5	0.15
4	0.12
3	0.09
2	0.03
1	0.01

Using scaled total distance (S_{dis}) and earning rate (ER) from above, total test tokens to be credited in the Rinkeby driver wallet is determined by the following formulation.

$$E_{tokens} = S_{dis} * ER$$

Following subsections explain the above methodology including the cooperative platooning.

3.4 Simulation with PERMIT and Feature Extraction

PERMIT [36], an open source platooning simulator based on SUMO and its platooning extension PLEXE. With PERMIT, Merge, Join, Leave and Split maneuvers can be performed. *Merge* is a maneuver in which two platoons join to form one platoon. *Join* refers to joining one vehicle into an existing platoon. *Leave* maneuver is when a car exits the current platoon. *Split* refers to dissolving one single platoon into two platoons. By using the PERMIT, we simulated all these maneuvers. This provides the data required which are (1) number of cars, and (2) distance traveled for evaluating the earnings with platoon for a driver. The model for the Simulation with Platooning is shown in the Figure 2.

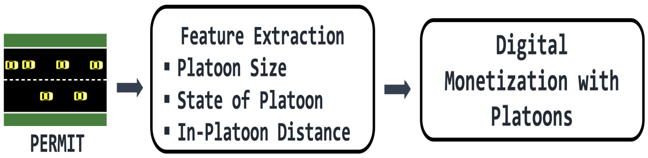


Figure 2: Permit system methodology

3.5 Digital Monetization with Platoons

With the features extracted from the previous step, we formalized the earnings (er_d) for the driver as summation of the earnings achieved while he drove in platoons (er^{in}) and earnings achieved while driving outside the platoon (er^{out})

$$er_d = er^{in} + er^{out} \tag{1}$$

Because different maneuvers can exist inside the platoon, we decomposed the earning offered inside the platoon into addition of earnings during join(P^{er}_{join}) and leave(P^{er}_{leave}) maneuver.

$$er^{in} = P^{er}_{join} + P^{er}_{leave} \tag{2}$$

However, the er^{out} is calculated as product of previous day earnings rate (ER_{d-1}) which determined by the Driving Rank Designation Model and distance travelled by the driver outside the platoon(d^{out}).

$$er^{out} = ER_{d-1} * d^{out} \tag{3}$$

As mentioned earlier that a platoon leader will have a little favor by this model due to his responsibilities, join and leave maneuvers are calculated using two different equation one representing Leader and other Follower in the platoon.

Join Maneuver For every platoon, driver joined on a particular day, and for all the states in each platoon, calculate the product of the average of the States of the platoon(S_i) and the sum of the previous earning rate of the driver(ER_{d-1}) and $n\delta$. The State(S_i) is defined as the product of the Length of Platoon(L_i) at state i and distance travelled inside the platoon(d_i^{in}). Here the term $j\delta$ is the additional incentive for the leader of the platoon. It represents the summation of the balancing factor over the number of cars joined in the platoon. The balancing factor δ is used to control the amount of incentive the driver will be provided during the platoon. We assigned it as 0.01.

$$P^{er}_{join}(L) = \sum_{p=1}^{w}\sum_{i=1}^{n} S_i * (ER_{d-1} + j\delta) \tag{4}$$

The difference between leader and follower is there will be no additional incentives for follower. Instead, only the balancing factor is added to the previous day earning rate(ER_{d-1}). Platoon follower doesn't require length of the platoon. So, it just uses the distance travelled in each state.

$$P^{er}_{join}(F) = \sum_{p=1}^{w}\sum_{i=1}^{n} d_i * (ER_{d-1} + \delta) \tag{5}$$

Leave Maneuver During the leave maneuver, for the leader, instead of the number of cars joined(j), number of cars left(l)

is considered. Additionally, there will be a penalty if a car leaves the platoon before travelling η miles. In other words, earnings for the followers will start only after travelling η miles. The overhead incurred by changing the structure of the platoon while on the move is the main reason for penalty. Here we considered $\eta = 10$.

$$P^{er}_{leave}(L) = \sum_{p=1}^{w}[\sum_{i=1}^{n} S_i * (ER_{d-1} - l\delta) + penalty_w] \quad (6)$$

where,

$$S_i = L_i * d_i^{in} \quad (7)$$

$$penalty = \begin{cases} (d^{in} - \eta) * \delta, & \text{if } d^{in} < \eta \\ 0, & \text{otherwise} \end{cases}$$

All the notations used in the model are summarized in the Table 3. To ensure the safety on the road, drivers with rank less than four cannot act as platoon leaders.

Table 3: List of Notations

Symbol	Definition
er_d	Earnings of particular day
er^{out}	Earnings outside of the platoon
er^{in}	Earnings inside of the platoon
ER_{d-1}	Earning rate of previous day
d^{out}	Out-platoon distance
n	Number of cars in a platoon
j	Number of followers in the platoon in join maneuver
l	Number of cars left the platoon in leave maneuver
δ	Balancing factor
η	Penalty factor
S	State-of-platoon
L_i	Length of the platoon at state i
d^{in}	In-platoon distance at state S
w	Number of platoons a driver travelled
$P^{er}_{join}(L)$	In-platoon earnings with join maneuver of leader
$P^{er}_{join}(F)$	In-platoon earnings with join maneuver of follower
$P^{er}_{leave}(L)$	In-platoon earnings with leave maneuver of leader

3.6 Storing in Blockchain

For secure transaction of the crypto tokens, the extracted data from the feature extraction step is stored in a blockchain technology. Features stored are driver ID, current earnings, rank designated for the driver, over speed limit count, distance traveled, number of sharp accelerations, number of sharp decelerations, number of platoons he joined, platoon leader activity and earning date. The above architectures can be simplified into one single architecture as shown in Figure 3.

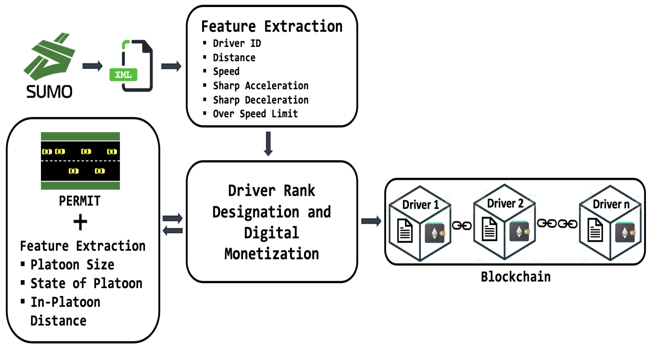


Figure 3: Methodology

4 Driver Safety Reward (DSR) Implementation

Upon the extraction of the features from the SUMO simulator and assigning a rank to the driver based on the driving, driver behavioral characteristics are stored in a blockchain [39]. Each block in the chain consists of the attributes such as driver ID, distance traveled, number of sharp breaking, sharp accelerations, exceeded speed limit count, rank, earnings, and earning date. While inserting a block into the blockchain, the driver's DSR wallet will be credited with the certain number of DSR test tokens which can only be used for vehicular purposes.

In a case considering a platoon (platoon size of 6), PERMIT is used for implementing join and leave maneuvers. With the data from the PERMIT and the above formulation, we calculated the earnings for a leader and a follower in both scenarios.

For implementation, a Rinkeby test network is used as our ethereum network. Two smart contracts one for tokenization, the other for storing driver record were deployed on Rinkeby Etherscan network. Tokenization contract will initially approve the driver record contract with certain limit of driver DSR test tokens and transfer few DSR test tokens to the driver record contract. Now, the driver record will be able to credit the DSR test tokens to the assigned drivers based on the ranking to their wallet.

For illustration, the tokenization contract generated the 10000000 (10^7) DSR test tokens and approved the data record to spend those. The contract transfers 10000 DSR test tokens to data record for further assigning them to the drivers. As shown in Figure 4, the MetaMask represents 9990000 ($0.999 * 10^7$) DSR test tokens in the admin wallet. After inserting the data record into the Rinkeby test network, 2 DSR test tokens and 26 DSR test tokens are credited into "Driver_1" account as shown in the Figure 5. Moreover, driver data can be retrieved from the network using defined method. Both transaction logs can be seen in Tables 4 and 5.

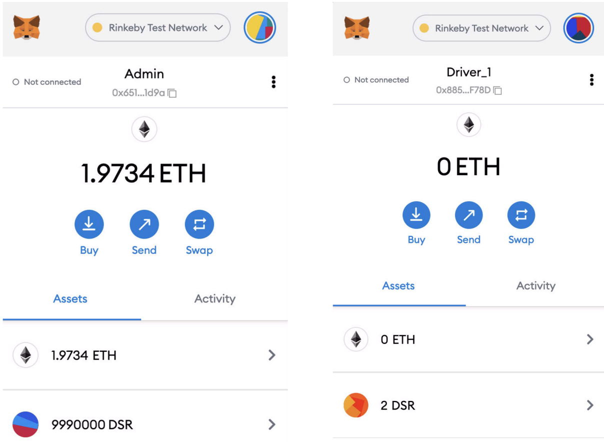


Figure 4: Admin and Driver Wallet without platooning

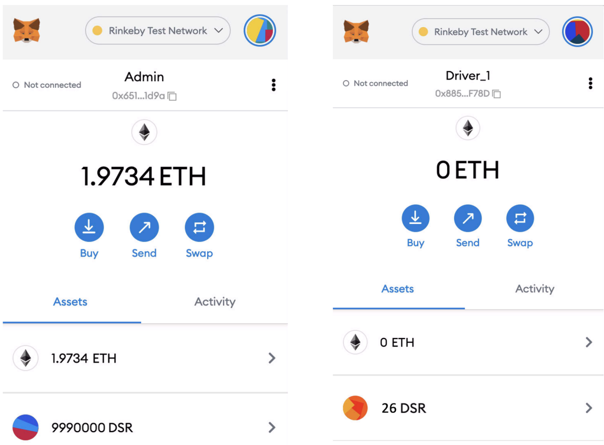


Figure 5: Admin and Driver Wallet with platooning

5 Conclusion and Future Work

To encourage safe driving practices, we proposed a methodology to quantify and monetize the driver's behavior along with platooning. This system considers the different maneuvers in platoon namely Join, Leave, Merge, and Split. Simulation tools SUMO and PERMIT were used. Based on the observed driving behavior from the aforementioned tools, the aggressive driving patterns and driver scoring model were presented. The score is measured on a scale from 1-5 (1 being Very Bad - 5 being Excellent). In addition to that a reward based system was proposed where crypto tokens were awarded provided the rank (these earned DSR test tokens are not to be exchanged for currency). This transaction data along with the driver properties are entered in a decentralized Rinkeby Test network for ease of access by the different end points. We would also like to extend this framework considering the other parameters like cornering, weather, time of the day, age, gender while using real time dataset.

References

[1] Aggressive driving chapter 1: Introduction. *NHTSA*, Jan 2006.

[2] Ethereum whitepaper, 2014.

[3] Global status report on road safety 2018. *World Health Organization*, June 2018.

[4] Facts and statistics on aggressive driving. *III*, Jun 2021.

[5] Road traffic injuries. *World Health Organization*, June 2022.

[6] Speeding. *NHTSA*, Jan 2022.

[7] Kibrom A Abay. How Effective Are Non-Monetary Instruments for Safe Driving? Panel Data Evidence on the Effect of the Demerit Point System in Denmark. *The Scandinavian Journal of Economics*, 120(3):894–924, 2018.

[8] Assad Alam, Bart Besselink, Valerio Turri, Jonas Mårtensson, and Karl H Johansson. Heavy-Duty Vehicle Platooning for Sustainable Freight Transportation: A Cooperative Method to Enhance Safety and Efficiency. *IEEE Control Systems Magazine*, 35(6):34–56, 2015.

[9] Mani Amoozadeh, Bryan Ching, Chen-Nee Chuah, Dipak Ghosal, and H Michael Zhang. Ventos: Vehicular network open simulator with hardware-in-the-loop support. *Procedia Computer Science*, 151:61–68, 2019.

[10] Mani Amoozadeh, Hui Deng, Chen-Nee Chuah, H Michael Zhang, and Dipak Ghosal. Platoon Management with Cooperative Adaptive Cruise Control Enabled by VANET. *Vehicular communications*, 2(2):110–123, 2015.

[11] Kevin Ashton. That 'Internet of Things' Thing. *RFID journal*, 22(7):97–114, 2009.

[12] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The Internet of Things: A Survey. *Computer networks*, 54(15):2787–2805, 2010.

[13] Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. Sumo–simulation of urban mobility: an overview. In *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind, 2011.

[14] Carl Bergenhem, Steven Shladover, Erik Coelingh, Christoffer Englund, and Sadayuki Tsugawa. Overview of platooning systems. In *Proceedings of the 19th ITS World Congress, Vienna, Austria*, pages 1–7, Oct 2012.

[15] Prashant Shridhar Bokare and Akhilesh Kumar Maurya. Acceleration-Deceleration Behaviour of Various Vehicle Types. *Transportation research procedia*, 25:4733–4749, 2017.

Table 4: insertRecord Log

insertRecord Log	
hash	0xe74************5c835a
from	0x651************9d4b7d
to	driverRecord.insertRecord(address,string,uint256,uint256, uint256, uint256,uint256,uint256, uint256,uint256,string) 0x4ABc************49554b
gas	278003 gas
transaction cost	278003 gas
hash	0xe74************5c835a
input	0xb12************00010
decoded input	"address driverAddress": "0x44************6A147", "string driverId": "Driver_1", "uint256 distance": "200", "uint256 sharpBreaking": "1", "uint256 sharpAcc": "0", "uint256 overSpeedLimit": "0", "uint256 totalPlatoonsJoined": "2", "uint256 platoonLeader": "0", "uint256 rank": "4", "uint256 earnings": "2000000", "string earningDate": "06-24-2022"

Table 5: getDriverInfo Log

getDriverInfo Log	
decoded input	"string driverId": "Driver_1"
decoded output	"0": "tuple(address,string,uint256, uint256,uint256,uint256, uint256, uint256,uint256,uint256,string): 0x44659c35594C9149C872F5813526fde5A8b6A147, driver1,200,1,0,0,2,0,4,2000000,06-24-2022"

[16] Jose Angel Leon Calvo and Rudolf Mathar. Secure blockchain-based communication scheme for connected vehicles. In *2018 European Conference on Networks and Communications (EuCNC)*, pages 347–351. IEEE, 2018.

[17] German Castignani, Thierry Derrmann, Raphaël Frank, and Thomas Engel. Driver Behavior Profiling Using Smartphones: A Low-Cost Platform for Driver Monitoring. *IEEE Intelligent transportation systems magazine*, 7(1):91–102, 2015.

[18] CDC. Globalroadsafety. *Centers for Disease Control and Prevention*, Dec 2020.

[19] Mohamed Fazeen, Brandon Gozick, Ram Dantu, Moiz Bhukhiya, and Marta C González. Safe driving using mobile phones. *IEEE Transactions on Intelligent Transportation Systems*, 13(3):1462–1468, 2012.

[20] Pedro Fernandes and Urbano Nunes. Platooning of autonomous vehicles with intervehicle communications in sumo traffic simulator. In *13th International IEEE Conference on Intelligent Transportation Systems*, pages 1313–1318. IEEE, 2010.

[21] Cole D Fitzpatrick, Siby Samuel, and Michael A Knodler Jr. The use of a driving simulator to determine how time pressures impact driver aggressiveness. volume 108, pages 131–138. Elsevier, 2017.

[22] Simon Hallé and Brahim Chaib-draa. A Collaborative Driving System Based on Multiagent Modelling and Simulations. *Transportation Research Part C: Emerging Technologies*, 13(4):320–345, 2005.

[23] Henry Hexmoor, Suray Alsamaraee, and Maha Almaghshi. Blockchain for improved platoon security. *International Journal of Information*, 7(2):1–6, 2018.

[24] Elham Semsar Kazerooni and Jeroen Ploeg. Interaction protocols for cooperative merging and lane reduction scenarios. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 1964–1970. IEEE, 2015.

[25] Omurcan Kumtepe, Gozde Bozdagi Akar, and Enes Yuncu. Driver aggressiveness detection via multisensory data fusion. *EURASIP Journal on Image and Video Processing*, 2016(1):1–16, 2016.

[26] Brian Ledbetter, Samuel Wehunt, Mohammad Ashiqur Rahman, and Mohammad Hossein Manshaei. Lips: A protocol for leadership incentives for heterogeneous and

dynamic platoons. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, volume 1, pages 535–544. IEEE, 2019.

[27] C Lei, EM Van Eenennaam, W Klein Wolterink, G Karagiannis, Geert Heijenk, and J Ploeg. Impact of packet loss on cacc string stability performance. In *2011 11th International Conference on ITS Telecommunications*, pages 381–386. IEEE, 2011.

[28] Xin-Gang Li, Bin Jia, Zi-You Gao, and Rui Jiang. A realistic two-lane cellular automata traffic model considering aggressive lane-changing behavior of fast vehicle. *Physica A: Statistical Mechanics and its Applications*, 367:479–486, 2006.

[29] Kuo-Yun Liang. *Fuel-efficient heavy-duty vehicle platoon formation*. PhD thesis, KTH Royal Institute of Technology, 2016.

[30] Tianshi Liu, Guang Yang, and Dong Shi. Construction of driving behavior scoring model based on obd terminal data analysis. In *2020 5th International Conference on Information Science, Computer Technology and Transportation (ISCTT)*, pages 24–27. IEEE, 2020.

[31] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. *IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 2575–2582, 2018.

[32] Inessa Love and Nataliya Mylenko. *Credit reporting and financing constraints*, volume 3142. 2003.

[33] Zhaojun Lu, Qian Wang, Gang Qu, and Zhenglin Liu. Bars: A blockchain-based anonymous reputation system for trust management in vanets. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 98–103. IEEE, 2018.

[34] Andreas A Malikopoulos and Juan P Aguilar. An Optimization Framework for Driver Feedback Systems. *IEEE Transactions on Intelligent Transportation Systems*, 14(2):955–964, 2013.

[35] Arif Mehmood. Evaluating impact of demerit points system on speeding behavior of drivers. *European Transport Research Review*, 2(1):25–30, 2010.

[36] Jesús Mena-Oreja and Javier Gozalvez. Permit-a sumo simulator for platooning maneuvers in mixed traffic scenarios. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3445–3450. IEEE, 2018.

[37] Margaret J Miller. *Credit reporting systems and the international economy*. Mit Press, 2003.

[38] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, page 21260, 2008.

[39] Michael Nofer, Peter Gomber, Oliver Hinz, and Dirk Schiereck. Blockchain. *Business & Information Systems Engineering*, 59(3):183–187, 2017.

[40] J Piao and Mike McDonald. Advanced Driver Assistance Systems from Autonomous to Cooperative Approach. *Transport reviews*, 28(5):659–684, 2008.

[41] Aditya Pradana, G Sing, Y Kumar, and A Mohammed. Blockchain traffic offence demerit points smart contracts: Proof of work. *International Journal of Advanced Computer Science and Applications*, 9(11):375–382, 2018.

[42] Sruthi Rachamalla and Henry Hexmoor. Dsrbt-driving safety reward based on blockchain technology. *Proceedings of 37th International Conference on Computers and Their Applications*, 82:72–81, 2022.

[43] Sean Rowan, Michael Clear, Mario Gerla, Meriel Huggard, and Ciarán Mc Goldrick. Securing vehicle to vehicle communications using blockchain through visible light and acoustic side-channels. *arXiv preprint arXiv:1704.02553*, 2017.

[44] Michele Segata, Stefan Joerer, Bastian Bloessl, Christoph Sommer, Falko Dressler, and Renate Lo Cigno. Plexe: A platooning extension for veins. In *2014 IEEE Vehicular Networking Conference (VNC)*, pages 53–60. IEEE, 2014.

[45] Steven E Shladover, Charles A Desoer, J Karl Hedrick, Masayoshi Tomizuka, Jean Walrand, W-B Zhang, Donn H McMahon, Huei Peng, Shahab Sheikholeslam, and Nick McKeown. Automated Vehicle Control Developments in the PATH Program. *IEEE Transactions on vehicular technology*, 40(1):114–130, 1991.

[46] Steven E Shladover, Christopher Nowakowski, Xiao-Yun Lu, and Robert Ferlis. Cooperative adaptive cruise control: Definitions and operating concepts. *Transportation Research Record*, 2489(1):145–152, 2015.

[47] Pranav Kumar Singh, Sahil Sharma, Sunit Kumar Nandi, Roshan Singh, and Sukumar Nandi. Leader election in cooperative adaptive cruise control based platooning. In *Proceedings of the 1st International Workshop on Communication and Computing in Connected Vehicles and Platooning*, pages 8–14, 2018.

[48] Stephen Stradling, Michelle Meadows, and Susan Beatty. Characteristics of speeding, violating and thrill-seeking drivers. In *International Conference on Traffic and Transport Psychology, Bern*, 2000.

[49] NHTSA U.S. Traffic safety facts 2019. *NHTSA CrashStats*, Aug 2021.

[50] Bart Van Arem, Cornelie JG Van Driel, and Ruben Visser. The impact of cooperative adaptive cruise control on traffic-flow characteristics. *IEEE Transactions on intelligent transportation systems*, 7(4):429–436, 2006.

[51] Wenfu Wang, Yao Yang, An Chen, and Zhijie Pan. A scoring method for driving safety credit using trajectory data. In *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pages 558–565. IEEE, 2019.

[52] Michael T Wolf and Joel W Burdick. Artificial potential functions for highway driving with collision avoidance. In *2008 IEEE International Conference on Robotics and Automation*, pages 3731–3736. IEEE, 2008.

Sruthi Rachamalla received her M.S. degree from Southern Illinois University, Carbondale. Currently she is a Doctoral candidate in the School of Computing at Southern Illinois University, Carbondale. Her current interests include cyber-physical systems, wireless networks, multiagent systems, cooperative driving and blockchain.

Henry Hexmoor received the M.S. degree from Georgia Tech, Atlanta, and the Ph.D. degree in computer science from the State University of New York, Buffalo, in 1996. He taught at the University of North Dakota before a stint at the University of Arkansas. Currently, he is a Professor with the School of Computing, Southern Illinois University, Carbondale, IL. He has published widely in artificial intelligence and multiagent systems. His long term and sponsored research include federal and state agencies and prime contractors. He is an IEEE senior member. His current interests include cyber-physical systems, wireless networks, complex networks, multiagent Systems, cognitive science, robotics, and digital forensics, and blockchain.

Video Surveillance Architecture from the Cloud to the Edge

Jeff McCann*
Dell Technologies and University of Limerick, Limerick, IRELAND

Liam Quinn†
Dell Technologies, Austin, Texas, USA

Sean McGrath‡ and Colin Flanagan‡
University of Limerick, IRELAND

Abstract

This paper examines the use of digital video in public safety and surveillance systems. Traditionally video recordings are used by law enforcement to review events retrospectively and for evidential purposes in the pursuance of criminal prosecution. We also examine how, due to the proliferation of cameras around cities, human operators are challenged to monitor these data feeds in real-time and how the emergence of AI and computer vision solutions can process this data. Computer vision can enable the move from a purely reactive to a predictive, real-time analysis platform. As camera numbers and the resolution and framerate of cameras grow, existing network infrastructure frequently causes challenges provisioning low latency, high bandwidth networking to private or public cloud infrastructure for evidential storage. These technical challenges can provide issues for law enforcement providing a data chain of custody to ensure its admissibility during court proceedings. Emerging technologies offer solutions to overcome these challenges: the use of emerging edge compute capabilities, including the use of on-camera and mobile edge compute nodes providing compute capabilities closer to the data source and new software paradigms, including CI/CD methodologies, and the use of micro-services and containerization to manage and deliver applications across the portfolio of devices, at the edge of the network. Using Amazon Web Services as an example, we review how cloud providers are now overcoming challenges in delivering real time video analytics solutions in the classical cloud model, and how they are enabling services and platforms closer to the edge, while delivering the cloud computing experience of scalability and manageability across different edges of the network.

Key Words: Surveillance, edge, CCTV, video analysis, cloud management, SaaS, PaaS, Amazon web services, AWS.

* Director, Customer Solution Centers, Email: jeff_mccann@dell.com, Jeffrey.Mccann@ul.ie.
† Senior Fellow, Dell Technologies, Email: liam_quinn@dell.com.
‡ Senior Lecturers, Department of Electronic and Computer Engineering, Email: sean.mcgrath@ul.ie, colin.flanagan@ul.ie.

1 Introduction

The use of closed-circuit television systems (CCTV) has its roots in the 1940s, with the first documented use of CCTV systems in Durham, UK, in 1956 [95]. This system enabled a police officer to monitor and operate traffic lights. The use of cameras in law enforcement has been, to date, mainly for evidential purposes, with data stored and then manually reviewed post-incident by CCTV operators. The migration from magnetic tape recordings to digital media stored on centralized computer systems has enabled the deployment of surveillance cameras at a much higher density than would have been possible previously.

The challenges of transporting data to the cloud for processing have long been acknowledged as problematic, especially for large datasets such as streaming video. On cloud platforms, Network latency is the primary challenge to processing streaming data in real-time [2]. Numerous methods of moving compute closer to the data source have been proposed to alleviate this latency, including Fog[9], Cloudlets [76] and Edge computing. Lin, et al. [50] discuss the difference between edge and fog computing: *"edge computing builds the architecture of computing at the edge, while fog computing uses edge computing and further defines the network connection over edge devices, edge servers, and the cloud."* These edge devices can provide traditional CPU and accelerator compute capabilities to enable computer vision code to run on resource-constrained edge devices. On-camera compute already provides significant bandwidth reductions in several use cases including motion detection and automatic number plate recognition[57]. The camera then only returns metadata, along with an evidential photograph of a speeding car, rather than a full video stream from the cameras to be processed in the cloud. By processing on the camera, both network traffic and the amount of storage required in the system[32] are reduced compared to traditional evidential recording platforms. To deliver timely, predictive and proactive computer vision analytics platforms, the design of conventional evidential recording systems needs to be reviewed to move the compute capabilities closer to the source of the data.

2 Evidential Recording Platforms

To provide evidence for law enforcement agencies after an incident and provide a chain of custody of video footage to be used in prosecutions, Digital (DVR) or Network (NVR) video recording systems provide a platform to deploy and manage the cameras and storage of the data they create. The systems also include management structures for the stored data to ensure storing, access and deletion according to legal data governance requirements. Centrally managed digital surveillance systems have several key platform components, described in Figure 1.

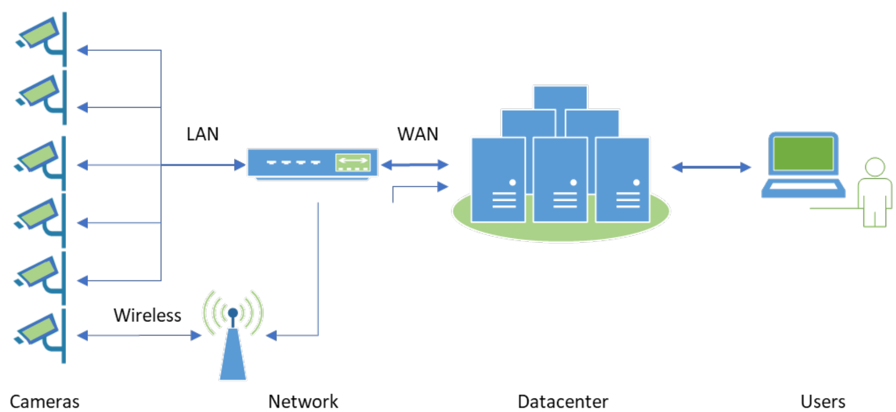


Figure 1: Evidential recording infrastructure

2.1 Cameras

The first digital IP cameras became commercially available in 1996, with the release of Axis Communications Neteye 200 camera [25], which supported a resolution of 352x288 pixels at a frame rate of 1 frame per second (FPS) in JPEG format [10]. Currently, the most popular resolution for digital surveillance systems is full HD (1920x1080), producing uncompressed data streams of up to 1.5Gbit/S [28]. Using H.264/AVC compression reduces this data stream by up to 70%. As H.264/AVC is an asymmetric process, with more compute required at the encoder than at the decoder [40], onboard microprocessors in cameras have evolved in parallel with the image sensor capabilities, with CPU, GPU & FGPA capabilities or by a specialist Digital Signal Processor (DSPs) [62]. Alongside compression, the compute capabilities also provide remote management capabilities, essential for large suites of cameras. The ONVIF [63] specification for camera management is included in published standards, such as IEC 62676, for Video Surveillance Systems.

2.2 Network

Digital surveillance cameras have standardized TCP/IP over ethernet and generally use IEEE standard 802.3u (Fast Ethernet/ 100Base-T). As TCP/IP is bi-directional, it also enables the management and manipulation of Point, Tilt & Zoom (PTZ) cameras movement controls [14] without the need for secondary cabling. Cameras connect to local area network (LAN) switches, which can also act as power sourcing equipment to provide power and data to cameras via one cable, using IEEE 802.3a(x) standards [60]. The use of Wi-Fi for surveillance systems to connect static cameras, using IEEE 802.11 is used in limited circumstances but provides range, reliability, and security challenges for critical systems. [23], but Wi-Fi and 4G cellular connectivity are widely utilized for body-worn and mobile/vehicle cameras[54]; however, these devices frequently have localized storage to overcome connectivity issues and limited recording periods due to battery charge longevity [33]. Backhaul to the datacenter is dependent upon each installation, with fully private fiber networks utilized in very high-security environments or built upon virtual private networks (VPN) provided by 3rd party telecom providers, with networking and security capabilities such as NAT, Firewalls, and VPN Tunnels used to protect the transmission of the data. [18] The ESTI standard for TErrestrial Trunked RAdio (TETRA) provides a data carrier protocol, with up to 600mbps throughput, to provide fully encrypted, secure communications but requires a separate infrastructure for broadcasting capabilities, aside from regular telco operated environments. One of the significant examples of the use of TETRA in surveillance was its use at the Athens Olympics, where feeds from live CCTV cameras were broadcast via Tetra to security/law enforcement officers handsets on the ground [75].

2.3 Datacenter

The core of all video surveillance systems is the network video recording (NVR) system, with leading providers including Milestone Systems, Avigilon, Bosch, Huawei and Genetec [26]. The NVR provides a range of features, including management of the cameras, storage management, including writing data to storage, and managing data, ensuring timely deletion, the chain of custody reporting, and access for users to review the recorded footage.

Servers are predominantly Intel x86 platforms, with many NVR providers using Microsoft Windows© Server or Linux operating systems. Depending on the scale of requirements, storage may be anything from a single hard disk to a complete server & storage area network configuration, as shown in Figure 2 below. Storage Area Networks (SAN) provide network-attached storage, using iSCSI, or Fiber-channel over IP connectivity with Redundant Array of Inexpensive Disks (RAID) offering fault-tolerant, highly scalable storage platforms for storage and data throughput capabilities [90]. Disk configuration is dependent on several factors from the dataflow: the number of cameras, frame resolution & speed; motion detection; compression algorithms; the number of days storage, expected activity levels in the cameras [20, 48] and the hardware in the SAN, including the number of disks, IOPs for each disk, RAID or other redundancy/data protection systems and SAN processor speed.

2.4 Users

Users require a method of accessing the stored data, either from individual cameras or in Command & Control walls using multiple screens, with thumbnail streaming video images of multiple cameras displaying concurrently. This gives the user the ability to observe many screens simultaneously, and to click into one of the thumbnails to maximize screens of interest. The NVR software also provides the user with methods to view historical material and protect the material of interest against overwriting by the NVR storage management schedule. User feeds are delivered to a proprietary application running on a personal computer or via HTTP/s web browser. NVR Manufacturers recommend that the user workstation provide substantial processing power, both from the CPU and Graphics Processing Unit (GPU), with 8GB RAM and a 64-bit Windows operating system, to deliver a satisfactory user experience significant numbers of camera feeds on screen [6, 59].

3 Video Analysis Platforms

The UK has led the global growth of surveillance [80], with over 500,000 cameras in London and 15,000 on the Underground alone. Research shows that video surveillance was useful to investigators in some 29% of crimes committed on the British transport systems [5]. With the proliferation of cameras for surveillance purposes, it is impossible to monitor video feeds in real-time. Alongside the growth of surveillance systems, the rise of computer vision technologies built upon research in artificial intelligence (AI) has provided the building blocks for video analysis. Using large previously labelled sets of data to train the convolutional neural networks (CNN) [79, 92] built upon deep neural networks (DNN) [72] before deployment to analyze real-time video feeds. AI-enabled video analysis provides evidential data and provides opportunities for law enforcement to offer proactive capabilities using motion detection, facial recognition, individual and crowd behavior analysis.

The software stack must provide the ability to allow developers to build scalable, manageable software platforms that can be remotely managed. Microservices container-based platforms such as Docker, Openstack and container management such as Kubernetes [46] and K3S [64] for resource-constrained hardware provide the infrastructure and management layers. Open-source toolkits such as YoLo provide a convolutional network framework for image recognition, [100] and Edge-X from the Linux Foundation provides an IIoT platform framework, to enable this scalability.

One of the challenges called out by Sada, et al. [73] in edge video analysis is the fragmentation of the original inference model across edge devices. They propose a federated learning platform for CNN across edge devices. Li, et al. [47] describe federated analytics as *"decentralized privacy-preserving technology to overcome challenges of data silos and data sensibility."* Deng, et al. [21] propose a federated system using neural networks spanning from the video cameras to mobile edge compute capabilities and an edge optimization capability, thereby optimizing latency and accuracy of queries to a video analytics system.

As camera resolution increases, H.264/AVC becomes less efficient, and H.265/HEVC provides a decrease the size of the bitstream by at least 50% compared to H.264/AVC, whilst supporting resolutions up to 8192x4320 with equivalent quality to H.264 [74, 87]. Tan, et al. [89] report up to 64% H.265/HEVC Bitrate deduction vs H.264/AVC for the same resolutions. H.265/HEVC does come with increased computational overheads. Sullivan, et al. [87] estimate that with more modern computing capabilities, the 40% increase in processing requirements over H.264/AVC for encoding is not a significant constraint for new equipment, but the existing install base of cameras will continue to use H.264/AVC due to compute constraints of the hardware [43].

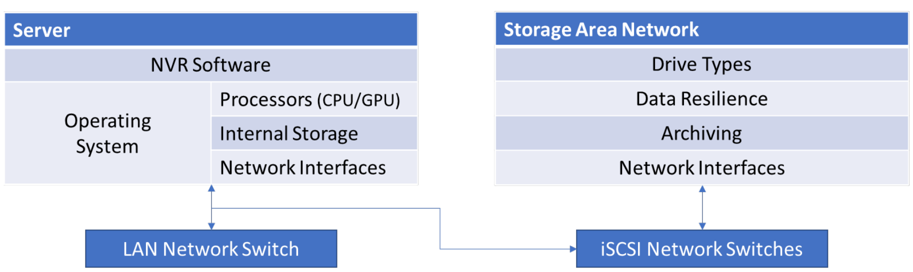


Figure 2: NVR architecture

3.1 Cloud Video Analytics

Alam, et al. [1] discusses the benefits of cloud computing and its ability to deliver platform, software, and infrastructure as a service to users. Research has identified several areas of challenges to processing streaming video analysis in the cloud. Three of the major industries using computer vision are autonomous vehicles, manufacturing and sport [51]. Mach and Becvar [56] identify some challenges of cloud computing. These technical challenges can be aligned into three main areas: connectivity, latency and security [27]. They are well documented in different vertical industries, as identified in1 Table 1.

Research from the challenges associated with cloud processing of data has focused on moving compute closer to the data source and has resulted in the emergence of edge computing capabilities. Sunyaev [88] reviews the emergence of edge computing and identifies the key goals these platforms aim to provide, overcoming the challenges posed by processing workloads in the cloud. Areas of focus for edge computing are around the hardware platforms, connectivity, and management of software to these remote devices, and the use of artificial intelligence algorithms within the software to undertake computer vision workloads. With emerging connectivity capabilities offered by 5G and the evolution of Mobile Edge Compute (MEC), new workload management platforms for edge compute such as Docker (with Kubernetes management for large deployments across edge devices), the ability to process streaming data at the edge is moving forward. Zhou, et al. [99] review the capabilities of edge platforms for AI Models to run at the edge: *"hardware acceleration technologies, such as field-programmable gate arrays (FPGAs), graphical processing units (GPUs)"*. Research by Najafi, et al. [61] suggests Application-specific integrated circuits (ASICs) offer significant promise for accelerating video analysis edge computing, and the use of smart network interface cards (SmartNiC) enables the offload of tasks from the computing platform. Emerging technologies, such as neuromorphic computing, look to overcome some challenges traditional edge hardware platforms are constrained by[58].

Cloud providers, such as Amazon Web Services, provide solutions across the globe, and in different Regions, to ensure that data sovereignty requirements can be maintained [3].

3.2 Edge Video Analytics

Moving the analytical processing of the image closer to the camera, or even onto it, can increase the performance of a system. This is especially evident where connectivity is limited or unreliable, or information from the processed data is deemed to be time-sensitive and is to be consumed at the edge, for example, real-time management of relays for complex traffic light systems. Processing can be either on the camera, Mobile Edge Compute platforms, or the cloud. The hardware required to enable a computer vision system has several separate components, outlined in Table 2, from the compute on the camera delivering specific tasks such as ANPR or motion detection using CNNs, or edge compute devices, the use of MEC to analyze data from multiple local cameras, backend cloud platform compute capabilities, with access to historical data sets for deep learning algorithms to process, the latency of the networks, and the compute capacity, in terms of memory and processor capabilities at each node in the infrastructure all play a role in identifying where the most efficient location to undertake the compute.

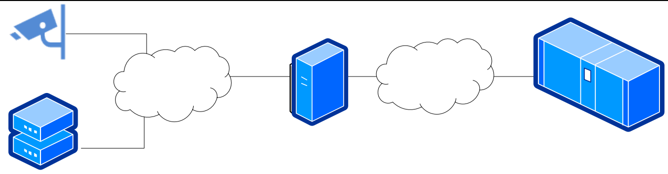
3.3 On Camera

Shi and Lichman [84] discuss cameras with "Application Specific Information Processing". The inbuilt microprocessors used to run code for specific purposes, such as motion and object detection, provide data to automated control systems.

Table 1: Cloud computing challenges

Connectivity	Sporadically connected devices and the use of streaming video data (along with lidar and radar) in autonomous vehicles to enable object detection[4], platooning [55], or to enable parking in cities [68] require reliable connectivity to the cloud. Environments with high levels of radio interference, such as manufacturing facilities [53] provide challenges to connect to cloud infrastructures.
Latency	Autonomous vehicles are highly dependent on reliable, low-latency communication, with round trip response times of under 100ms required due to the high speeds of the vehicles, especially in the realms of object detection and avoidance, and in interaction with other vehicles, such as intersection management[42]. The use of cloud analysis in the area of sports analysis, to provide statistics that can be used for presentation purposes [82] is well established. The framerate required by cameras to enable Goal Line monitoring requires localized compute to provide the referee with timely and accurate analysis and information [86]. Wireless connectivity using LTE and Wi-Fi [85] to the cloud also presents challenges where sub 100ms response times are required.
Security	The security of data flowing to the cloud, both in transit and at the final location are concerns for many cloud-based platforms. In Healthcare, patient confidentiality and protection of Individually Identifiable Health information is enshrined in standards (HIPPA, GDPR etc.) [22]. Liu, et al. [52] discusses the security requirements in vehicle to everything (V2X) autonomous vehicles in the realm of safety as the backbone to all autonomous vehicle systems.

Table 2: Processing matrix in machine vision systems

Characteristics	Edge	LTE/5G	MEC	WAN	Cloud
					
Network Latency		100ms+		75ms	
GPU Cores	240		5120		20,480
Data set Scale	GB		TB		PB

The benefits of onboard processing can reduce the bandwidth required to transmit the data from many megabits to several bytes, denoting motion or object detected. The first use of onboard compute within a camera was in the area of motion detection. While compression algorithms identify activity for prediction purposes, motion detection is used to determine the movement within the camera's view and trigger an action when identified. Sehairi, et al. [81] identified three separate categories of motion detection: Background Subtraction, Temporal Difference and Optical flow techniques, and evaluated the effectiveness of the differing algorithms. Challenges such as bad weather, thermal changes, vibration etc. [39] can cause difficulties for motion detection. Large bodies of work exist exploring the areas of false positive and false negative identification of motion detection. [82] As part of the processing of motion detection, cameras also allow for the masking of images. Masking allows regions of the image not of interest to the operator to be eliminated from processing, saving time and compute. [70] Automatic Numberplate Recognition (ANPR) or License Plate Recognition (LPR) have been in use extensively since the early 2000s [94]. They are based on Optical Character Recognition performed on video captured and streamed to a central video management system. Jeffrey, et al. [38] discuss the use of ARM-based processors and FPGAs to undertake ANPR on-camera analysis. With the increasing compute power on the camera, ANPR enabled camera algorithms can now provide descriptive feedback across the network (i.e., the number plate details), rather than just the video stream that would have to be further processed. Farhat, et al. [24] demonstrated that a Zynq-7000 programmable system on a chip (SoC) within a camera could provide ANPR recognition with a success rate of 99.5%, and with a power consumption rate 80% less than that of a Intel PC based platform undertaking the same calculations. Apostolo, et al. [4] discuss the use of video analytics to control Pan-Tilt-Zoom (PTZ) functions on cameras, allowing enabling real time actuation of PTZ functions, allowing tracking of objects of interest, enabling an active, *"human-out-of-the-loop"* automation of the camera functions.

3.4 Mobile Edge Compute

Processing IoT data closer to the data source was first discussed in 2009 [77], using virtual machines to provide 'Cloudlets' close to a 'thin' or mobile client which has limited computing capabilities [76]. The evolution of Edge computing led to ETSI launching a Mobile Edge Compute (MEC) working group in 2015 [4] with a goal to *"...enable ultra-low-latency requirements as well as a rich computing environment for value-added services closer to end users."* [31] MEC is a key component in the promise of high speed, low latency Massive IoT (MIOT) platforms described in 3GPP 5G Release 16 [30]. Baek, et al. [11] discuss 3GPP R16 and the use of mmWave [78] and MEC to provide ultra-reliable and low-latency communications (URLLC) and massive-input, massive-output (MIMO) capabilities, enabling sensor densities of up to one million sensors per square kilometer. This low latency, high-speed connectivity [65] is critical for the effective delivery of emerging technologies such as traffic management and collision avoidance systems in robotic and autonomous systems. Interlinked with the platform and communications, research into the real time processing of video streams has developed, and the emerging use of artificial intelligence (AI) for the extraction of information from streaming video. Xu, et al. [98] discuss the challenges of running AI-based video analytics on resource-constrained edge devices, such as CCTV cameras. Research into the use of deep learning algorithms [93] and federated analytics [49] are currently at the forefront of computing research. Deploying these models to the edge requires significant computing power [73]. Edge computing devices are evolving in terms of CPU, accelerators [61] and SmartNiC providing offload of networking functions [37], are enabling more complex workloads to run on edge compute platforms.

Figure 3 demonstrates the locations of the compute aspects of a surveillance system, including the on-camera, localized edge compute and MEC in 5G environments, but also the analytics backend platform, providing meta-analysis across the system, but also uses the combined datasets to train the algorithms for use at the edge, improving accuracy and ensuring that federated systems do not become fragmented, due to differing datasets flowing through the DNN.

4 Moving Cloud to the Edge

Cloud providers, such as Amazon Web Services (AWS),

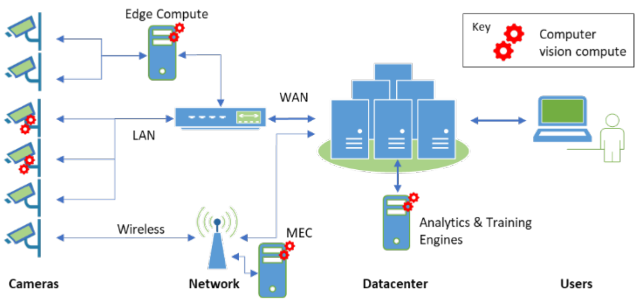


Figure 3: Edge compute

Google Cloud Platform (GCP), Microsoft Azure and VMWare Cloud, provide platform as a service (PaaS) and Software as a Service (SaaS) cloud computing platforms global availability zones across the globe. AWS, provides SaaS video analysis platform, Amazon Recognition Video. Amazon Recognition is a machine learning based video analysis platform that can be used for video analytics, including facial recognition, designed to analyze streaming video data in conjunction with AWS in real time [29].

AWS divides their cloud based platforms into physically different geographical regions, to overcome the challenges of connectivity, latency, security 37] and data sovereignty [3] with over two hundred data centers globally, AWS offers users the ability to deploy virtual machines, and SaaS platforms in twenty six separate global zones, as shown in Figure 4.

While the geographic regions provide the capabilities to ensure data sovereignty requirements be provided for, especially where PII information such as surveillance and security video footage is being processed, AWS also currently offers eighty four availability zones [8] across these geographic regions, to reduce latency between the edge and the cloud compute capabilities [12].

Despite these regions and zones, Chen and Ran [17] discuss video frame transmission latency to an AWS cloud to execute a computer vision process can take upwards of 200ms, depending on the proximity of the regional datacenter. Rao, et al. [71] suggest that the 100ms+ round trip processing time experienced in classical cloud models cannot meet the sub 10ms requirements posed for low latency video analysis using inference engines.

Cloud providers want to deliver a cloud experience to users, with consistent programming interfaces for application, consistent operations no matter where their code is executed, and a scalable, reliable platform to deliver video surveillance and analytics platforms on. The global cloud providers are building and deploying new platform models, to enable them to offer cloud services closer to or at the customers edge, where the video data is being created, with the goals to reduce latency, increase security and reliability and ensure legal data management policies such as GDRP/HIPPA etc. are delivered within the offering.

4.1 Local Datacenters

While having local regions globally will overcome issues with data sovereignty, the WAN network can cause challenges, depending on the infrastructure, bandwidth and utilization of the networks connecting the metropolitan centers to the regional cloud-based datacenters in the system.

To reduce potential latency, AWS are providing Local Zones, AWS instances that are close to metropolitan and industrial zones, to reduce the latency of the WAN [34], by bringing the compute capabilities closer to the user. Customers can then extend their existing Virtual Private Cloud (VPC) to this local zone [67], to manage the entire AWS platform as one virtual platform.

Koch and Hao [49] review the use of AWS Lambda platform and document the difference the number of network hops and distance the user is from the AWS instance. They identify that the AWS Local Zones platform performs significantly better with large data streams, than using the regional AWS cloud platforms.

Providing a reduced set of AWS services, AWS Local Zones are available currently in seventeen cities in the USA, but plans to launch local zones in thirty two cities across twenty six countries in the near future [7].

4.2 Cellular Service Provider Datacenters

As 5G infrastructure is deployed, the high speed, low latency from edge devices to the cellular network will reduce the

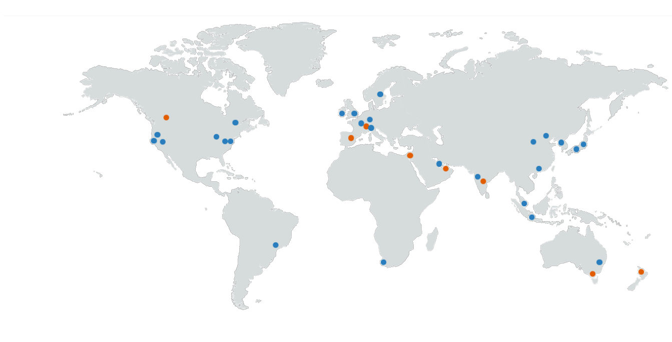


Figure 4: AWS regions [1]

challenge of latency significantly, with 3GPP URLLC specifications proposing end to end latencies of sub 5ms, not including the backhaul across the internet to the cloud platform providing the video analytics platform [36]. AWS Wavelength Zones have been designed to be physically hosted in Cellular Service Providers (CSP) local datacenters as a shared user MEC platform, on which users can run PaaS and SaaS workloads in these CSPs, while being able to manage the applications using the AWS management suite – without the data having to leave the CSPs network. This removes the latency of connections from the CSP to the AWS cloud platform. AWS Wavelength Zones are connected via high speed, low latency networks directly to the AWS regional core infrastructure using Amazon VPC to enable backhaul to other centrally hosted applications on AWS infrastructure, and carrier gateways to allow connectivity from the user application running on the Wavelength Zone access to the CSPs 5G network, or directly to the internet.

The goal of the Wavelength Zone is to provide a limited range of S3 virtual machine configurations compared to the regional or local accessibility zones, with one of these offerings being a platform designed specifically for ML and video analytics workloads.

Premkumar and Bhandari [69] discuss the use of AWS Wavelength with Verizon CSP, and demonstrates significant reduction in time taken to process streaming video while connected to a Verizon 5G network, when using AWS Wavelength, compared to the same code running on a traditional AWS S3 cloud node.

There are currently few AWS Wavelength Zones globally, with five separate CSPs, as listed in Table 3, however Amazon estimates that 34% of all mobile data traffic will move to 5G by 2030 and is aggressively deploying Wavelength Zones platforms to CSPs globally.

4.3 On Premise

The ease of use and manageability of Cloud platforms such as AWS is a key reason for users to select the ‘As a Service’ model. AWS provides a fully managed, dedicated on-premise offering to customers, called AWS Outposts. AWS Outposts can scale from a 1U rack server, to full 42U Racks of equipment, as shown in Figure 5. Using AWS Outposts, the customer only pays for the resource utilization as an operational expense, rather than the traditional capital expenditure required to deploy traditional IT enterprise platforms. Outposts is built on the same hardware platforms as deployed in the AWS regional and local datacenters globally, to ensure exactly the same hardware performance on prem or in the cloud. Other benefits also include the ability to manage and deploy workloads using the same AWS services and APIs available in the cloud, and manage these centrally across multiple sites [19]. As the platform is onsite, latency is reduced, but for mission critical and sensitive datasets such as law enforcement, surveillance and pharmaceutical research, data never leaves the confines of the user, overcoming concerns of hosting data

Table 3: AWS wavelength zones[1]

CSP	Region	Wavelength Zone
Bell	Canada	Toronto
Verizon	US East (N. Virginia)	Boston
Verizon	US East (N. Virginia)	Atlanta
Verizon	US East (N. Virginia)	Washington DC
Verizon	US East (N. Virginia)	New York City
Verizon	US East (N. Virginia)	Miami
Verizon	US East (N. Virginia)	Dallas
Verizon	US East (N. Virginia)	Houston
Verizon	US East (N. Virginia)	Chicago
Verizon	US East (N. Virginia)	Charlotte
Verizon	US East (N. Virginia)	Detroit
Verizon	US East (N. Virginia)	Minneapolis
Verizon	US West (Oregon)	San Francisco
Verizon	US West (Oregon)	Las Vegas
Verizon	US West (Oregon)	Denver
Verizon	US West (Oregon)	Seattle
Verizon	US West (Oregon)	Phoenix
Verizon	US West (Oregon)	Los Angeles
KDDI	Asia Pacific (Tokyo)	Tokyo
KDDI	Asia Pacific (Tokyo)	Osaka
SK Telecom	Asia Pacific (Seoul)	Daejeon
Vodafone	Europe (London)	London
Vodafone	Europe (Frankfurt)	Dortmund
Vodafone	Europe (Frankfurt)	Berlin
Vodafone	Europe (Frankfurt)	Munich

in the cloud [35]. While the data is hosted locally, there are connections to the AWS cloud required, both for of the workloads, and also for AWS to provide out of band

Figure 5: AWS outposts rack

[1] [9] AWS. (2022, ¼). *AWS Wavelength Zones.* Available: https://aws.amazon.com/wavelength/location.

management for the underlying infrastructure onsite, which may create security concerns and attack vectors for some customers [96].

Microsoft Azure can also extend cloud based runtime environments to a local on premise deployment using Azure Stack Edge platform, and Azure Video Analyzer software [97]. Microsoft supply and support the hardware in the same way as AWS Outposts, but users also have the option to supply certified hardware from vendors such as Dell Technologies and HPE [16], and use the cloud based Azure platform to then manage the workloads on the Azure edge platform.

4.4 Sporadically Connected / Disconnected Systems

The one thing in common with all cloud-based platforms, is the requirement to have a reliable internet connection with the capacity and latency to deliver and enable the workloads required, and on premise, or local instantiations of cloud-based platforms have similar requirements. For edge systems that have sporadic, or no connectivity to the internet, a different regime is required, especially where the user wants to leverage the capabilities cloud computing can offer. AWS Snow range of devices are ruggedized hardware platforms, originally designed to be used to transport large amounts of data from local datacenters to an AWS facility, for ingestion into an AWS Cloud VPC, but AWS Snowball also offers the capability of running Amazon Machine Images, with workloads pre-loaded [before being sent into the field. This allows machine learning and video analytics models to be ran offline, outside of the datacenter [91].

Figure 6: AWS snowball

Pawloski, et al. [66] demonstrated the use of AWS Snowball platform to provide off-grid, ruggedized compute capabilities, in which the workloads could be loaded onto the system before deployment, for use in areas of natural or man-made disasters, where reliable internet connectivity may be sporadic, or not available.

5 Conclusions

The use of edge compute capabilities, combined with modern coding and management capabilities, can overcome challenges with network latency and enable real-time, preventative surveillance solutions for law enforcement. The reduction in compute cost and the emergence of lightweight neural network algorithms for computer vision can allow resource-constrained edge compute nodes to deliver an accurate analysis of streaming data in a timely manner.

The emergence of data-focused wireless technologies such as 5G, with mobile edge compute capabilities built into the core design of the networks to provide ultra-low latency analysis of the video data, will drive more analysis out of the central and cloud data centers. Moving these compute capabilities closer to the source of the data on edge devices will provide benefits to deliver surveillance solutions. The removal of latency due to network backhaul to cloud platforms will improve decision making processes locally in time-critical applications, such as facial recognition for law enforcement. Cloud providers are providing new capabilities combining the ease of deployment and management of workloads in the cloud, on platforms at, or closer to the edge of the network. Emerging technologies provide the capability to analyze the video stream at the edge using autonomous decision making provided by neural network algorithms to decide when data should be transmitted. These capabilities can enable proactive interaction and intervention by users or for evidential purposes.

References

[1] A. Alam, I. Ullah, and Y.-K. Lee, "Video Big Data Analytics in the Cloud: A Reference Architecture, Survey, Opportunities, and Open Research Issues," *IEEE Access,* 8:152377-152422, 2020.

[2] A. Aliyu, A. H. Abdullah, O. Kaiwartya, S. H. H. Madni, U. M. Joda, A. Ado, and M. Tayyabl, "Mobile Cloud Computing: Taxonomy and Challenges," *Journal of Computer Networks and Communications,* 2020:10, 2020.

[3] L. Amoore, "Cloud Geographies: Computing, Data, Sovereignty," *Progress in Human Geography,* 42(1):4-24, 2018.

[4] G. H. Apostolo, P. Bauszat, V. Nigade, H. E. Bal, and L. Wang," Live Video Analytics as a Service, " *Proceedings of the 2nd European Workshop on Machine Learning and Systems*, pp. 37-44, 2022.

[5] M. P. Ashby, "The Value of CCTV Surveillance Cameras as an Investigative Tool: An Empirical Analysis," *European Journal on Criminal Policy and Research,* 23(3):441-459, 2017.

[6] Avigilon, *System Requirements - Avigilon Control Center Software*, Available: https://www.avigilon.com/products/acc/system-requirements, 2020.

[7] AWS, *AWS Local Zones Locations*. Available: https://aws.amazon.com/about-aws/global-infrastructure/localzones/locations/?nc=sn&loc=3. 4/5 2022.

[8] AWS. "*AWS Regions & Availability Zones*," Available: https://aws.amazon.com/about-aws/global-infrastructure/regions_az/, 04/05, 2022.

[9] AWS, *AWS Wavelength Zones*. Available: https://aws.amazon.com/wavelength/locations/, 1/4, 2022.

[10] Axis, *Axis NetEye 200*, Available: https://www.axis.com/ files/datasheet/200/neteye_200_ds.pdf, 1/4, 1996.

[11] S. Baek, D. Kim, M. Tesanovic, and A. Agiwal, "3GPP New Radio Release 16: Evolution of 5G for Industrial Internet of Things," *IEEE Communications Magazine,* 59(1):41-47, 2021.

[12] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and Its Role in the Internet of Things," *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, pp. 13-16, 2012.

[13] S. Borge and N. Poonia, "Review on amazon web services, google cloud provider and microsoft windows azure," *Advance and Innovative Research,* p. 53, 2020.

[14] G. Di Caterina, I. Hunter, and J. J. Soraghan, "An Embedded Smart Surveillance System for Target Tracking Using a PTZ Camera," *4th European Education and Research Conference (EDERC 2010)*, IEEE, pp. 165-169, 2010.

[15] A. Caulfield, P. Costa, and M. Ghobadi, "Beyond SmartNICs: Towards a Fully Programmable Cloud," *2018 IEEE 19th International Conference on High Performance Switching and Routing (HPSR)*, IEEE, pp. 1-6, 2018.

[16] H. Chawla and H. Kathuria, "Building Microservices Applications on Azure Stack," *Building Microservices Applications on Microsoft Azure*: Springer, pp. 245-255, 2019.

[17] J. Chen and X. Ran, "Deep Learning With Edge Computing: A Review," *Proc. IEEE,* 107(8):1655-1674, 2019.

[18] R. Dautov, S. Distefano, D. Bruneo, F. Longo, G. Merlino, A. Puliafito, and R. Buyya, "Metropolitan Intelligent Surveillance Systems for Urban Areas by Harnessing IoT and Edge Computing Paradigms," *Software: Practice and Experience,* 48(8):1475-1492, 2018.

[19] M. Deb and A. Choudhury, "Hybrid Cloud: A New Paradigm in Cloud Computing," *Machine Learning Techniques and Analytics for Cloud Security,* pp. 1-23, 2021.

[20] Dell_Technologies,. *Dell EMC Streaming Data Platform*, Available: https://www.delltechnologies.com/ en-us/storage/streaming-data-platform.htm, 03-19, 2020.

[21] Y. Deng, T. Han, and N. Ansari, "FedVision: Federated Video Analytics With Edge Computing," *IEEE Open Journal of the Computer Society,* 1:62-72, 2020.

[22] A. A. Elgujja, "Impact of Information Technology on Patient Confidentiality Rights: A Perspective," *Impacts of Information Technology on Patient Care and Empowerment*: IGI Global, pp. 365-387, 2020.

[23] O. Elharrouss, N. Almaadeed, and S. Al-Maadeed, "A Review of Video Surveillance Systems," *Journal of Visual Communication and Image Representation,* 77:103116, 2021.

[24] A. Farhat, O. Hommos, A. Al-Zawqari, A. Al-Qahtani, F. Bensaali, A. Amira, and X Zhai,. "Optical Character Recognition on Heterogeneous SoC for HD Automatic Number Plate Recognition System," *EURASIP Journal on Image and Video Processing,* 2018(1):58, 2018.

[25] M. Fayaz and D. Kim, "An Effective Control Method of IP Camera Based on Fuzzy Logic and Statistical Moments," *International Journal of Control and Automation,* 10:97-108, 2017.

[26] S. Feldstein, *The Global Expansion of AI Surveillance*, Carnegie Endowment for International Peace, Washington, DC, 2019.

[27] M. Firdhous, O. Ghazali, and S. Hassan, "Fog Computing: Will It be the Future of Cloud Computing?," The Third International Conference on Informatics & Applications (ICIA 2014), 20-14.

[28] W. Fischer,"Video Coding (MPEG-2, MPEG-4/AVC, HEVC), " *Digital Video and Audio Broadcasting Technology*, Springer, pp. 125-175, 2020.

[29] D. V. Ganesan, S. Das, T. Kundu, J. Naren, and S. Bushra, "Deep Learning based Smart Survilance Robot," 2021.

[30] A. Ghosh, A. Maeder, M. Baker, and D. Chandramouli, "5G Evolution: A View on 5G CellularTechnology beyond 3GPP Release 15," *IEEE Access,* 7:127639-127651, 2019.

[31] F. Giust, X. Costa-Perez, and A. Reznik, "Multi-Access Edge Computing: An Overview of ETSI MEC ISG," *IEEE 5G Tech Focus,* 1(4):4, 2017.

[32] S. Hadavi, H. B. Rai, S. Verlinde, H. Huang, C. Macharis, and T. Guns, "Analyzing Passenger and Freight Vehicle Movements from Automatic-Number Plate Recognition Camera Data," *European Transport Research Review,* 12:1-17, 2020.

[33] F. Henry, "Review of Ferguson, Andrew Guthrie, 2017," *The Rise of Big Data Policing: Surveillance, Race, and the Future of Law Enforcement*, New York: New York University Press, 259 pages, Paperback ISBN: 9781479892822, Ed: Springer, 2018.

[34] M. Herson and K. King, "Remote Innovations on the Cloud," *SMPTE Motion Imaging Journal,* 130(8):98-100, 2021.

[35] J. Hildén, "Mitigating the Risk of US Surveillance for Public Sector Services in the Cloud," *Internet Policy Review,* 10(3):1-24, 2021.

[36] E. Iradier, A. Abuin, L. Fanari, J. Montalban, and P. Angueira, "Throughput, Capacity and Latency Analysis of P-NOMA RRM Schemes in 5G URLLC," *Multimedia Tools and Applications,* pp. 1-23, 2021.

[37] H. Iqbal, A. Singh, and M. Shahzad, "Characterizing the Availability and Latency in AWS Network From the Perspective of Tenants," *IEEE/ACM Transactions on Networking,* 2022.

[38] Z. Jeffrey, X. Zhai, F. Bensaali, R. Sotudeh, and A. Ariyaeeinia, "Automatic Number Plate Recognition System on an ARM-DSP and FPGA Heterogeneous SoC Platforms," *2013 IEEE Hot Chips 25 Symposium (HCS)*, IEEE, pp. 1-9, 2013.

[39] H. Jung, J. Ju, W. Hwang, and J. Kim, "Refining Background Subtraction using Consistent Motion Detection in Adverse Weather," *Journal of Electronic Imaging,* 28(2):020501, 2019.

[40] A. A. Juri and A. B. Jambek, “Review on the latest H. 264 Motion Estimation Techniques,” *2011 International Conference on Electronic Devices, Systems and Applications (ICEDSA)*, IEEE, pp. 199-202, 2011.

[41] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada, “An Open Approach to Autonomous Vehicles,” *IEEE Micro,* 35(6):60-68, 2015.

[42] M. Khayatian, Y. Lou, M. Mehrabian, and A. Shirvastava, “Crossroads+ a Time-Aware Approach for Intersection Management of Connected Autonomous Vehicles,” *ACM Transactions on Cyber-Physical Systems,* 4(2):1-28, 2019.

[43] G. Kioumourtzis, M. Skitsas, N. Zotos, and A. Sideris, “Wide Area Video Surveillane Based on Edge and Fog Computing Concept,” *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)*, IEEE, pp. 1-6, 2017.

[44] G. I. Klas, “Fog Computing and Mobile Edge Cloud Gain Momentum Open Fog Consortium, ETSI MEC and Cloudlets,” *Google Scholar,* 2015.

[45] J. Koch and W. Hao, “An Empirical Study in Edge Computing using AWS,” *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, pp. 0542-0549, 2021.

[46] E. Kristiani, C.-T. Yang, Y. T. Wang, and C.-Y. Huang, “Implementation of an Edge Computing Architecture using Openstack and Kubernetes,” *International Conference on Information Science and Applications*, Springer, pp. 675-685, 2018.

[47] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, “A Review of Applications in Federated Learning,” *Computers & Industrial Engineering,* p. 106854, 2020.

[48] J. Li, J. Jin, D. Yuan, M. Palaniswami, and K. Moessner, “EHOPES: Data-Centered Fog Platform for Smart Living,” *2015 International Telecommunication Networks and Applications Conference (ITNAC)*, IEEE, pp. 308-313, 2015.

[49] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y-C. Liang, Q. Yang, D. Niyato, and C. Miao, “Federated Learning in Mobile Edge Networks: A Comprehensive Survey,” *IEEE Communications Surveys & Tutorials,* 22(3):2031-2063, 2020.

[50] L. Lin, X. Liao, H. Jin, and P. Li, “Computation Offloading toward Edge Computing,” *Proceedings of the IEEE,* 107(8):1584-1607, 2019.

[51] S. Liu, “Artificial Intelligence Software Market Revenue Worldwide 2018-2025,” Ed, 2019.

[52] S. Liu, L. Liu, J. Tang, B. Yu, Y. Wang, and W. Shi, "Edge Computing for Autonomous Driving: Opportunities and Challenges,” *Proceedings of the IEEE,* 107(8):1697-1716, 2019.

[53] X. Liu, X. Wei, L. Guo, Y. Liu, Q. Song, and A. Jamalipour, “Turning the Signal Interference into Benefits: Towards Indoor Self-Powered Visible Light Communication for IoT Devices in Industrial Radio-Hostile Environments," *IEEE Access,* 7:24978-24989, 2019.

[54] L. Liu, X. Zhang, Q. Zhang, A. Weinert, and Y. Wang, “AutoVAPS: An IoT-Enabled Public Safety SDervice on Vehicles,” *Proceedings of the Fourth Workshop on International Science of Smart City Operations & Platform Engineering*, pp. 41-47, 2019.

[55] D. Lu, Z. Li, and D. Huang, “Platooning as a Service of Autonomous Vehicles,” 2017 *IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 1-6, 2017.

[56] P. Mach and Z. Becvar, “Mobile Edge Computing: A Survey on Architecture and Computation Offloading,” *IEEE Communications Surveys & Tutorials,* 19(3):1628-1656, 2017.

[57] A. A. Mathews and A. Babu, “Automatic Number Plate Detection,” *Journal of Electronics and Communication Engineering (IOSR-JECE),* 2:50-55, 2017, https://www.iosrjournals.org/iosr-jece/pages/ICRTEM-volume2.html, 2017.

[58] Milestone-Systems, *Smart Client System Requirements*, Available: https://www.milestone sys.com/support/ tools-and-references/system-requirements/, 2020.

[59] P. Merolla, J. Arthur, F. Akopyan, N. Imam, R. Manohar, and D. S. Modha, “A Digital Neurosynaptic Core using Embedded Crossbar Memory with 45pJ per Spike in 45nm,” *2011 IEEE Custom Integrated Circuits Conference (CICC)*, IEEE, pp. 1-4, 2011.

[60] D. Minoli, K. Sohraby, and B. Occhiogrosso, “IoT Considerations, Requirements, and Architectures for Smart Buildings—Energy Optimization and Next-Generation Building Management Systems,” *IEEE Internet of Things Journal,* 4(1):269-283, 2017.

[61] M. Najafi, K. Zhang, M. Sadoghi, and H.-A. Jacobsen, “Hardware Acceleration Landscape for Distributed Real-Time Analytics: Virtues and Limitations,” *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, IEEE, pp. 1938-1948, 2017.

[62] J. Niu, R. Fie, and J. Hu, “Mpeg-4 Video Encoder Based on DSP-FPGA Techniques,” *Proceedings. 2005 International Conference on Communications, Circuits and Systems*, IEEE, 1:518-522, 2005.

[63] ONVIF, *ONVIF Mission*. Available: https://www.onvif.org/about/mission/, 4th April, 2017.

[64] P. Pääkkönen, D. Pakkala, J. Kiljander, and R. Sarala, “Architecture for Enabling Edge Inference via Model Transfer from Cloud Domain in a Kubernetes Environment,” *Future Internet,* 13(1):5, 2021.

[65] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, “A Survey on Low Latency Towards 5G: RAN, Core Network and Caching Solutions,” *IEEE Communications Surveys & Tutorials,* 20(4):3098-3130, 2018.

[66] A. Pawloski, D. Pilone, J. Siarto, P. Pilone, T. Lang, and C. Fayock, “Improving Information and Communica-tions in a Disaster Scenario with AWS Snowball Edge,” *AGU Fall Meeting Abstracts*, 2019:IN23B-09, 2019.

[67] J. Peng and X. Liu, "Disease and Death Monitoring on Amazon Managed Streaming for Apache Kafka,” *Proceedings of the 2nd International Symposium on Artificial Intelligence for Medicine Sciences*, pp. 24-27,

2021.

[68] T. N. Pham, M.-F. Tsai, D. B. Nguyen, C.-R. Dow, and D.-J. Deng, "A Cloud-Based Smart-Parking System Based on Internet-of-Things Technologies," *IEEE Access,* 3:1581-1591, 2015.

[69] V. Premkumar and V. Bhandari, "Ultra-Low Latency Video Streaming using Verizon 5G and AWS Wavelength," *International Journal of Management IT and Engineering,* 11(5):29-33, 2021.

[70] J. L. Randall, "Security System with Maskable Motion Detection and Camera with an Adjustable Field of View," Ed: Google Patents, 2004.

[71] K. Rao, G. Coviello, W.-P. Hsiung, and S. Chakradhar, "ECO: Edge-Cloud Optimization of 5G Applications," *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, IEEE, pp. 649-659, 2021.

[72] M. I. Razzak, S. Naz, and A. Zaib, "Deep Learning for Medical Image Processing: Overview, Challenges and the Future," *Classification in BioApps,* pp. 323-350, 2018.

[73] A. B. Sada, M. A. Bouras, J. Ma, H. Runhe, and H. Ning, "A Distributed Video Analytics Architecture Based on Edge-Computing and Federated Learning," *2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing*, IEEE, pp. 215-220, 2019.

[74] M. A. Saleh, H. Hashim, N. M. Tahir, and E. Hisham, "Review for High Efficiency Video Coding (HEVC)," *2014 IEEE Conference on Systems, Process and Control (ICSPC 2014)*, IEEE, pp. 141-146, 2014.

[75] M. Samatas, "From Thought Control to Traffic Control: CCTV Politics of Expansion and Resistance in post-Olympics Greece," *Sociology of Crime Law and Deviance,* 10:345-369, 2008.

[76] A. Sammoud, A. Kumar, M. Bayoumi, and T. Elarabi, "Real-Time Streaming Challenges in Internet of Video Things (IoVT)," *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, IEEE, pp. 1-4, 2017.

[77] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The Case for VIN-Based Cloudlets in Mobile Computing," *IEEE Pervasive Computing,* 8(4):14-23, 2009.

[78] N. Saxena, A. Roy, B. J. Sahu, and H. Kim, "Efficient IoT Gateway over 5G Wireless: A New Design with Prototype and Implementation Results," *IEEE Communications Magazine,* 55(2):97-105, 2017.

[79] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview," *Neural Networks,* 61:85-117, 2015.

[80] D. Seckiner, X. Mallett, C. Roux, D. Meuwly, and P. Maynard, "Forensic Image Analysis–CCTV Distortion and Artifacts," *Forensic Science International,* 285:77-85, 2018.

[81] K. Sehairi, F. Chouireb, and J. Meunier, "Comparative Study of Motion Detection Methods for Video Surveillance Systems," *Journal of Electronic Imaging,* 26(2):023025, 2017.

[82] M. J. Shafiee, P. Siva, P. Fieguth, and A. Wong, "Real-Time Embedded Motion Detection via Neural Response Mixture Modeling," *Journal of Signal Processing Systems,* 90(6):931-946, 2018.

[83] W. Shi, G. Pallis, and Z. Xu, "Edge Computing [Scanning the Issue]," *Proceedings of the IEEE,* 107(8):1474-1481, 2019.

[84] Y. Shi and S. Lichman, "Smart Cameras: A Review," *Proceedings of Citeseer*, pp. 95-100, 2005

[85] H.-C. Shih, "A Survey of Content-Aware Video Analysis for Sports," *IEEE Transactions on Circuits and Systems for Video Technology,* 28(5):1212-1231, 2017.

[86] P. Spagnolo, P. L. Mazzeo, M. Leo, M. Nitti, E. Stella, and A. Distante, "On-Field Testing and Evaluation of a Goal-Line Technology System," in *Computer Vision in Sports*: Springer, pp. 67-90, 2014.

[87] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology,* 22(12):1649-1668, 2012.

[88] A. Sunyaev, "Fog and Edge Computing," *Internet Computing*, Springer, pp. 237-264, 2020.

[89] T. Tan, M. Mrak, V. Baroncini, and N. Ramzan, "Report on HEVC Compression Performance Verification Testing," *Joint Collab. Team Video Coding (JCT-VC),* 2014.

[90] J. Tate, P. Beck, H. H. Ibarra, S. Kumaravel, and L. Miklas, *Introduction to Storage Area Networks*, IBM Redbooks, 2018.

[91] V. Upadrista, "The IoT Standards Reference Model " *IoT Standards with Blockchain*: Springer, pp. 61-86, 2021.

[92] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep Learning for Computer Vision: A Brief Review," *Computational Intelligence and Neuroscience,* 2018:4-6, 2018.

[93] F. Wang, M. Zhang, X. Wang, X. Ma, and J. Liu, "Deep Learning for Edge Computing Applications: A State-of-the-Art Survey," *IEEE Access,* 8:58322-58336, 2020.

[94] B. C. Watson and K. M. Walsh, "The Road Safety Implications of Automatic Number Plate Recognition Technology (ANPR)," The Centre for Accident Research and Road Safety, Queensland, Available Online: https://eprints.qut.edu.au/13222/, accessed 16 January 2022

[95] C. A. Williams, "Police Surveillance and the Emergence of CCTV in the 1960s," *Crime Prevention and Community Safety,* 5(3):27-37, 2003.

[96] L. Wu, X. Du, W. Wang, and B. Lin, "An Out-of-Band Authentication Scheme for Internet of Things using Blockchain Technology," *2018 International Conference on Computing, Networking and Communications (ICNC)*, IEEE, pp. 769-773, 2018.

[97] Y. Xiong, Y. Sun, L. Xing, and Y. Huang, "Extend Cloud to Edge with Kubeedge," *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, IEEE, pp. 373-377, 2018.

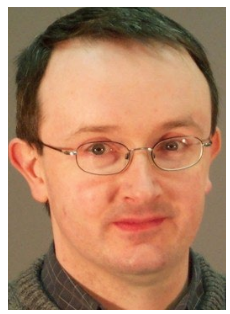
[98] R. Xu, R. Kumar, P. Wang, P. Bai, G. Meghanath, S. Chaterji, S. Mitra, and S. Bagchi, "ApproxNet: Content and Contention Aware Video Analytics System for the Edge," *arXiv preprint arXiv:1909.02068,* 2019.

[99] X. Zhou, R. Canady, S. Bao, and A. Gokhale, "Cost-Effective Hardware Accelerator Recommendation for Edge Computing," *3rd {USENIX} Workshop on Hot Topics in Edge Computing (HotEdge 20)*, 2020.
[100] X. Zou, "A Review of Object Detection Techniques," *2019 International Conference on Smart Grid and Electrical Automation (ICSGEA)*, IEEE, pp. 251-254, 2019.

Jeff McCann received his B.Sc. (Hons) from the Ulster University in Computer Science in 1994, and master's in project management from the University of Limerick in 2006. Jeff leads the global strategy for the Dell Technologies Customer Solution Centers, is an adjunct lecturer in IoT at the University of Limerick Electronics and Computer, sits on the industrial advisory committee of the Confirm Industry 4.0 Science Foundation Ireland Research center, and provides input to numerous industry and standards bodies, including National Standards Association of Ireland (NSAI) IoT Architecture & Smart Cities Working Groups, Smart Limerick Leaders Council, IEEE BizOps for IoT, American Chamber in Ireland working group on Industry 4.0, and the Health Service Executive Digital Academy.

Liam Quinn received a BSc in Electrical Engineering from the Dublin Institute of Technology in 1979, a BSc in Computer Systems from the University of Limerick in 1982, an MSc in Computer Engineering from Boston University in 1990. Liam is a Sr Fellow at Dell Technologies where he leads the strategies for 5G and Edge computing. He served as an adjunct lecturer in Electrical Engineering at the Dublin Institute of Technology, and in networking and mobile computing at the University of Texas and Texas State University. Liam is a board member of the Wi-Fi Alliance and is a published author in Networking and IoT in Smart Cities. He holds over 150 granted and pending US patents.

Sean McGrath received his MSC in Heriot-Watt University in 1986 and his PhD in 1992 from the University of Limerick in the area of indoor wireless data communication systems. He has worked in mobile communication research for the last twelve years. He lectures in Telecommunication systems and digital communication. He is a member of the Committee of the IEEE Communications Chapter and IEEE Committee on Information Theory. He has presented and published a large number of research papers. His research interests include mobile communications, communication theory and its applications, and satellite systems.



Colin Flanagan received his B. Eng. (hons) from the National Institute for Higher Education, Limerick, in 1986. In 1988 he received an M. Eng. from the University of Limerick for work on computer architecture, and in 1991 his PhD, in the area of neural networks. He has worked in A.I. and networking research for the past thirty years, and has published extensively in conferences and refereed journals. His research interests include the application of artificial intelligence and deep learning, particularly in medicine, computer architecture and quantum computing.

Application of Machine Learning on Software Quality Assurance and Testing: A Chronological Survey

Mohammad Hossain*
University of Minnesota Crookston, MN, USA

Hongkai Chen†
University of California San Diego, CA, USA

Abstract

Ensuring the quality is essential for a successful Software System. Software systems need to be tested in every stage of the Software Development Life Cycle (SDLC) irrespective of the type of software being developed. If a software bug remains undetected in the early phase of the SDLC, it becomes harder to fix it at a later stage and becomes very costly. The application of machine learning in Software Quality Assurance and Testing can help testers in the testing process, including the early detection and prediction of a software bug. However, employing machine learning techniques brings new challenges to testing and quality assurance. Machine Learning (ML) uses Artificial Intelligence (AI) techniques that focus on a given dataset to find any trend present in the data. It has been observed that some software testing activities can, in fact, be represented as a learning problem. Thus, ML can be used as an efficient tool to automate software-testing activities, especially when the software system becomes very complex. This survey aims to study and summarize the application of machine learning on software quality assurance and testing in a chronological manner by selecting from articles published in the last twenty-six years or so.

Key Words: Software quality assurance and testing, machine learning, artificial intelligence, chronological survey, neural network, support vector machine.

1 Introduction

Machine learning (ML) is the study of computer algorithms designed to exhibit intelligence by self-learning through experience observed from its surrounding environment. Machine learning is a branch of artificial intelligence. The basic idea of machine learning is to build a model based on sample data or training data to predict or make decisions without programming. Today, machine learning is widely used in many industries and applications, including pattern recognition, computer vision, aeronautical engineering, finance, entertainment, computational biology, biomedical engineering, and medical applications [3].

Software quality assurance and testing refer to testing the software and ensuring the proper quality of the software. It is a crucial phase of SDLC in terms of time and money. Much effort has been taken to reduce the cost of the testing phase to keep the software development cost within the budget. For that, machine learning has been introduced in software testing for a long time [12, 45]. However, the use of machine learning brings some new challenges to the quality assurance and testing field. At the same time, it also provides some potential new methods for software testing. This paper will discuss the application of machine learning in software quality assurance and testing.

In the following section, we discussed our methodology for this study. The next section summarizes our findings in various subsections as found in different year groups. Finally, the following section analyzed our findings on various machine learning techniques.

2 Methodology

For this survey, we collected fifty different papers published from 1995 to 2021. We tried to investigate the trend that might be found in utilizing artificial intelligence in general and machine learning in particular in software testing and quality assurance over the period of the last twenty-six years or so. We grouped the papers into five-year periods and tried to focus on the main discussion topics of each group. Table 1 shows the papers that we gathered for this purpose.

3 Summaries of Our Findings

This section listed the papers that we studied in chronological order in the following subsections. We tried to group the papers uniformly in a five-year span with the exception of 3.5 and 3.6 to keep the number of papers in each group somewhat uniform. We discussed the authors' motivation for their research and their findings in each subgroup.

3.1 1995-1999

In this period, we have studied two papers. In [3], the authors talked about the problem of a large volume of test cases generated by automated tools as the effectiveness of these test cases is not clear. The authors present experimental results on using a neural network for pruning a test case set while preserving its effectiveness. The authors concluded that based

* hossain@crk.umn.edu.
† hongkai@ucsd.edu.

Table 1: Articles sorted by year with key words

Year	Number of Articles	Keywords	References
1995	2	Neural Network (NN)	[3], [12]
2002	1	Neural Network	[45]
2003	1	Software Engineering	[50]
2004	1	Active Learning, Automatic Classification, Markov models	[6]
2006	3	Infeasible Paths, Data Flow Based Testing, Classification, Value-Based Software, Test Data Generation, Decision Tree, Genetic Algorithms	[11], [46], [49]
2007	2	Support Vector Machines, MartiRank, Test Oracle, Statistical Software Testing	[5], [32]
2008	4	Decision trees, Fault-proneness prediction, Test Oracles Generation, NN, Support Vector Machines, Defect-prone Software Modules	[8], [7], [19], [25]
2009	1	Category-Partition, Black Box Testing	[9]
2010	2	Bayesian Reasoning, Asymmetric Function, NN	[35], [41]
2011	4	Classification Framework, Metamorphic Testing, GUI Testing, Test Oracle, Support Vector Machines, Grammar Induction, Clustering, Regression Test	[13], [20], [36], [48]
2012	5	Black Box Testing, Clustering, GUI Testing, NN, Test Oracle, Mutation Testing	[1], [21], [22], [32], [43]
2013	3	Test Coverage Criteria, Combining Testing Techniques, Data Mining	[14], [28], [47]
2016	1	Metamorphic Relations, Graph Kernels	[26]
2017	3	Aging Related Bug, Data Mining, Big Data, Metamorphic Testing	[31], [32], [33]
2018	4	Test Oracle, Dataset Diversity, Metamorphic Testing	[34], [35], [36], [37]
2019	8	Test Case Generation, Classification, Clustering, Test Automation, NN, Fault Localization	[38], [39], [40], [41], [42], [43], [44], [45]
2020	4	Data Cleaning, NN, Code Review, Continuous Integration	[46], [47], [48], [49]
2021	1	Statistical Regression	[50]

on their experiment, neural networks are promising test case effectiveness predictors. They further concluded that their method is able to adapt as the software matures with sufficient accuracy. In [12], the authors tried to accurately estimate the cost of software testing. Authors applied machine learning techniques to determine the software testing attributes that are important in predicting software testing costs and time.

Figure 1 shows the distribution of the articles over the years.

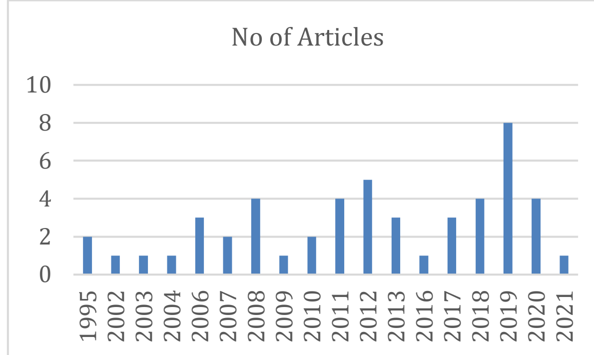


Figure 1: Distribution of the articles over the years

3.2 2000-2004

In this period, we selected three papers [6, 45, 50] to discuss. In [45], the authors talked about a test oracle to determine whether a given test case exposes a fault or not. This paper presents a new concept of using a two-layered artificial neural network as

an automated oracle for testing a real software system. The authors concluded that the neural network is a promising method of testing a software application capable of learning new versions of evolving software. In [50], the authors discussed the domain of the machine learning approach and how it can be utilized in software engineering. They showed how the software development and maintenance tasks could be formulated as learning problems. In [6], the authors focused on the automatic classification of program behavior using execution data. They introduced a technique that models program executions as Markov models and also devised a clustering method for Markov models to combine multiple program executions to form an effective behavior classifier.

3.3 2005-2009

We found ten papers [5, 8, 9, 11, 17, 19, 25, 32, 46, 49] to study in this period. In [46], the authors spent on infeasible paths, basically on three main approaches: prediction, classification, and identification of infeasibility. They also addressed these aspects in the scope of integration and object-oriented testing. The authors claimed that the finding of this paper would aid in the planning of the testing activity and in the establishment of testing strategies. In [49], the authors proposed a framework for value-based software test data generation. Value-based software engineering considers value into the software engineering principles and practices as opposed to value-neutral software engineering where each product in software development, such as requirement, use case, test case, and defect, is treated as equally important. The authors talked about applying machine learning methods to value-based software engineering in this paper. In [11], the authors analyzed the problems in software code and proposed a model that will help catch those problems earlier in the project life cycle using machine learning methods. In [32], authors discussed issues of testing machine learning applications and proposed machine learning algorithms such as SVM and MurtiRank. In [5], the authors presented an adaptive sampling mechanism using machine learning for software testing. In [8], the author provided a brief overview of state of art and reports on a number of novel applications of machine learning in the area of software testing. In [19], the author argued that there is no general technique for estimating software fault-proneness. In this paper, the author proposed the use of machine learning for software fault-proneness prediction. In [25], the authors talked about the artificial neural network for test oracles generation, the same theme as seen in [45]. In [17], the authors evaluated the capability of SVM in predicting defect-prone software modules and claimed that the prediction performance of SVM is generally better than the models they compared with. In [9], the authors proposed a methodology and a tool based on machine learning to help people understand the limitations of test suites and their possible redundancies so that people are able to refine them in a cost-effective manner. The authors claimed that their proposed solution to show promising results on a case study involving students as testers.

3.4 2010-2014

In this period, we found fourteen papers [1, 13-14, 20-22, 28, 32, 35-36, 41, 43, 47-48] to study. In [35], the authors provided a brief overview of some popular Bayesian reasoning methods for achieving reliable and efficient software testing and program analysis. They also explained why those Bayesian reasoning methods are applicable to software testing. In [41], the authors focused on the application of machine learning tools and variable selection tools in order to solve the problem of estimating the execution effort of functional tests. The authors concluded that there is a high complexity of the effort estimation problem. Managers and other testing professionals require easy-to-use and efficient estimation methods. In [36], the authors introduced a classification framework that can help to systematically review research work in the ML and software testing domains. This framework can be used to construct guidelines for choosing the most appropriate learning method and then using it in the software testing stage. The authors claimed that the classification framework is quite strong in capturing various aspects of work in ML and software testing. Furthermore, it helps researchers systematically investigate and extract the prominent information from existing research works in ML and software testing. In [48], the authors presented a technique based on "metamorphic testing" for testing the implementations of machine learning classification algorithms. The authors claimed that their approach enables users and programmers to easily and effectively verify and validate the machine learning components of the software. In [20] and [21], the authors talked about GUI software testing. They proposed avoiding infeasible test cases altogether by predicting which test cases are infeasible using two supervised machine learning methods: support vector machines (SVMs) and grammar induction. The authors concluded that classifying test case feasibility is possible. In [13], the authors introduced a semi-supervised clustering method named semi-supervised K-means (SSKM) to improve cluster test selection. The authors claimed that the semi-supervised clustering method SSKM could improve test selection in most cases. In [32], the authors discussed software quality improvement by early prediction of error patterns. They advocated the use of case-based reasoning (i.e., CBR) to build a software quality prediction system with the help of human experts. In [1], the authors compared the use of artificial neural networks (ANN) and info-fuzzy networks (IFN) as automated oracles to confirm that the developed software complies with its specification and determine whether a given test case exposes faults. The authors concluded that IFN outperforms the ANN for faults causing a large number of faulty records, while the ANN appears to be more suitable for identifying hard-to-detect faults in more stable versions. The IFN clearly outperforms the ANN with respect to training time. In [43], the authors proposed an approach to classifying mutants as a tool to reduce the number of mutants to be executed and to evaluate the quality of test suits without executing them against all possible mutants. They concluded that the results obtained so far are encouraging, but this approach still needs more experiments to fully confirm its validity. In [22], the authors discussed some of the relationships between the work of Artificial Intelligence (AI) techniques and Software Engineering (SE) problems. In [28], the authors proposed an approach using machine learning techniques to link test results from the application of different testing techniques. The authors claimed that the major advantage of this approach is that it can automatically determine equivalence classes, which are, in general, manually determined according to subjective rules. In [14], the authors presented a method that can combine testing techniques adaptively during the testing process. The authors concluded that this method could improve the fault detection effectiveness with respect to single testing techniques and their random combination. In [47], the authors presented cooperative testing and analysis, including human-tool cooperation and human-human cooperation. The authors claimed that this could reduce human efforts and burden in software engineering activities.

3.5 2015-2018

In this period, we found eight papers [15, 18, 24, 26-27, 30, 33-34] to study. In [26], authors conducted a feature analysis to identify the most effective features for predicting metamorphic relations for testing scientific software using graph kernels. In [27], the authors presented a study on the application of machine learning techniques and static source code metrics to predict aging-related bugs. The authors concluded that static source code metrics could be used as predictors for aging-related bugs. In [18], the authors discussed many different works in the field of software vulnerability analysis and discovery that utilize machine-learning and data-mining techniques. In [15], the authors presented a framework for validating the large-scale image data as well as adequately verifying both the software tools and machine learning algorithms. They claimed that the framework addresses the most important issues of verification and validation in big data. In [30], the authors discussed problems with ML applications and discovered software engineering approaches and software testing research areas to solve these problems. They concluded some key areas such as deep learning, fault localization, and prediction. In [24], the authors discussed the characteristics of some machine learning algorithms and concluded the main challenges of testing machine learning applications. Meanwhile, the authors presented two techniques: comparing results of different implementations and metamorphic testing to mitigate the test oracle problem. In [33], the author reviewed two traditional views of service and product qualities of a machine learning software. In [34], the authors demonstrated a new metamorphic testing method that can be used to test neural network learning models. This method mainly lies on dataset diversity and behavioral oracle. The authors conjectured that their approach could be effective in the software testing of machine learning programs.

3.6 2019-2021

In this period, we selected thirteen papers to discuss [2, 7, 10, 16, 23, 29, 31, 38-40, 42, 44, 51]. In [31], the authors proposed a self-adaptive learning-based test framework that learns the optimal policy of generating stress test cases for different types of software systems. The paper [39] presented a strategy to identify the tests that have passed or failed by combining clustering and semi-supervised learning. The paper [23] discussed that the Artificial Intelligence key pillars that can be used in software testing and talked about how the future will look like in terms of artificial intelligence and software testing. In [40], authors used a deep neural network to build a software defect prediction model and compared their proposed model with other machine learning algorithms like random forests, decision trees, and naive Bayesian networks. The authors claimed that their results showed small improvement over the other learning models in most cases. In [2], the authors proposed a methodology predict and localize faults in a software system. The authors used the random forest machine learning technique to train their model. The articles [38] and [44] talked about the use of machine learning in software quality assurance and prediction. In the article [16], authors studied 48 different papers focusing on making a survey of research efforts based on using ML algorithms to support software testing. The authors believed their mapping study would provide significant insights into machine learning applied to software testing. In [7], the authors discussed the current existing testing practices for ML programs. And the authors also explained the main sources of faults in an ML program. In [10], the authors focused on the relevant features of a large dataset in order to improve the accuracy of software quality estimation. The authors concluded that machine learning algorithms could help to estimate the quality level of software. In [51], the authors presented a comprehensive survey of machine learning testing research. The authors summarized the current research status of different ML testing properties, testing components, and testing workflow. In [42], the authors presented their work that can reduce the need for manual reviews by automatically identifying which code fragments should be reviewed manually. The authors concluded that their work could improve the speed of code reviews. In [29], the authors investigated the application for STEP of five machine learning (ML) models reported as the most accurate ones when applied to SDLC effort prediction.

4 Analysis Based on Various ML Techniques

This section analyzed various ML techniques used in software testing, which we study in this survey. Table 2 summarizes our findings.

Table 2: ML techniques used for software testing in this survey

Techniques	References	Number of Articles
Neural Network	[1], [3], [8], [11], [19], [25], [34], [40], [41], [42], [44], [45], [51]	13
Support Vector Machine	[8], [17], [19] [20], [21], [32], [41], [43]	8
Clustering	[1], [13], [39]	3
Decision Tree	[8], [10], [11], [28], [38], [40]	6
Grammar Induction	[20], [21]	2
Bayesian Based Method	[14], [24], [35], [38], [40]	5
Random Forest	[2], [10], [38], [40]	4
Generic ML Techniques	[5], [7], [9], [12], [15], [16], [18], [22], [23], [26], [27], [29], [30], [31], [32], [33], [36], [46], [47], [48], [49], [50]	22

4.1 Neural Network

Neural network is widely used in software testing. We found thirteen articles talking about this technique. Among the articles, neural network is used for fault-proneness prediction in [19] and is used for automatic test oracles generation in [25] and [1]. In [41], neural network is used to estimate the execution effort of software testing. In [34], neural network is used for metamorphic testing. In [40], neural network is used for software defect prediction. In [42], neural network is used for identifying code fragments for manual review.

4.2 Support Vector Machines

Support vector machines is widely used in software testing. We found eight articles talking about this technique. Among the articles, support vector machines is used for generating reliable test oracle in [32]. In [19] and [17], a support vector machine is used for fault-proneness prediction. In [41], the support vector machine is used to estimate the execution effort of software testing. In [20] and [21], the support vector machine is used for GUI testing. In [43], support vector machines is used for mutation testing.

4.3 Clustering, Decision Tree, and Grammar Induction

We found three articles talking about the clustering technique. In [13], clustering is used for improving regression test selection. Clustering is used for automatic test oracles generation in [1] and is used for classifying test outcomes in [39].

We found six articles talking about the decision tree. Decision tree is used for defect prediction in [8] and [11]. In [40], decision tree is used for software defect prediction. In [10] and [38], decision tree is used for software quality prediction.

We found two articles talking about grammar induction. In both [20] and [21], grammar induction is used for GUI testing.

4.4 Bayesian Based Method and Random Forest

We found five articles talking about Bayesian based method. In [14], Bayesian based method is used for combining testing techniques. In [40], Bayesian based method is used for software defect prediction. In [43], Bayesian based method is used for software quality prediction.

We found four articles talking about random forest. In [40], random forest is used for software defect prediction. In [2], random forest is used for software fault localization. In [10] and [38], random forest is used for software quality prediction.

5 Software Testing Activities Found in this Study

In our study we found different types of software testing activities that uses different types of machine learning techniques. We are discussing few of them in the following subsections.

5.1 Creating Test Data

In [49], the authors proposed a framework for value-based software test data generation. Value-based software engineering considers value into the software engineering principles and practices as opposed to value-neutral software engineering where each product in software development, such as requirement, use case, test case, and defect, is treated as equally important. The authors talked about applying machine learning methods to value-based software engineering in this paper.

5.2 Test Case Generation

In [31], the authors proposed a self-adaptive learning-based test framework that learns the optimal policy of generating stress test cases for different types of software systems. In test case generation, one of the important tasks is to reduce manual review of the code segments. In [42], the authors presented how machine learning can be used to reduce the need for manual reviews by automatically identifying which code fragments should be reviewed manually.

5.3 User Interface Testing

The success of a software product largely depends on an error free user interface. That is why user interface testing places a vital role in software quality. In [20] and [21], the authors talked about GUI software testing. They proposed avoiding infeasible test cases altogether by predicting which test cases are infeasible using two supervised machine learning methods: support vector machines (SVMs) and grammar induction.

5.4 Regression Testing

A regression testing is done whenever a change has been made to a software product to ensure that the change made did not introduce any new error. For a large product, running an effective regression test is challenging because of the amount of test cases needed to run. In [13], the authors introduced a semi-supervised clustering method named semi-supervised K-means (SSKM) for improving regression test selection. The authors claimed that the semi-supervised clustering method SSKM could improve test selection in most cases.

6 Conclusion

Software testing has been a great area of research as it is a vital issue for producing quality software [4]. Researchers are interested in performing successful testing with minimal effort possible by doing test automation. Machine learning can play an important role in this regard. It is evident from our study that the intersection of these fields has drawn attention from many researchers for a long time. From our study, we have seen many early techniques of machine learning, such as neural network, decision tree, etc., and modern techniques like deep learning are equally applicable in software testing.

Our citations in the paper are arranged in chronological order. A smaller reference number indicates an earlier publication, and a larger number indicates a recent publication. The summary of our study shown in Table 2 shows the popular techniques of ML as applied in software testing. For instance, neural network has been a popular technique all through our selected period. Also, we see that SVM and grammar induction are the techniques found in the middle period, whereas the recent trends are focused on the techniques like decision tree, Bayesian-based method, and random forest. Of course, a substantial number of papers applied machine learning in general without being specific to any particular techniques.

The papers in our study discussed many different approaches to applying ML in software testing and introduced the challenges in them. Researchers have proposed some potential ways to solve those challenges and suggested a number of future directions. The common theme we found in those works is that machine learning techniques can be employed during the software testing process. Using machine learning techniques can assist testers in predicting software defects, localizing software faults, finding some specific bugs, and improve effectiveness and efficiency. We hope the researchers and practitioners in this field will be benefited from our study.

We have also studied the techniques of the machine learning that can be used for different types of software testing activities such as creating test data, test case generation, testing user interfaces, and regression testing.

In the future, we will explore how machine learning is used for improving the testing of different categories of the software such as object-oriented software, distributed software, etc.

References

[1] D. Agarwal, D. E. Tamir, M. Last, and A. Kandel, "A Comparative Study of Artificial Neural Networks and Info-Fuzzy Networks as Automated Oracles in Software Testing," IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 42(5):1183-1193, 2012.

[2] H. Ali and T. A. Khan, "On Fault Localization Using Machine Learning Techniques," 2019 International Conference on Frontiers of Information Technology (FIT), pp. 357-3575, doi: 10.1109/FIT47737.2019.00073, 2019.

[3] C. Anderson, A. Von Mayrhauser, and R. Mraz, "On the Use of Neural Networks to Guide Software Testing Activities," *Proceedings of 1995 IEEE International Test Conference (ITC)*, IEEE, pp. 720-729, October 1995.

[4] A. Bandi and P. Heeler, "Usability Testing: A Software Engineering Perspective," 2013 International Conference on Human Computer Interactions (ICHCI), pp. 1-8, doi: 10.1109/ICHCI-IEEE.2013.6887809, 2013.

[5] N. Baskiotis, M. Sebag, M. C. Gaudel, and S. D. Gouraud, A Machine Learning Approach for Statistical Software Testing," *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI),* pp. 2274-2279, January 2007.

[6] J. F. Bowring, J. M. Rehg, and M. J. Harrold, "Active Learning for Automatic Classification of Software Behavior," *ACM SIGSOFT Software Engineering Notes*, 29(4):195-205, 2004.

[7] H. B. Braiek. and F. Khomh, "On Testing Machine Learning Programs," *Journal of Systems and Software*, 164:110542, 2020.

[8] L. C. Briand, "Novel Applications of Machine Learning in Software Testing," 2008 The Eighth International Conference on Quality Software, IEEE, pp. 3-10, August 2008,

[9] L. C. Briand, Y. Labiche, Z. Bawar, and N. T. Spido, "Using Machine Learning to Refine Category-Partition Test Specifications and Test Suites," Information and Software Technology, 51(11):1551-1564, 2009.

[10] A. A. Ceran and Ö. Ö. Tanriover, "An Experimental Study for Software Quality Prediction with Machine Learning Methods," 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), IEEE, 2020.

[11] E. Ceylan, F. O. Kutlubay and A. B. Bener, "Software Defect Identification Using Machine Learning Techniques," 32nd EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO'06), pp. 240-247, 2006, doi: 10.1109/EUROMICRO.2006.56.

[12] T. J. Cheatham, J. P. Yoo, and N. J. Wahl, "Software Testing: A Machine Learning Experiment," *Proceedings of the 1995 ACM 23rd Annual Conference on Computer Scienc*e, pp. 135-141, February 1995.

[13] S. Chen, Z. Chen, Z. Zhao, B. Xu, and Y. Feng, "Using Semi-Supervised Clustering to Improve Regression Test Selection Techniques," 2011 Fourth IEEE International Conference on Software Testing, Verification and Validation, IEEE, pp. 1-10, March 2011.

[14] D. Cotroneo, R. Pietrantuono, and S. Russo, "A Learning-Based Method for Combining Testing Techniques," 2013 35th International Conference on Software Engineering (ICSE) , IEEE, pp. 142-151, May 2013.

[15] J. Ding, X.-H. Hu, and V. Gudivada. "A Machine Learning Based Framework for Verification and Validation of Massive Scale Image Data," IEEE Transactions on Big Data, 2017.

[16] V. H. S. Durelli, R. Durelli, S. Borges, A. Endo, M. Elder, D. Dias, and M. Guimaraes, "Machine Learning Applied To Software Testing: A Systematic Mapping Study," IEEE Transactions on Reliability, 68.3:1189-1212, 2019.

[17] K. O. Elish and M. O. Elish, "Predicting Defect-Prone Software Modules Using Support Vector Machines," *Journal of Systems and Software*, 81(5):649-660, 2008.

[18] S. M. Ghaffarian, and H. R. Shahriari. "Software Vulnerability Analysis and Discovery Using Machine-Learning and Data-Mining Techniques: A Survey." ACM Computing Surveys (CSUR) 50.4:1-36, 2017.

[19] I. Gondra, "Applying Machine Learning to Software Fault-Proneness Prediction," *Journal of Systems and Software*, 81(2):186-195, 2008.

[20] R. Gove and J. Faytong, "Identifying Infeasible GUI Test Cases Using Support Vector Machines and Induced Grammars," 2011 IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops, IEEE, pp. 202-211, March 2011.

[21] R. Gove and J. Faytong, "Machine Learning and Event-Based Software Testing: Classifiers for Identifying Infeasible GUI Event Sequences," *Advances in Computers*, Elsevier, 86:109-135, 2012.

[22] M. Harman, "The Role of Artificial Intelligence in Software Engineering," 2012 First International Workshop on Realizing AI Synergies in Software Engineering (RAISE), IEEE pp. 1-6, June 2012.

[23] H. Hourani, A. Hammad, and M. Lafi, "The Impact of Artificial Intelligence on Software Testing," 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), pp. 565-570, doi: 10.1109/JEEIT.2019.8717439, 2019.

[24] S. Huang, E. H. Liu, Z. W. Hui, S. Q. Tang, and S. J. Zhang, "Challenges of Testing Machine Learning

Applications," *International Journal of Performability Engineering*, 14(6):1275, 2018.

[25] H. Jin, Y. Wang, N. W. Chen, Z. J. Gou, and S. Wang, "Artificial Neural Network for Automatic Test Oracles Generation," 2008 International Conference on Computer Science and Software Engineering, IEEE, 2:727-730, December 2008.

[26] U. Kanewala, J. M. Bieman, and A. Ben ‑ Hur, "Predicting Metamorphic Relations For Testing Scientific Software: A Machine Learning Approach Using Graph Kernels," Software Testing, Verification and Reliability, 26(3):245-269, 2016.

[27] L. Kumar and A. Sureka, "Aging Related Bug Prediction using Extreme Learning Machines," 2017 14th IEEE India Council International Conference (INDICON), pp. 1-6, 2017, doi: 10.1109/INDICON.2017.8487925, 2017.

[28] A. R. Lenz, A Pozo, and S. R. Vergilio, "Linking Software Testing Results with a Machine Learning Approach," Engineering Applications of Artificial Intelligence, 26(5-6):1631-1640, 2013.

[29] C. López-Martín, "Machine Learning Techniques for Software Testing Effort Prediction," *Software Quality Journal*, 30(1):65-100, 2022, https://doi.org/10.1007/s11219-020-09545-8, 2022.

[30] S. Masuda, K. Ono, T. Yasue, and N. Hosokawa,. A survey of software quality for machine learning applications. In 2018 IEEE International conference on software testing, verification and validation workshops (ICSTW), IEEE, pp. 279-284, April 2018.

[31] M. H. Moghadam, M. Saadatmand, M. Borg, M. Bohlin, and B. Lisper, "Machine Learning to Guide Performance Testing: An Autonomous Test Framework," 2019 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), IEEE, pp. 164-167, April 201.

[32] C. Murphy, G. E. Kaiser, and M. Arias, "An Approach to Software Testing of Machine Learning Applications," Proceedings of the 19th International Conference on Software Engineering & Knowledge Engineering, Technical Program, Hyatt Harborside Hotel, Boston, Massachusetts, USA, July 9-11, January 2007.

[33] S. Nakajima, "Quality Assurance of Machine Learning Software," 2018 IEEE 7th Global Conference on Consumer Electronics (GCCE), IEEE, pp. 601-604, October 2018.

[34] S. Nakajima, "Dataset Diversity for Metamorphic Testing of Machine Learning Software," International Workshop on Structured Object-Oriented Formal Language and Method, Springer, Cham, pp. 21-38, November 2018.

[35] A. S. Namin and M. Sridharan, "Bayesian Reasoning for Software Testing," *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research*, pp. 349-354, November 2010.

[36] M. Noorian, E. Bagheri, and W. Du, "Machine Learning-based Software Testing: Towards a Classification Framework," In SEKE, pp. 225-229, July, 2011.

[37] E. Rashid, P. Srikanta, and V. Bhattacherjee. "A Survey in the Area of Machine Learning and Its Application for Software Quality Prediction," ACM SIGSOFT Software Engineering Notes, 37.5:1-7, 2012.

[38] S. Reddivari and J. Raman. "Software Quality Prediction: an Investigation Based on Machine Learning," 2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI), IEEE, 2019.

[39] M. Roper, Using Machine Learning to Classify Test Outcomes," 2019 IEEE International Conference on Artificial Intelligence Testing (AITest), IEEE, pp. 99-100, April 2019.

[40] M. Samir, M. El-Ramly and A. Kamel, "Investigating the Use of Deep Neural Networks for Software Defect Prediction," 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), pp. 1-6, doi: 10.1109/AICCSA47632.2019.9035240, 2019.

[41] D. G. e Silva, M. Jino, and B. T. de Abreu, "Machine Learning Methods and Asymmetric Cost Function to Estimate Execution Effort of Software Testing," 2010 Third International Conference on Software Testing, Verification and Validation, IEEE, pp. 275-284, April 2010.

[42] M. Staron and O. Soder, "Using Machine Learning to Identify Code Fragments for Manual Review," 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), IEEE, 2020.

[43] J. Strug and B. Strug, "Machine Learning Approach in Mutation Testing," IFIP International Conference on Testing Software and Systems, Springer, Berlin, Heidelberg, pp. 200-214, November 2012.

[44] L. Surya, "Machine Learning-Future of Quality Assurance," *International Journal of Emerging Technologies and Innovative Researc*h (www. jetir. org), ISSN:2349-5162, 2019.

[45] M. Vanmali, M. Last, and A. Kandel, "Using a Neural Network in the Software Testing Process," *International Journal of Intelligent Systems*, 17(1):45-62, 2002.

[46] S. R. Vergilio, J. C. Maldonado, and M. Jino," Infeasible Paths in the Context of Data Flow Based Testing Criteria: Identification, Classification, and Prediction," *Journal of the Brazilian Computer Society*, 12(1):73-88, 2006.

[47] T. Xie, "The Synergy of Human and Artificial Intelligence in Software Engineering," 2013 2nd International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE), IEEE, pp. 4-6, May 2013.

[48] X. Xie, J. W. Ho, C. Murphy, G. Kaiser, B. Xu, and T. Y. Chen, "Testing and Validating Machine Learning Classifiers by Metamorphic Testing," *Journal of Systems and Software*, 84(4):544-558, 2011.

[49] D. Zhang, "Machine Learning in Value-Based Software Test Data Generation," 2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06), IEEE, pp. 732-736, November 2006..

[50] D. Zhang and J. J.Tsai, "Machine Learning and Software Engineering," *Software Quality Journal*, 11(2):87-119, 2003.

[51] J. M. Zhang, M Harman, L. Ma, and Y. Liu, , "Machine

Learning Testing: Survey, Landscapes and Horizons," IEEE Transactions on Software Engineering, 2020.

Mohammad Hossain is an Assistant Professor of Software Engineering and IT Management at the University of Minnesota Crookston where he teaches various software engineering courses. He earned his Ph.D. from North Dakota State University in 2016. His Ph.D. dissertation title was "Foundational Algorithms Underlying Horizontal Processing of Vertically Structured Big Data Using pTrees". His research interest includes Data Mining, Machine Learning, Software Engineering, Cybersecurity, Algorithm etc. He is a member of ISCA and served as the Program Chair of CATA 2021 and CATA 2022.

Hongkai Chen received his BS in Software Engineering from University of Minnesota Crookston in 2021. He was a honor student at UMC and received UROP scholarship from University of Minnesota. He is currently pursuing his MS in Computer Science at University of California, San Diego. His research interest includes Software Engineering, Machine Learning, Cybersecurity, etc.

Semantic Reasoning to Support End User Development in Intelligent Environment

Narayan C. Debnath*, Shreya Banerjee*, Giau Ung Van*, Phat Tat Quang*, and Dai Nguyen Thanh*
Eastern International University, Binh Duong, VIETNAM-

Abstract

Intelligent Environment (IE) creates complex applications on top of an existing network of sensors and actuators. Proliferation of Internet of Thing (IoT) objects in Intelligent Environments exhibits the rapid growth of End User Development (EUD). Trigger Action programming is a popular approach for EUD in IE. However, the inability of end users to interpret and compose suitable Trigger Action rules often makes inconsistencies in behavior of IoT objects. To address this issue, a semantic based reasoning framework is proposed in this paper to support end users. The proposed framework is based on an upper-level ontology specification named as Trigger Action Ontology (TAO). This framework includes a rule-based reasoner implemented in Apache Jena. The framework will assist end users of IE applications on various domains to represent triggers, actions and their respective combinations. In addition, the proposed reasoner can aid end users in recognizing programming bugs and reason about how to fix them. Further, two case studies in two different domains home automation and smart factory have been specified to prove the efficiency of the proposed framework. Moreover, a detailed comparison study has been provided to demonstrate the usefulness of the proposed work over the existing approaches.

Key Words: Intelligent environment, semantic reasoning, end user development, inference rules, reasoner, trigger action programming, ontology, programming bugs.

1 Introduction

End-User Development (EUD) empowers non-professional developers to build or modify their own applications to address their various and frequently changing requirements. One of the approaches considered in this area is the use of rule-based systems [16]. Trigger Action Programming (TAP) is such kind of rule-based system in EUD. TAP is a programming model enabling users to connect services and devices by writing if-then rules in the form: *if condition then action* [2, 20]. These kinds of rules are simple to implement. However, nuances in their interpretation can lead to user errors with consequences such as incorrect and undesirable functionalities [20].

The term Intelligent Environments (IE) refers to a diverse range of scenarios and applications that include smart homes, smart factories, smart farming, autonomous vehicles, and so on. Several commercial approaches facilitate end user to specify the creation of trigger action rules in IE such as Smart Home. Some popular names are IFTTT [13], Microsoft Flow [14], Zapier [22], Mozilla's Things Gateway [12], Stringify [19], etc. Intelligent environments are full of diverse and complicated device automation scenarios that constantly arise with multiple devices. Unfortunately, in these kinds of environments, end users may write rules with bugs or struggle to understand why particular automations are running [24]. In general, TAP connects a single trigger towards a single action. However, these rules can become complicated based on underlying behaviors, which require precise and rigorous expressiveness [2]. To cope with these complexities, several commercial platforms such as Stringify [19] and SmartRules [21] support conjunctions in a single trigger. Nevertheless, behaviors can become more complicated and will demand creation of more complex rules.

In this context, existing TAP based approaches face several challenges. Crucial challenges among these are as follows. Firstly, these approaches have nil or very little support towards empowering end users in order to realize different types of triggers, actions and their in between various connections. Existing literature [2, 8, 11] represents that triggers can be of different kinds such as event and state. Likewise, actions can be also of different kinds such as immediate action, extended action and sustained actions. Event kind of triggers happen in a specific moment. "When I enter the room" is an example of an event type trigger. State kind of triggers persist for a long period. "As long as it's raining" is an example of a state type trigger. On the other hand, immediate actions can happen at a moment. "Sending an e-mail" is an example of an immediate action. Extended actions can persist for some time and then end. "Brewing the coffee" is an example of extended action. Sustained actions can persist until other behavior is defined on the same object [21]. "Turn on the light" is an example of sustained action. Identification of these different kinds of triggers and actions are essential to form a correct and consistent TAP rule. Besides these, to achieve complex behaviors in intelligent environment, various triggers can be connected with each other. Similarly, different actions also can be connected with each other. These intra trigger or intra action connections can be different kinds such as "and" connection, "or" connection. However, non-expert end users have no or very little interpreting capability of these kinds of semantics. Hence, a semantically empowered framework is needed that can assist end users to realize the correct semantics of TAP.

Secondly, existing approaches lack facilities that may help end users to discover their mistakes and wrong interpretation when they are composing the rules. A formal semantic based tool can help end users to identify bugs in their rules and help them to rectify their mistakes [4]. However, most of the

* Department of Software Engineering. Email: narayan.debnath@eiu.edu.vn, shreya.banerjee@eiu.edu.vn, giau.ung@eiu.edu.vn, phat.tat@eiu.edu.vn, dai.nguyen@eiu.edu.vn.

commercial approaches lack these features. Existing studies [2, 11, 21] have shown that several bugs can be present in TAP rules due to incorrect interpretation of users. Several crucial examples of those bugs are infinite loop, contradict action, and timing window fallacy. Infinite loop bugs occur when rules trigger one another, resulting in loops [2]. Contradict action bugs can happen if two rules include the same trigger and create two contradictory actions on the same device. Timing window fallacy bugs arise when the time window is mismatched if multiple triggers or multiple actions are combined. For example, two event kinds of triggers will generate an immediate action. One event and one state kind trigger will generate sustained or extended actions. In this case, event trigger should happen within the time window of the state kind trigger. In addition, if one event produces a sustained or extended action, then it is also included in the timing window fallacy bug since, an event can only create an immediate action [2]. Similarly, a state kind of trigger can only create sustained or extended action. Besides presence of these crucial bugs, synthesis of TAP rules also includes several issues such as redundant rules and unused rules. End users create rules, which have the same triggers but different actions happening at the same time. Similarly, end users can also create rules, which have the same actions but different triggers than those occurring at the same time. These kinds of redundant rules are created by the end users very frequently. Besides these, several TAP rules are created which are never getting a chance to be executed. These kinds of rules are unused rules [11]. Further, end users often forget to deactivate the sustained action. This kind of bug is known as lack of action reversal [20]. Sometimes end users have not specified how long the extended action can be executed. This kind of inconsistent behavior of TAP rules is known as extended action related bugs [2]. All of these above-mentioned issues are very crucial for creation of consistent and correct TAP rules [21]. Hence, a tool is required that can detect these bugs and other issues efficiently. Moreover, the specified issues are related with precise semantics of different building blocks of TAP rules. Therefore, a semantic based reasoning tool will be in great demand that may assist an end user to discover bugs and reasons to repair them.

This paper is aimed to address these aforementioned challenges. The proposed work deals with the following research questions related with these specified challenges. *Q1*. How a semantically empowered framework can assist end users in order to create TAP rules? *Q2*. How a semantic based reasoning tool can be developed that can empower end users to identify crucial bugs and rectify those? With the objective to answer these research questions, this paper has proposed a semantic based reasoning framework. The proposed framework is based on an ontology-based specification named as Trigger Action Ontology (TAO) described in [6]. This framework can assist end users to create TAP rules according to precise semantics. These precise semantics are provided by the upper ontology TAO [6]. Ontology is defined as an explicit specification of shared conceptualization. It specifies an abstract view of the world in terms of concepts and their in between relationships [9]. The literature recognizes the value of semantic enrichments, through ontologies, for facilitating the event driven programming of IoT devices also in other domains [5]. In addition, the proposed framework includes a rule-based reasoner that can assist end users to identify important bugs and reason about how to fix them.

The contribution of the proposed work are manifolds. Firstly, it assists end users to synthesize TAP rules based on precise semantics, since the framework is based on an upper-level based ontology specification. To represent precise semantics related to a single TAP rule, end users are asked a set of questions related with the 5W1H (Why, Who, When, What, Where and How) contextual information. Based on the answers provided by end users and with the help of the formal semantics of underlying ontology-based specification TAO, the proposed framework assists the end users to create efficient TAP rules. Secondly, the proposed framework provides a rule based generic reasoner that can help end users to identify crucial bugs. The proposed reasoner is based on a set of inference rules. These inference rules are proposed based on the formal semantics provided by axioms in TAO. Using this proposed reasoner, end users can identify indefinite loop, contradict actions, time-window fallacy, lack of action reversal, extended action related bugs, redundant rules and unused rules in the synthesized TAP rules. Thirdly, the reasoner also helps end users to reason how to fix them. The proposed semantic based reasoning framework is implemented using Java based Ontology API Apache Jena [1]. Fourthly, the proposed framework can be applied in various IE domains such as smart healthcare, smart factory, smart home etc. The framework is based on an upper-level ontology specification TAO which is domain independent. In addition, effective case studies in two different domains, smart home and smart factory are used to prove the effectiveness of the proposed framework. A detailed comparison study is provided between the proposed approach and the existing approaches to illustrate the improved performance of the proposed work.

The rest of the paper is organized in the following way. Related work is represented in Section 2. A brief description of TAO [6] is represented in Section 3. Proposed methodology is specified in Section 4. Further, the effectiveness of the framework is evaluated using suitable case studies in Section 5. A detailed comparison study is provided in Section 6. Finally, in Section 7, the paper is concluded with indication of crucial future works.

2 Related Work

Several state-of-the art approaches exist in the literatures that have created framework and bug identification and fixing tools to assist end users in IE. Very few approaches have applied ontology. The majority of those approaches have used other methodologies rather than ontology. Brief descriptions of these approaches are specified next.

In [4], authors have created both user interface and a tool that can identify bugs in TAP rules. Authors have developed a debugging tool that generates the possible problems by the rules synthesized by end users. The described debugging tool also displays systematic simulation. Authors have applied a hybrid approach named as Semantic Colored Petri Net Model (SCPN) in devising the tool. This hybrid approach is based on colored petri net models and an ontology specification. However, in this approach the authors mainly focus on three kinds of bugs indefinite loop, inconsistent rule and redundancy. Authors have not focused on bugs such as lack of action reversal or extended action related bugs. The approach is applicable in a smart home domain. In [23], authors let end users represent desired properties for devices and services. Authors have transformed these properties into linear temporal logic (LTL) and then create property satisfying TAP rules from scratch and repairs existing TAP rules. However, proposed approach in [23] cannot consider the fact, that an end user can make mistakes in specifying the properties. The approach does not consider the identification of different kinds of triggers, actions, intra-trigger, intra-action combinations and related bugs. In [24], the authors automatically synthesize TAP rules from traces. Traces are time-stamped logs of sensor readings and manual actuations of devices. This approach applies to both symbolic reasoning and SAT to synthesize consistent TAP rules, although their application area is mainly smart homes. They have also not considered about bugs and wrong interpretation due to different kinds of triggers and actions. In [21], authors have verified Event-Condition-Action (ECA) in IE rules using symbolic verification. They have mainly focused on three criteria: unused rules, redundant rules and incorrect rules. However, they have also not considered different kinds of action related bugs such as extended action related bugs and sustained action related bugs. In [15], authors have described the use of visual analytics to support analysis of the interactions carried out by users with trigger action rule-based personalization tools. Authors have also presented the application of the described method to data generated by the use of the PersRobIoTE tool. However, the approach does not include any tools that can identify bugs. In [18], the authors have presented a technique that mainly identifies errors due to missing triggers and the consequent unexpected behavior and security vulnerabilities specifically in a smart home. The authors have focused on event kinds of triggers. They also have developed a tool based on the described methodology and they considered that actions are defined by end users correctly. Further, they did not mention other kinds of bugs such as indefinite loop and contradict actions. In [3, 17], authors have developed visual tools analyzing the users' behavior when interacting with a trigger action rule editor for personalizing their IoT context dependent applications. In [17], authors also recommended how to combine multiple triggers or actions. However, in both approaches, authors have not developed any tool to find out bugs. In [25], authors have introduced interfaces that help users compare and contrast TAP program variants. The described interfaces help users reason about syntax differences, differences in actions under identical scenarios and property differences. However, they have not considered differences among triggers. In [7], authors have developed a composition paradigm of events and actions in the domain of IoT. They also have considered about 5W1H. However, they did not consider about a bug identifying tool. In [16], authors have developed a tool that mainly focuses on semantic correctness of TAP rules. They have provided end users the information related to "why/why not". Yet, they have not considered various bugs such as lack of action reversal or indefinite loops.

The majority of the existing approaches have not considered bugs that arose due to a wrong interpretation of different kinds of action's semantics. Further, very few considered a generic tool that can be applied over various kinds of domain. The proposed framework and the reasoning tool in this paper have identified different kinds of bugs related with both triggers and actions. The proposed framework can also be applied in various domains, since it is supported by an upper-level ontology specification TAO. A detailed comparison study between the proposed work and selected existing approaches have been provided in section 6.

3 Brief Description of Trigger Action Ontology (TAO) [6]

Trigger Action Ontology (TAO) described in [6] is an upper-level ontology to represent meta-rules for TAP. TAO consists of three layers - *Rules*, *Context* and *IoT Resources*. The bottom most layer of proposed TAO is *IoT Resources*. It provides ontology-based descriptions for IoT devices, services and related attributes. *Context* is the middle layer that represents the contextual information (5W1H) related to triggers and actions. This 5W term presents the basic information related to trigger and actions as follows. "Who" represents who is responsible for triggering or performing an action? "Who" can be an IoT device, a service or an end user. "When" represents, the temporal aspects that when trigger or action can happen. "Where" provides the location information related to the trigger and action. "What" represents what the trigger and the action specified. "Why" describes the reason of the trigger and performing the action. This contextual information is classified as *primary context* and *auxiliary context*. The top most layer is *Rules* that provides the precise semantics towards different kinds of triggers, actions, multiple triggers and multiple actions. Formal semantics of TAO is expressed in first order mathematical logic. Figure 1 has demonstrated the TAO model. Table 1 has described the facades of TAO.

4 Proposed Methodology

A semantic based reasoning framework is proposed in this section in order to support end-users in IE to synthesize trigger action rules. In addition, using the proposed framework, end users can identify bugs in their TAP programming rules and reasons to fix them. Section 4.1 has represented the modules and workflow of the proposed semantic based reasoning framework. Further, Section 4.2 has represented the proposed inference rules and the generic reasoner to identify the bugs in synthesized TAP rules. Section 4.3 represents the implementation of the proposed framework.

4.1 Proposed Semantic Based Reasoning Framework

The proposed semantic model is based on an upper-level ontology specification TAO described in Section 3. The proposed framework includes interfaces for both end users and service providers. In addition, it includes a rule based generic reasoner. Figure 2 has illustrated the outline of the proposed framework. The framework consists of two modules. The first module is the *User Interface module*. Using this user interface module, both end users and service providers can connect with the framework. Second is *Reasoning Module*. This module consists of *a Generic rule-based reasoner*, *Questionnaire Module* and the *Web*

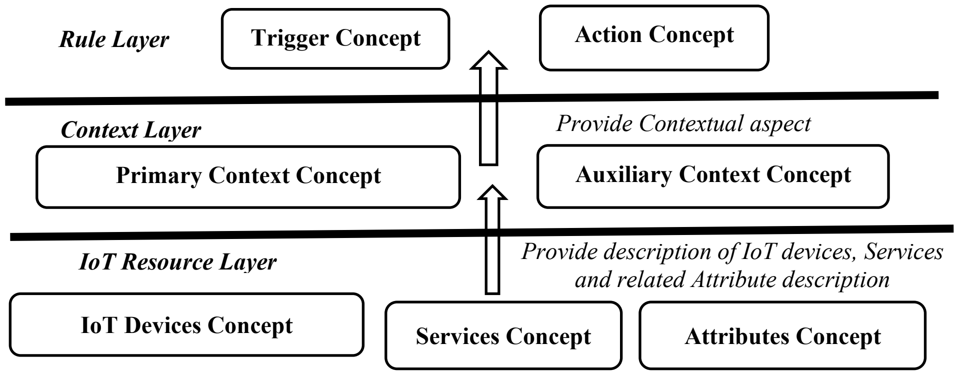


Figure 1: Trigger action ontology (TA) model specified in [6]

Table 1: Brief description of different constructs of TAO

Principle Constructs in TAO [6]		Description
IoT resource layer	IoT devices	Distinct domains based on IoT consists of different kinds of devices, such as sensors, actuators, and tag devices.
	Service	IoT devices can provide and consume several services
	Attributes	This concept represents properties of different IoT devices and services
	Different Kinds of relationships	Provide, Consume and Has Attribute. Provide and Consume relationship exist between device and service. Has Attribute relationship exist between either device and attribute, or between service and attribute
Context layer	Primary Context	Primary context represents 5Ws information – What, Why, Where, When and Who
	Auxiliary Context	Auxiliary context represents additional information related to primary context
Rule layer	Trigger	This concept represents causes for activation of actions. Triggers can be further categorized as two types – event and state
	Action	Actions are activated due to triggers. Actions can be further classified as two types – immediate, extended, and sustained
	Multiple Triggers and Multiple Actions	Triggers can be connected to each other using different kinds of connections such as “And”, “Or”. Likewise, actions are connected with each other using different kinds of connections
	Triggering Relationship	This relationship exists between triggers and actions.

Ontology Language (OWL) specification of TAO. At first (1), service providers are asked to enter the details about devices, services and attributes related with the domain. Then, end users are asked 5W1H questions to get the answers related with a TAP rule. End users are asked to choose trigger, action, and triggering relationships. They are also asked about the devices, location and time related to triggers and actions. Next in (2), based on these answers provided by end users, a specific TAP rule is generated. In the proposed framework, a rule file is created from TAP, which consists of inference rules specified in Apache Jena rule Language [1]. Following in (3), based on the inference rules, the generic reasoner evaluates TAP rules synthesized by end users in order to find out the bugs and inconsistencies within the rule. If any bug or inconsistencies are discovered, then end users will be notified through the user interface module.

Figure 3 has specified the workflow of the framework. According to Figure 3, the first step (1), service providers are asked to enter the details of triggers and actions. For example, if the service providers belong to a Smart Home domain, they will submit details related to triggers and actions included to that domain. Likewise, if the service providers belong to Smart Factory domain, they will submit detailed triggers and actions related to the smart factory domain. In the second step (2), the Web Ontology Language (OWL) specification of TAO is populated based on the information entered by service

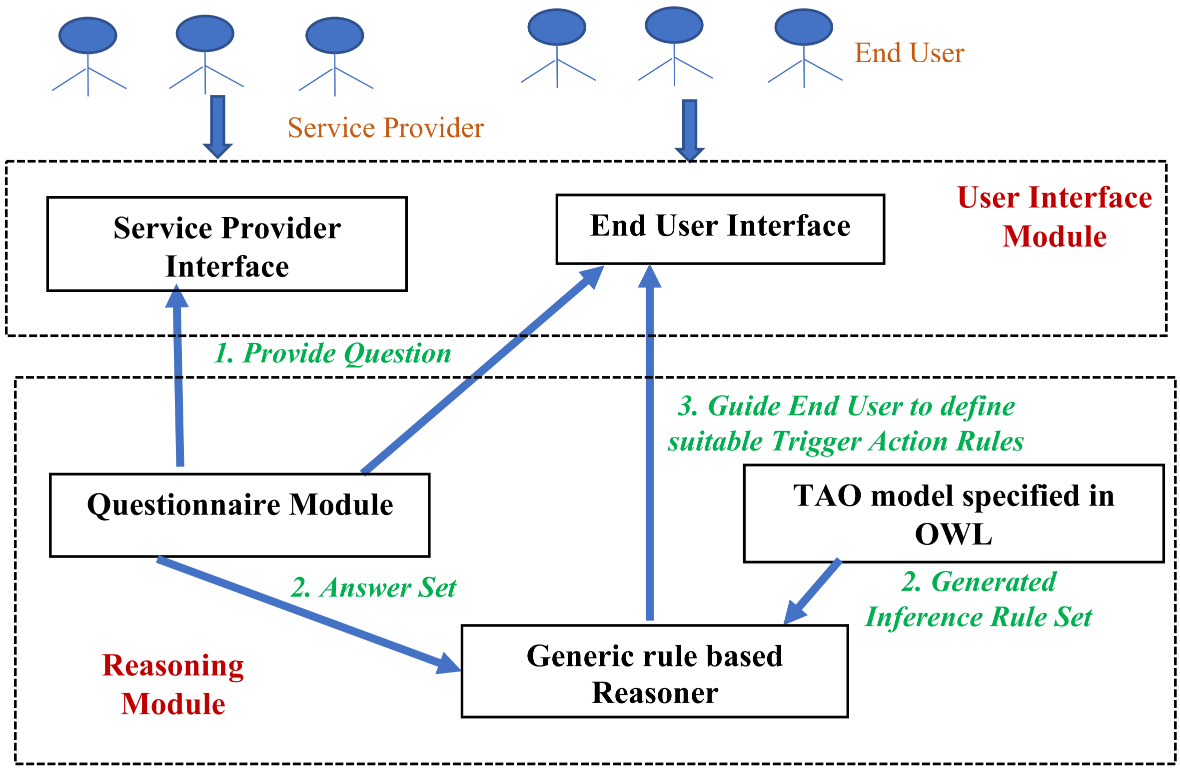


Figure 2: Proposed semantic reasoning framework

providers. In step (3), end users are asked to enter a behavior description. For example, in the case of smart factory domain, end users such as a manufacturing company can enter a behavior such as, "We want to be notified when the RPM count of our machine is unusual". Similarly, in the case of smart home domain, end users can enter a behavior such as, "When it is raining, the doors and windows need to be closed." In step (4), end users are asked to choose triggers and actions related with the entered behaviour from the list of triggers and actions entered by the service providers. They are also asked to enter detailed information related to triggers and actions based on a set of questions. These questions are based on 5W1H contextual information. The entered information from end users is used to populate the OWL specification of TAO. In step (5), the rule is synthesized by the framework based on the answers achieved in the previous step. Next, the rule is evaluated by the reasoner based on the proposed inference rules in Section 4.2 to check if the rule contains any bugs or inconsistencies. In step (6), the bugs and inconsistent rules are displayed toward end users along with the reason why the rule creates a problem. This helps end users find the solution to fix those bugs. Table 2 has specified an example set of questions related to 5W1H context asked to the end users. This set of questions specified in Table 2 is related with actions. Likewise, similar sets of questions are asked to end users to acquire the 5W1H context related information about triggers. These sets of questions are derived from the context layer of TAO.

4.2 Proposed Inference Rules and Reasoner

This section has represented a set of inference rules based on whether the proposed generic reasoner will check if there are any bugs present in the synthesized TAP rule or not. The proposed inference rules are created based on the formal axioms of TAO. These inference rules are based on IE and domain independent. Hence, the proposed reasoner can check the consistency or presence of bugs in TAP rules in various kinds of domains such as smart home, smart factory, health automation system etc. In this paper, the proposed inference rules are related with identification of several crucial bugs and consistency issues. Table 3 has represented those bugs with brief descriptions and examples. Further, Figure 4 has illustrated an example of an inference rule that can identify a timing window fallacy bug. This example represents, that there is a TAP rule *r1*. This rule has trigger *t1* and *t2*. The rule initiates an extended action *a1*. Trigger *t1* is an event and trigger *t2* is a state. Trigger *t1* happens at moment *Ti1*. Trigger *t2* is started at *TiS* time stamp and ended at *TiE* time stamp. Now, if value of *Ti1* is less than *TiS* or greater than *TiE*, this means the event *t1* is not occurring within the period of the state *t2*. However, *t1* needs to happen within the period of *t2* since, both have initiated an action *a1*. This rule is defined to identify the bug as specified in Table 3 time window fallacy bug description (iv).

4.3 Implementation of the Proposed Framework

In this section the proposed semantic based reasoning framework is implemented using Java programming language and Apache Jena [16]. Apache Jena is a java-based API used for building semantic web and linked data applications. This API supports different reasoners. Rule based generic reasoner is one of those. Using this kind of reasoner, an ontology specification can be reasoned based on inference rules in order to get additional information. In this paper, the inference rules proposed in Section 4.2 are used to create a rule based generic reasoner. The proposed inference rules are implemented using Jena Rule Language [16]. The implemented generic reasoner in Apache Jena will take OWL specification of the populated TAO obtain after step 4 in Figure 3. Then reasoner checks for bugs and inconsistencies within the TAP rules using proposed inference rules. If bugs or inconsistencies are found then the reasoner will display the issues toward end users along with the solutions to fix them. The proposed framework has two text-based user interfaces. One interface is devised to interact with service providers. Another interface is devised to interact with end users using a 5W1H questionnaire. Using Java programming language and Apache Jena, both interfaces are implanted and information users acquired through these are populated into OWL specification of TAO. Figure 5 has specified the partial view

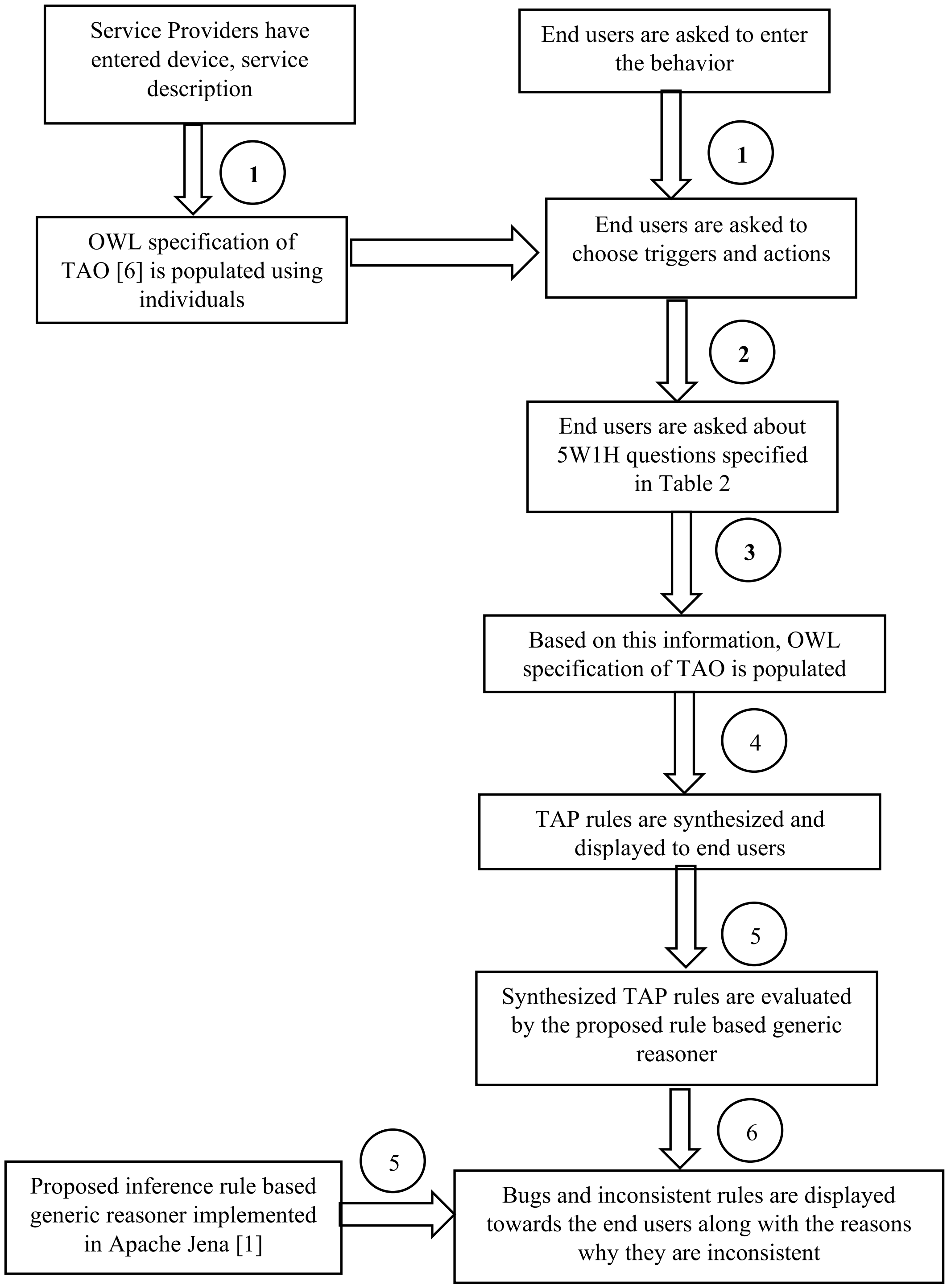


Figure 3: Workflow of proposed semantic reasoning framework

Table 2: Example of 5WlH context related question asked to end

Questions	Descriptions
Q1: What do you want the action to do?	What aspect of action
Q2: Do you want the action to do the job instantly or during some time? Specify time information.	When aspect of action. Thus, end users can choose between immediate action and other types of action
Q3: Do you want the action to be terminate itself or upon activation of another trigger?	When aspect of action. Thus, end users can choose between extended action and sustained action
Q4: Who is responsible to do the action?	Who aspect of action
Q5: Do you want to make "AND" combinations with other actions?	Add combination of triggers
Q6: Do you want to make "OR" combinations with other actions?	Or combination of triggers
Q7: Do you want to prevent other actions to this job, when this action is going on?	No combinations of trigger
Q8: Where the action will happen?	Where aspect of the action

Table 3: Descriptions of crucial bugs and related examples

Bug Name	Bug Description	Example
Indefinite Loop	If several rules trigger each other, then those rules have created indefinite loop bug.	Rule 1: If AC in the room is switched on, then windows in the room need to be closed. Rule 2: If windows in the room are closed, then the AC in the room will be switched on
Contradict Action	If one similar trigger can activate contradict actions in two different rules, then those rules can create contradict action bug.	Rule 1: Within 7 a.m. to 9 p.m. if room temperature is beyond 30° centigrade, then temperature of AC needs to be decreased. Rule 2: Within 7 a.m. to 9 p.m. if it is raining, then temperature of AC needs to be increased.
Redundant Rules	If two triggers activate the same action in different rules or one trigger activates two different actions in different rules, then those rules become redundant rules.	Rule 1: If it is raining, then the windows need to be closed. Rule 2: If AC is switched on, then the windows need to be closed.
Time-Window fallacy	(i) Semantics of event and state trigger can be changed based on the wrong time-window interpretation.	Rule 1: If music system is on and someone is present in the room, then lights in the room need to be dim.
	(ii) Semantics of immediate, sustained and extended action can be changed based on the wrong time interpretation.	In this rule, music system needs to be on within the time window when someone is present in the room. This rule is representing (iv) bug.

	(iii) When two states are combined, if their time windows are not matched, then this kind of bug is generated.	
	(iv) When one state and one event are combined, if the event does not happen within the time window of state, then this kind of bug is generated.	
	(v) When different kinds of actions are combined, then immediate action does not happen within the time frame of extended or sustained action, then this kind of bug is generated.	
	(vi) One event and one state should generate either extended or sustained action. Two states can generate either immediate or extended, sustained actions.	
Lack of Action Reversal	If sustained action is not deactivated by another rule, then it can be a continued. This situation creates lack of action reversal bug.	Rule 1: If requested item part is found inside the warehouse, then it should be delivered by drone toward the assembly line In this rule, the delivery action should be deactivated after it is finished. Otherwise, the drone will be in the assembly line.
Extended Action Bug	If end users forget to define the finishing time or finishing condition of extended actions, then the generated bug is known as Extended Action Bug.	When it is 7 pm, the coffee starts to brew. In this rule, there is no finishing time or condition about how long coffee will be brewed.
Unused Rules	Several times end users have defined some rule, which yet cannot be used in real time. This kind of bug is known as Unused Rules.	If it is raining heavy outside, then close the windows of the garage room. In this rule, if garage room has no window, then this rule will never be executed.

of the proposed framework implemented in Java. Figure 6 has represented a partial view of OWL specification of TAO in Protégé Tool [10].

5 Illustration of the Proposed Framework using Case Studies

In this section, proposed framework is evaluated based on two case studies from two different domains based on IE. Section 5.1 has demonstrated the applicability of the proposed framework using a case study from smart home domain. Section 5.2 has illustrated the effectiveness of the proposed framework using a case study from smart factory domain.

5.1 Illustration of Case Study 1

The case study specified in this section is based on smart home domain. Let's assume, an end user in smart home domain wants to implement the following behavior: Jenny wants to open the windows of the living room from morning to night. During this time, if it is raining, then windows will be closed. When the rain is over, then windows will be opened again. If the temperature of the living room exceeds 28° centigrade, then temperature in AC situated in that room will be decreased. If it is raining, then the temperature in AC situated in the living room will be increased. When the coffee maker starts to brew then the music system will be

```
IF r1 is an instance of concept Rule
t1 is an instance of concept Event
a1 is an instance of concept Extended Action
t2 is an instance of concept State
Ti1 is a value of date time stamp
TiS is a value of date time stamp
TiE is a value of date time stamp
AiS is a value of date time stamp
AiE is a value of date time stamp
has_Trigger is a relationship
has_Action is a relationship
has_Start_Time is a relationship
has_End_Time is a relationship
has_Time_Stamp is a relationship
And_Connection is a relationship
r1 has_Trigger t1
r1 has_Action  a1
r1 has_Trigger t2
t1 has_Time_Stamp Ti1
t2 has_Start_Time TiS
t2 has_End_Time  TiE
a1 has_Start_Time AiS
a1 has_End_Time  AiE
t1 And_Connection t2
greaterThan(TiS, Ti1)
lessThan(TiE,Ti1)
---------------------------------------------------------------
Then, print (Rule r1 has a bug. Trigger t1 need to be occurred within
the time span if Trigger t2)
```

Figure 4: Example of a proposed inference rule for identifying time window fallacy bug

turned on in the living room. When someone turns on the music system in the living room, then the coffee is starts to brew.

Let assume based on this behaviour, the end user has created the following TAP rules using the proposed framework.

(1) If it is 7 a.m., the window in the living room will be opened.
(2) If it is 9 p.m., the window in the living room will be closed.
(3) If the time is between 7 a.m. and 9 p.m. and it is raining, the window in the living room will be closed.
(4) If the temperature goes beyond 28°centigrade in the living room, the AC temperature situated in the living room will be decreased.
(5) If coffee starts to brew, then the music system will be turned on in the living room.
(6) If music system is turned on, then coffee starts to brew.
(7) If it is raining, then the AC temperature in the living room will be increased.

Among these above-mentioned rules, rule number 5 and rule number 6 have created an indefinite loop, because both rules have triggered each other. Rule number 3 has produced lack of action reversal bug. In this case, the action is a sustained type of action. End users have not defined any rule to deactivate the rule. Rule number 4 and rule number 7 have created contradictory actions. There may be situations when raining and the temperature of the room exceeds 28° centigrade can occur at the same time. Rule number 1 and 2 will produce time window fallacy bugs. In both rules triggers are event kinds, but actions are not immediate types of actions. These rules also create bugs such as how long the extended action should be continued. Figure 7 has demonstrated that the proposed framework has helped the end users to create the above-mentioned rules. Figure 8 has illustrated, that illustrated, that the proposed framework has assist to find the bugs in the created TAP rules.

5.2 Illustration of Case Study 2

The case study specified in this section is based on smart factory domain. Let's assume an end user in a smart factory domain wants to implement the following behavior.

ABC is a manufacturing company. It wants to monitor the work performance of a water purification system if the water pressure is greater than 50 and water flow is greater than 15. It also wants to check the stock of parts of the water purification system. If the stock of parts is nil, then the supplier employee is notified through Short Message Service (SMS). Further, when the warehouse has received

```
 public class Main {
      public static void main(String[] args) {
            BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));
            String choice=null, choiceRule=null,
choiceBehavioure=null;
            DomainExpert expert=new DomainExpert();
            //EndUser user=new EndUser();
            Interface inter=new Interface();
            System.out.println("What is your role? Please enter
either 'Service Provider' or 'End User'.");
--------------------------------------------------------------
                  //expert.readOntology();
                  expert.createDomainTrigger();
                  expert.createDomainAction();
                  expert.createContradictRelation();
                  expert.writeOntology();
            } catch (Exception e) {

                  e.printStackTrace();
            }
            System.out.println("..You as a Service Provider done
your Job....");
            }
            if(choice.equalsIgnoreCase("End User"))
            {

------------------------------------------------------------------
                  System.out.println("..You as an End User done
your Job....");
                  System.out.println("Debugging is started");
                  inter.reasoningRule();
                  System.out.println("Debugging is stopped");

            }
      }
}
```

Figure 5: Partial view of proposed semantic reasoning framework implemented in Java and Apache Jena [1]

a request on a particular lightweight material and the material is found in the warehouse, then it will be delivered to the assembly line.

Let's assume based on this behavior, the end user has created the following TAP rules using the proposed framework:

(1) If water pressure is greater than 50, then working performance of the water purification needs to be monitored.
(2) If water pressure is greater than 25, then working performance of the water purification needs to be monitored.
(3) If the stock of parts is nil, then the supplier employee is notified through email.
(4) If the warehouse received a request on a particular lightweight material, the material will be delivered to the assembly line using a drone.
(5) If the requested material is found in the warehouse, the material will be delivered to the assembly line using a drone.

Among the previous specified rules, rule 1 and rule 2 are redundant, since they have created the same actions. Likewise, rule 4 and rule 5 are redundant. Hence, these rules can be combined. Rule 3 can be an example of an unused rule, since in the rule how suppliers will be notified is mentioned by SMS. However, in rule 3, the end user wants that supplier employee to be notified through email. Figure 9 has demonstrated that the proposed framework has helped the end users to create the above-mentioned rules. Figure 10 has illustrated that the proposed framework has assist to find the bugs in the created TAP rules.

6 Comparison Study

In this section, the proposed work is evaluated based on the comparison with several selected existing works. All of these

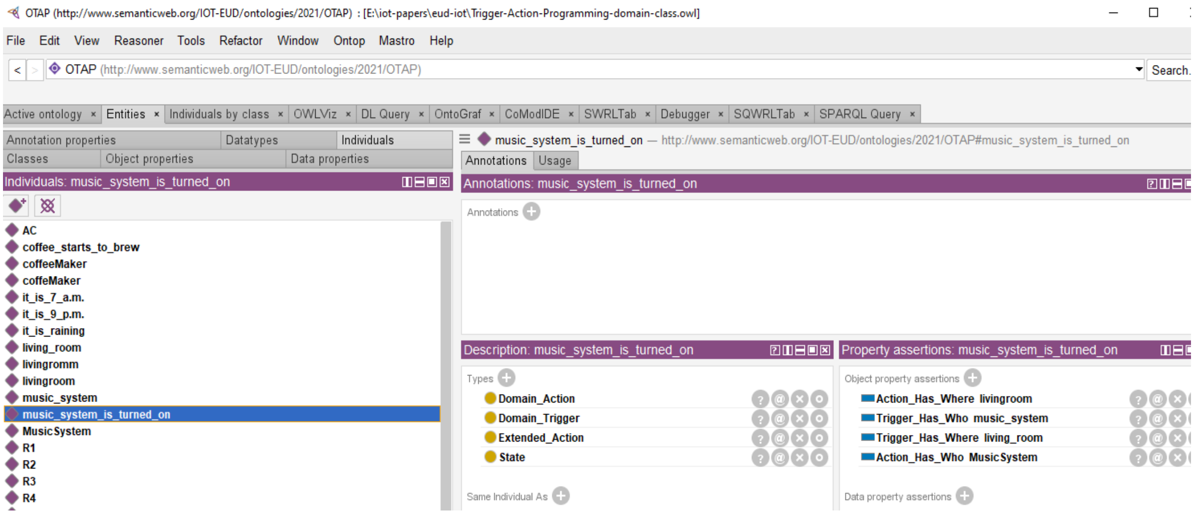


Figure 6: Partial view of OWL specification of TAO in Protégé tool [10]

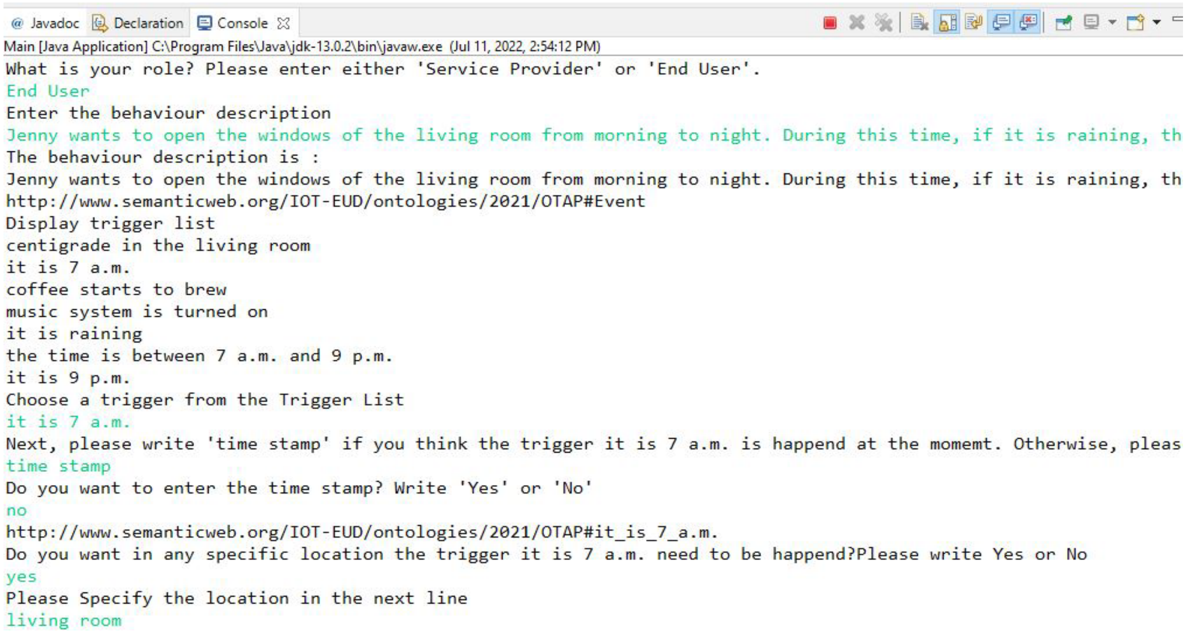


Figure 7: Illustration of Case study 1 - end users are supported to enter Answers of 5W1H questions and synthesize TAP rules

works have devised frameworks to support end users in synthesizing TAP rules. However, the proposed framework in this paper is distinct from these existing works due to possessing some useful features. The set of those useful features is specified next. Table 4 has summarized the comparison.

(a) *Trigger Related Features*: These kinds of features represent semantics related to different kinds of triggers – event and state.

(b) *Action Related Features*: These kinds of feature represent semantics related to different kinds of actions – immediate, extended and sustained.

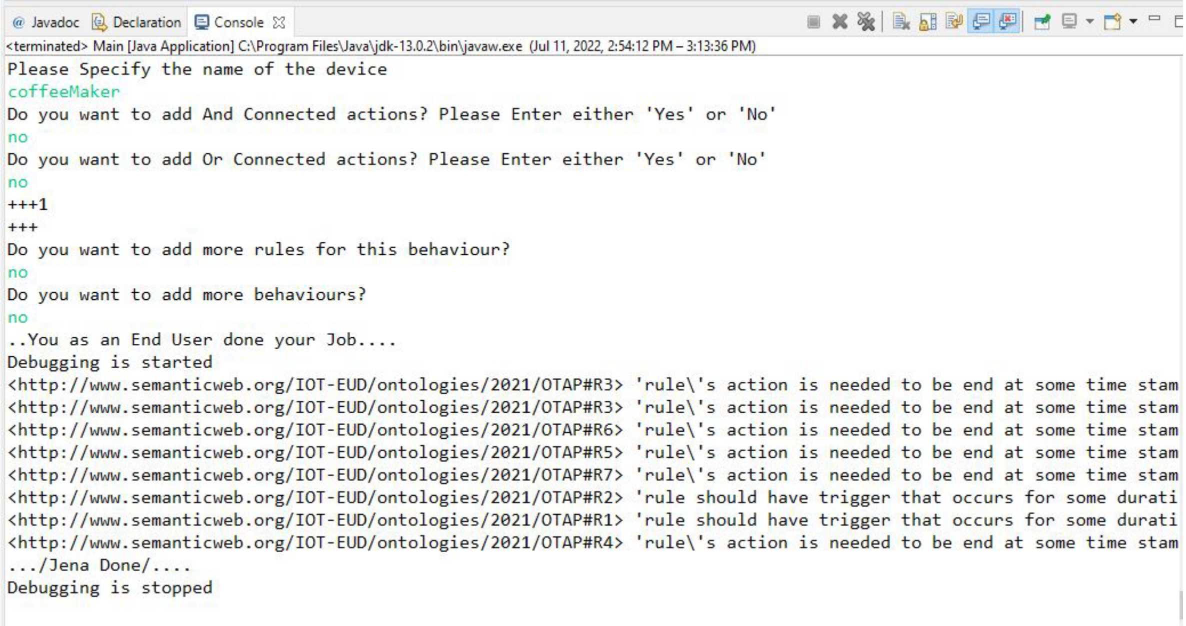
(c) *Triggering Feature*: This feature represents semantics that will specify the kind of triggering, single triggering or multiple triggering. Single triggering specifies that, a single trigger can activate a single action. On the other hand, multiple triggering represented the presence of either multiple triggers, multiple actions or both in a triggering

(d) *Context Support*: This feature represents the support towards representation of 5W1H contextual information. 5W1H has represented Why, Who, What, Where, When and How related context for a particular trigger and action.

(e) *Domain Independency:* This feature represents the scope of the application towards different domains.

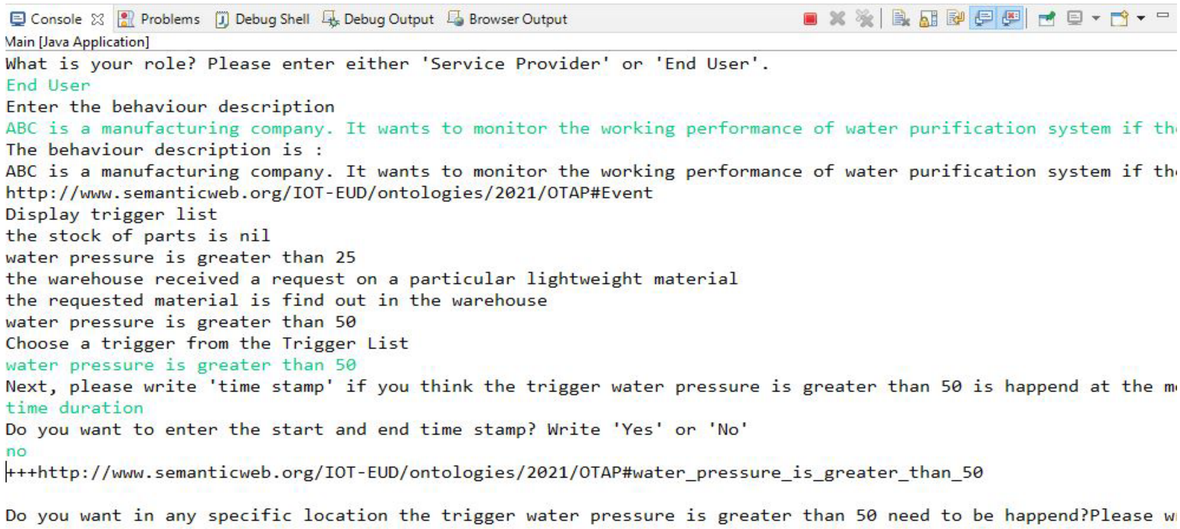
(f) *Reasoning Support:* This feature represents the support devise TAP rules.

(g) *Easy Interaction*: Interface between user and the framework needs to be simple and convenient for use. provided to the end users in making reasons when they



```
Javadoc  Declaration  Console
<terminated> Main [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Jul 11, 2022, 2:54:12 PM – 3:13:36 PM)
Please Specify the name of the device
coffeeMaker
Do you want to add And Connected actions? Please Enter either 'Yes' or 'No'
no
Do you want to add Or Connected actions? Please Enter either 'Yes' or 'No'
no
+++1
+++
Do you want to add more rules for this behaviour?
no
Do you want to add more behaviours?
no
..You as an End User done your Job....
Debugging is started
<http://www.semanticweb.org/IOT-EUD/ontologies/2021/OTAP#R3> 'rule\'s action is needed to be end at some time stam
<http://www.semanticweb.org/IOT-EUD/ontologies/2021/OTAP#R3> 'rule\'s action is needed to be end at some time stam
<http://www.semanticweb.org/IOT-EUD/ontologies/2021/OTAP#R6> 'rule\'s action is needed to be end at some time stam
<http://www.semanticweb.org/IOT-EUD/ontologies/2021/OTAP#R5> 'rule\'s action is needed to be end at some time stam
<http://www.semanticweb.org/IOT-EUD/ontologies/2021/OTAP#R7> 'rule\'s action is needed to be end at some time stam
<http://www.semanticweb.org/IOT-EUD/ontologies/2021/OTAP#R2> 'rule should have trigger that occurs for some durati
<http://www.semanticweb.org/IOT-EUD/ontologies/2021/OTAP#R1> 'rule should have trigger that occurs for some durati
<http://www.semanticweb.org/IOT-EUD/ontologies/2021/OTAP#R4> 'rule\'s action is needed to be end at some time stam
.../Jena Done/....
Debugging is stopped
```

Figure 8: Illustration of Case study 1 – Identification of bugs in synthesized TAP rules and reasons to fix those bugs



```
Console  Problems  Debug Shell  Debug Output  Browser Output
Main [Java Application]
What is your role? Please enter either 'Service Provider' or 'End User'.
End User
Enter the behaviour description
ABC is a manufacturing company. It wants to monitor the working performance of water purification system if th
The behaviour description is :
ABC is a manufacturing company. It wants to monitor the working performance of water purification system if th
http://www.semanticweb.org/IOT-EUD/ontologies/2021/OTAP#Event
Display trigger list
the stock of parts is nil
water pressure is greater than 25
the warehouse received a request on a particular lightweight material
the requested material is find out in the warehouse
water pressure is greater than 50
Choose a trigger from the Trigger List
water pressure is greater than 50
Next, please write 'time stamp' if you think the trigger water pressure is greater than 50 is happend at the m
time duration
Do you want to enter the start and end time stamp? Write 'Yes' or 'No'
no
+++http://www.semanticweb.org/IOT-EUD/ontologies/2021/OTAP#water_pressure_is_greater_than_50

Do you want in any specific location the trigger water pressure is greater than 50 need to be happend?Please w
```

Figure 9: Illustration of Case study 2 – end users are supported to enter Answers of 5W1H questions and synthesize TAP rules

(h) *Debugging Support*: The feature indicates the support to identification of bugs' presence and reasons to fix them.

(i) *Identification of Crucial Bugs*: This feature represents support to important bugs – (i) Time Window Fallacy, (ii) Contradict Rules, (iii) Indefinite Rules, (iv) Redundant Rules, (v) Lack of Action Reversal Bugs, (vi) Extended Action Related Bug, and (vii) Unused Rules.

(j) *Interoperability:* This feature indicates the representation of interoperable TAP rules that can be applied on different domains at the same time.

From Table 4 it is proved that the proposed framework has supported different specified features such as trigger, action, triggering and contextual information related semantics. On the other hand, the majority of existing approaches have represented trigger related semantics. Very few have recognized the importance of action, triggering and contextual information related semantics. However, semantics related to all of these building blocks of TAP are essential in order to interpret the create a TAP TAP rule effectively. Proposed framework can be applied to various domain on IE since it is based on an upper-level ontology specification. In addition, the proposed framework is domain independent and also capable to represent domain interoperable rules. This feature is not exhibited in the majority of the approaches. This reasoning support has been provided by very approaches partially. The majority of the existing approaches have support to identifications of bugs applied to various domains such as smart home, smart factory, autonomous vehicle etc. Two separate interfaces have been such as contradict action,

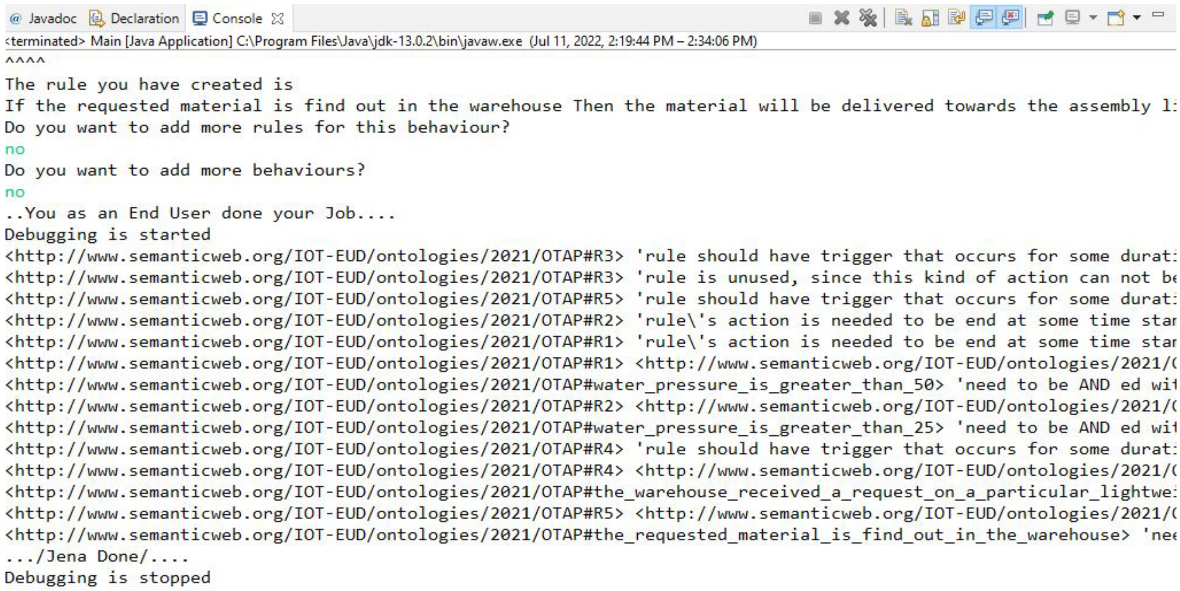


Figure 10: Illustration of Case study 2 – Identification of bugs in synthesised TAP rules and reasons to fix those bugs

Table 4: Comparison of proposed work with existing work based on several crucial features of TAP

Approaches	**(a)**	**(b)**	**(c)**	**(d)**	**(e)**	**(f)**	**(g)**	**(h)**	**(i)**							**(j)**
									(i)	**(ii)**	**(iii)**	**(iv)**	**(v)**	**(vi)**	**(vii)**	
[4]	Y	-	Y	P	-	P	Y	Y	P	Y	Y	Y	-	-	-	-
[23]	-	-	-	-	-	-	Y	Y	-	P	P	P	-	-	-	-
[24]	-	-	-	-	-	Y	Y	-	-	P	P	P	-	-	-	-
[21]	Y	-	P	P	Y	Y	Y	Y	P	Y	Y	Y	-	-	Y	Y
[15]	Y	Y	Y	P	-	-	Y	-	-	-	-	-	-	-	-	-
[18]	P	-	P	-	-	-	Y	Y	P	-	-	P	-	-	-	-
[3, 17]	Y	Y	Y	Y	-	-	Y	-	-	-	-	-	-	-	-	-
[25]	-	Y	P	-	-	Y	Y	Y	-	-	-	-	-	-	-	-
[7]	Y	Y	Y	Y	-	Y	Y	-	-	-	-	-	-	-	-	-
[16]	Y	P	Y	Y	-	Y	Y	Y	P	P	-	P	-	P	P	-
Proposed Framework	Y	Y	Y	Y	Y	Y	P	Y	Y	Y	Y	Y	Y	Y	Y	Y
Notes: P: Partial support; Y: Full support; –: Not Mentioned.																

redundant rules, and indefinite rules. Very few have support to other mentioned bugs. These drawbacks of the existing approaches can be resolved by the proposed framework. Further, the user interface of proposed framework is textual based which is inconvenient for end users. Hence, there is a need to devise a graphical user interface for the proposed framework and this need is identified as a future work.

7 Conclusion

In Intelligent Environment (IE) context, end user development still exhibits a lack of tool supports that can assist end users to create Trigger Action programming (TAP) rules effectively. In order to address this issue, this paper has proposed a semantic based reasoning framework that may help end users to create TAP rules. In addition, the proposed framework also can help end users to check the semantic correctness of the TAP rules. The contributions of the proposed work are manifolds. Firstly, the proposed framework is based on an upper-level ontology TAO thus, it can support end users to devise TAP rules based on precise and rigorous semantics. Secondly, it has proposed a rule based generic reasoner that can check the semantic correctness of the synthesized TAP rules. The proposed reasoner also helps end users to reason about the mistakes and fixing those. Thirdly, a set of inference rules have been proposed based on the formal semantics provided by the underlying upper ontology TAO. The proposed reasoner can check the semantic correctness of TAP rules based on these inference rules. The proposed inference rules are aiming to find crucial bugs present in TAP. Examples of those bugs are, time window fallacy, contradict rules, indefinite rules, redundant rules, lack of action reversal bugs, extended action, related bug, and unused rules. Fourthly, the proposed rules are domain independent based on an upper-level ontology. Hence, the proposed framework and the reasoner can be provided to support both service provider and end users. In addition, a comparative study has been conducted to demonstrate the usefulness of the proposed work. Future work includes developing a graphical user interface application based on the proposed framework that can make easy interaction with both end users and service providers. Further, an exploratory user test of the proposed framework in various domains will be important. In addition, a security vulnerability test of synthesized TAP rules will be a crucial future work. Moreover, integration of the proposed framework within the intelligent environment architecture will be a prime future work.

References

[1] Apache Jena. Retrieved June 10, 2022, from https://jena.apache.org.

[2] W. Brackenbury, A. Deora, J. Ritchey, J. Vallee, W. He, G. Wang, and B. Ur, "How Users Interpret Bugs in Trigger-Action Programming," *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1-12, May 2019.

[3] L. Corcella, M. Manca, F. Paternò, and A. Santoro, "A Visual Tool for Analysing IoT Trigger/Action Programming," International Conference on Human-Centred Software Engineering, Springer, Cham, pp. 189-206, September 2018.

[4] F. Corno, L. De Russis, and A. Monge Roffarello, "Empowering End Users in Debugging Trigger-Action Rules" *Proceedings of the 2019 CHI Conference on Human Factors in Computing System*s, pp. 1-13, May 2019.

[5] F. Corno, L. De Russis, and A. Monge Roffarello, "RecRules: Recommending IF-THEN Rules for End-User Development," ACM Transactions on Intelligent Systems and Technology (TIST), 10(5):1-27, 2019.

[6] N. C. Debnath, S. Banerjee, G. U. Van, and P. T. Quang, "An Ontology Based Approach towards End User Development of IoT," *Proceedings of 37th International Confer*, 82:1-10, 2022.

[7] G. Desolda, C. Ardito, and M. Matera, "End-User Development for the Internet of Things: EFESTO and the 5W Composition Paradigm," International Rapid Mashup Challenge, Springer, Cham, pp. 74-93, June 2016.

[8] G. Gallitto, B. Treccani, and M. Zancanaro, "If When is Better Than if (and While Might Help): On the Importance of Influencing Mental Models in EUD (A Pilot Study)," EMPATHY@ AVI, pp. 7-11, September 2020.

[9] N. Guarino, D. Oberle, and S. Staab, "What is an Ontology?" Handbook on Ontologies, Springer, Berlin, Heidelberg, pp. 1-17, 2009.

[10] M. Horridge, "A Practical Guide to Building OWL Ontologies using Protégé 4 and COODETools, Edition 1.3, The University of Manchester, Retrieved June 10, 2022, from https://mariaiulianadascalu.files.wordpress.com/2014/02/owl-cs-manchester-ac-uk_-eowltutorialp4_v1_3.pdf.

[11] J. Huang and M. Cakmak, "Supporting Mental Model Accuracy in Trigger-Action Programming," *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 215-225, September 2015.

[12] Matthew Hughes, "Mozilla's New Things Gateway is an Open Home for Your Smart Devices," TheNextWeb, 2018.

[13] IFTTT, Retrieved June 10, 2022, from https://ifttt.com/explore.

[14] Nat Levy, "Microsoft Updates IFTTT Competitor Flow and Custom App Building Tool PowerApps", GeekWire, 2017.

[15] M. Manca, F. Paternò, and C. Santoro, "Analyzing Trigger-Action Programming for Personalization of Robot Behaviour in IoT Environments," International Symposium on End User Development, Springer, Cham, pp. 100-114, July 2019.

[16] M. Manca, F. Paternò, C. Santoro, and L. Corcella, "Supporting End-User Debugging of Trigger-Action Rules for IoT Applications," *International Journal of Human-Computer Studies*, 123:56-69, 2019.

[17] A. Mattioli and F. Paternò, "A Visual Environment for End-User Creation of IoT Customization Rules with Recommendation Support," Proceedings of the International Conference on Advanced Visual Interfaces, pp. 1-5, September 2020.

[18] C. Nandi and M. D. Ernst, "Automatic Trigger Generation for Rule-Based Smart Homes," *Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security*, pp. 97-102, October 2016.

[19] Ed Oswald, "IFTTT Competitor Stringify gets a Major Update," TechHive, 2016.

[20] M. Palekar, F. Earlence and R. Franziska, "Analysis of the Susceptibility of Smart Home Programming Interfaces to End User Error," IEEE Security and Privacy Workshops (SPW), IEEE, pp. 138-143, 2019.

[21] C. Vannucchi, M. Diamanti, G. Mazzante, D. Cacciagrano, R. Culmone, N. Gorogiannis, and F. Raimondi, "Symbolic Verification of Event–Condition–Action Rules in Intelligent Environments," *Journal of Reliable Intelligent Environments*, 3(2):117-130, 2017.

[22] Zapier, Retrieved June 10, 2022, from https://zapier.com/.

[23] L. Zhang, W. He, J. Martinez, N. Brackenbury, S. Lu, and B. Ur, "AutoTap: Synthesizing and Repairing Trigger-Action Programs using LTL Properties," 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE), IEEE, pp. 281-291, May 2019.

[24] L. Zhang, W. He, O. Morkved, V. Zhao, M. L. Littman, S. Lu, and B. Ur, "Trace2tap: Synthesizing Trigger-Action Programs from Traces of Behavior," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(3):1-26 (2020).

[25] V. Zhao, L. Zhang, B. Wang, S. Lu, and B. Ur, "Visualizing Differences to Improve End-User Understanding of Trigger-Action Programs," Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems, pp. 1-10, April 2020.

Narayan C. Debnath is currently the Founding Dean of the School of Computing and Information Technology at Eastern International University (EIU), Vietnam. He is also serving as the Head of the Department of Software Engineering at EIU, Vietnam. Dr. Debnath has been the Director of the International Society for Computers and their Applications (ISCA), USA since 2014. Formerly, Dr. Debnath served as a Full Professor of Computer Science at Winona State University, Minnesota, USA for 28 years where he also served as the Chairperson of the Computer Science Department for 7 years. Dr. Debnath earned a Doctor of Science (D.Sc.) degree in Computer Science and also a Doctor of Philosophy (Ph.D.) degree in Physics. In the past, he served as the elected President and Vice President of ISCA, and has been a member of the ISCA Board of Directors since 2001. Professor Debnath has made significant contributions in teaching, research, and services across the academic and professional communities. Dr. Debnath is an author or co-author of over 500 publications in numerous refereed journals and conference proceedings in Computer Science, Information Science, Information Technology, System Sciences, Mathematics, and Electrical Engineering. Dr. Debnath is an editor of several books published by Springer, Elsevier, CRC Press, and Bentham Science Press on emerging fields of computing. He also served as a guest editor of the Journal of Computational Methods in Science and Engineering (JCMSE) published by the IOS Press, the Netherlands.

Shreya Banerjee is presently a Faculty Member in the Department of Software Engineering, Eastern International University, Vietnam. She received her PhD degree in Computer Science and Engineering from National Institute of Technology, Durgapur, India, in 2019. She received a Master degree in Software Engineering from National Institute of Technology, Durgapur, India, in 2013. She is an author and co-author of several publications in numerous refereed journals and conference proceedings. Her main areas of research interests include Software Engineering, Ontology Engineering, Semantic Reasoning, Database Management System and Service Oriented Computing.

Van-Giau Ung graduated MSc at HCMC University of Science in Dec 2015. After that, he continued working in software development companies for many years. He began teaching and research at University of Information Technology – Vietnam National University HCMC in 2017. In 2019, he moved to Software Engineering Department at Eastern International University. His research interests include Software Engineering, System and application Security, and Natural language processing. He loves to research new technologies, and share his knowledge and experience with students.

Phat Tat Quang received Master degree of Computer Science from VNU - University of Science, Ho Chi Minh, Vietnam in 2011, and Bachelor degree in Information Technology from Binh Duong University – Vietnam in 2007. Since 2011, he has been a full-time lecturer in School of Computing & Information Technology at Eastern International University, Binh Duong, Vietnam. His research areas include Computer Vision, Deep learning and their applications, and Software engineering.

Dai Thanh Nguyen joined The Eastern International University in 2015 as a lecturer in the School of Computing and Information Technology. He achieved a Bachelor's degree in Engineering in Computer Networks and Communication at Vietnam National University - Ho Chi Minh City - the University of Information Technology in 2012. He received his Master's degree in Computer Science at the University of Houston - Clear Lake, Houston, Texas, US in 2014. He is interested in software development and sharing knowledge to help students develop their careers

Univariate and Bivariate Entropy Analysis for Modbus Traffic over TCP/IP in Industrial Control Systems

Tirthankar Ghosh*, Sikha Bagui*, Subhash Bagui*, Martin Kadzis*,
Logan Day*, and Jackson Bare*
University of West Florida†, Pensacola, FLORIDA

Abstract

Anomalies in network traffic are usually detected by measuring unexpected deviation from what constitutes a baseline. Several statistical techniques have been proposed to create baselines and measure deviation. However, simply looking at traffic volume to find anomalous deviation may result in increased false positives. Traffic feature distributions need to be created, and deviations need to be measured for these features. An effective approach to finding anomalous deviations starts with entropy analysis on these features. This paper presents an entropy analysis on an industrial control system network using selected features with datasets obtained from an HVAC system. The paper starts with a fundamental question: whether preliminary entropy analysis on Modbus-over-TCP data using only a few TCP/IP features, without going into Modbus traffic, gives information about an anomaly in the network. Relative entropy was computed using Kullback-Leibler divergence to study deviation of malicious traffic from non-malicious. To gain further insight on detecting anomalies within the ICS traffic, the work was extended to bivariate joint entropy analysis using pairs of features. Initial analysis of the bivariate joint entropy also showed some promising results, but as in the univariate analysis, the bivariate joint entropy analysis showed that none of the feature pairs indicated a presence of reconnaissance†.

Keywords: Industrial control systems security, modbus traffic analysis, entropy analysis.

1 Introduction

Industrial Control Systems (ICS) are networks of devices used in critical infrastructure and industrial environments for control of physical processes. These networks typically span a large geographic area, and some examples of such systems are water distribution systems, gas pipelines, and power transmission systems. An ICS can be a large multifaceted infrastructure like a Supervisory Control and Data Acquisition (SCADA) system, which collects data and processes it in a centralized environment where it can be viewed and interacted with. There are simpler configurations of industrial control systems that are more readily available to any system with lesser changes to an already established system. Programmable Logic Controllers (PLC) are widely used in the area of automation and come equipped with programmable memory, various input and output channels, and communication interfaces that make them incredibly valuable.

Industrial Control Systems are fundamentally different from traditional Information Technology (IT) systems. Because ICSs provide an interface with physical devices like sensors and actuators, they are generally categorized as Operational Technology (OT). Updates and patching play a pivotal role in the differences in maintainability between IT and OT systems (Cardenas [6]). An ICS requires real-time availability for their systems to communicate and function, implying that they must be able to identify, diagnose, and respond appropriately to irregular flow of information as quickly as they appear. This necessitates that updates and patching happen as frequently as possible, making them as less vulnerable as possible to potential attacks. While this is a drawback that is difficult to circumvent, the design of an intrusion detection system may be simpler for ICS because of their static nature and predictability in communication.

Intrusion Detection and Prevention Systems (IDPS) are designed and deployed to monitor traditional IT systems for signs of undesirable and malicious behavior. A common way to secure a traditional system is to deploy an IDPS to monitor and identify anomalies that take automated action or warrant future investigation. There are two broad classifications of IDPS based on their detection technique misuse detection or signature-based, and anomaly detection or behavior-based (Angséus & Ekbom [3]). Signature-based systems are not very effective in ICS environments where very few known signatures are encountered. An effective way to identify communication anomaly in an ICS is to consider the minimal and maximal values in a given feature within the system and have precautionary measures in place that can react to values that defy the provided range (Angséus & Ekbom [3]). In other words, measuring the change in entropy or randomness in the communication pattern of an industrial control system is beneficial in detecting a problem.

There are four levels in a SCADA architecture that need to be secured to ensure a stable environment (Koucham [13]). The lowermost level, or the basic control level, generally consists of sensors and actuators that collect and send information to the upper levels. These machines are found at field sites with Remote Terminal Units. Above the basic control is the

*Email: tghosh@uwf.edu, bagui@uwf.edu, sbagui@uwf.edu, mjk34@students.uwf.edu, ldd14@students.uwf.edu, and cjb86@students.uwf.edu
† This work is partially supported by the Office of Naval Research (ONR) under grant number N00014-21-1-2025.

supervisory control level consisting of Distributed Control Systems (DCS) servers and PLCs which are accompanied by the Human Machine Interface (HMI) and engineering stations. This level focuses on the global view of the system's control state and operations and collects the information relayed from the first level for analysis. This information is also presented through the HMI to make it more easily accessible to a user reviewing the system's nodes and features. The engineering workstations allow the specification of setpoints and the programming of controllers, which allows for boundaries to be manipulated. The two uppermost levels are commonly grouped together as the "backbone network" and contain servers that are connected to the enterprise IT systems backbone. This level has a variety of purposes, which can most broadly be classified as the allocation and optimization of resources, maintenance, planning, and quality control. Data collected previously is housed within this level in database servers (Koucham [13]).

Communication characteristics in an ICS setting are also quite different from traditional TCP/IP communication. Sensors and actuators generate a low volume of data that is periodic with short transfer time and low delay (Koucham [13]). The controllers that accompany these nodes use a communication protocol like Modbus (Koucham, [13]), which is an application protocol that defines the syntax and semantics of the communication and structure. The Protocol Data Unit (PDU) is seven bytes long and consists of the transaction identifier, protocol identifier, length, and unit identifier. Figure 1 shows the Modbus frame and its encapsulation in the TCP header. The transaction identifier is used for transaction pairing when multiple messages are sent and make up two of the seven bytes. The protocol identifier also makes up two bytes and is either empty or padded with zeros to be used for future extensions. The unit identifier is one byte and identifies a remote server located on a non TCP/IP network, and the length is the byte count of the remaining fields. Modbus is open source making it the most widely used protocol in ICS environments (Koucham [13]).

In the Supervisory Level, a much higher volume of data is collected than sensors and actuators, and transfer time is restricted to a lesser extent. This level is also much more representative of a traditional IT system, and therefore can be treated as such. Network protocols like OPC DA and OPC UA are commonly utilized in this level, as they cover data access and client/server technology which are important to enable human interaction with the system and the data collected (Koucham [13]).

Although, because of their static nature and predictability in communication, securing industrial control systems may seem trivial at first, but there are several challenges that arise because of these very characteristics. We previously mentioned that having periodic system update is often a challenge in ICS environments, and real-time communication poses a challenge for deploying a security solution that adds latency. In addition, because of the operational technology requirements of ICS, physical interaction with the sensors and actuators poses a challenge to secure the system as a whole. Any intrusion detection and prevention systems that need to be designed for these environments must take into account the need for real-time data transfer and must be aware of stringent latency requirements. Hence, adding detection techniques that are resource-intensive may not be the most efficient approach in these ICS environments.

This paper discusses a univariate and bivariate joint entropy analysis on ICS Modbus over TCP/IP data and an analysis of relative entropy using Kullback-Leibler divergence.

1.1 Entropy

Entropy represents the amount of uncertainty that exists in a random variable X. Suppose the random variable X takes on

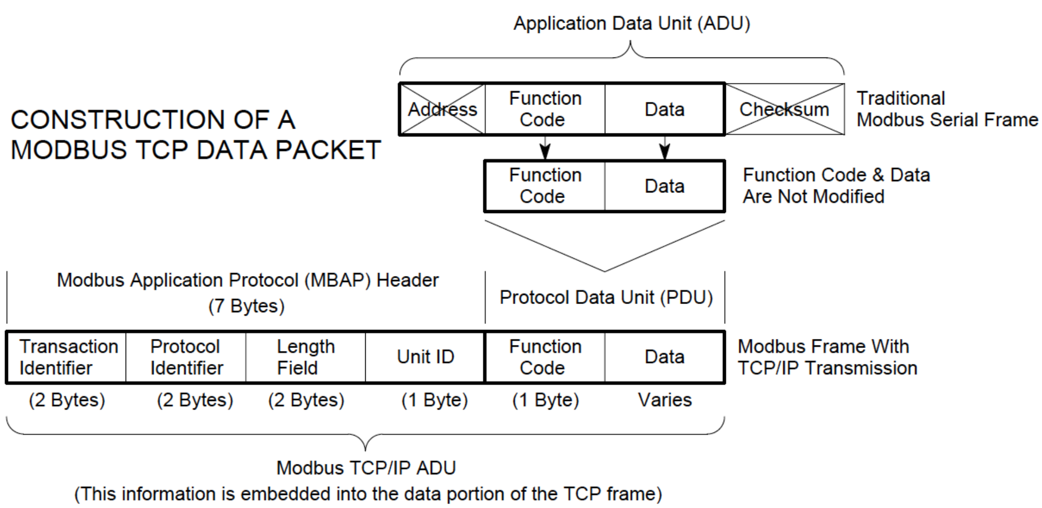


Figure 1: Modbus frame[1]

[1]Acromag Modbus TCP/IP Technical Reference.

values $x_1, x_2, \cdots, x_n$ with respective probabilities $p(x_1), p(x_2), \cdots, p(x_n)$, then the entropy of the random variable X is given by

$$E(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i)$$

which is equivalent to

$$E(X) = \sum_{i=1}^{n} p(x_i) \log(1/p(x_i))$$

where, $E(X)$ is the entropy value for the selected feature X, $p(x_i)$ is the probability that the feature X takes the value x_i. The quantity $-\log p(x_i)$ represents the surprise evoked if X takes on the value x_i. Thus, the expected amount of surprise after learning X is known as the entropy of the random variable X.

Looking at entropy values of a certain traffic feature over a period of time will provide a basis for possible anomalous deviation. However, if traffic features are analyzed independently and their entropy computed, it may not reflect an anomalous behavior and will lead to high false positives. In that respect, relative entropy analysis needs to be conducted. Relative entropy is computed as below:

$$D(p \| q) = \sum_{i=1}^{n} p(x_i) \log(p(x_i)/q(x_i))$$

which shows the relative entropy or deviation of the probability distribution $p(x)$ from the probability distribution $q(x)$. $D(p \| q)$ is called a measure of Kullback-Leibler divergence. The probability $q(x)$ is the probability distribution of non-malicious traffic and $p(x)$ is the probability distribution of malicious traffic.

1.2 Joint Entropy

Consider now two random variables X and Y, which take on respective values $x_1, x_2, \cdots, x_m$ and $y_1, y_2, \cdots, y_n$ with joint probability mass function

$$P(X = x_i, Y = y_j) = p(x_i, y_j) = p_{ij},$$
$$i = 1, \ldots, m; \quad j = 1, \ldots, n.$$

The joint entropy of the random vector (X, Y), may be denoted by $E(X, Y)$, is given by

$$E(X, Y) = -\sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij} \log p_{ij}.$$

The joint entropy $E(X, Y)$ represents the amount of uncertainty that exists in the random vector (X, Y).

As above, the relative entropy in joint probability distribution cases is defined as

$$D(\mathbf{p} \| \mathbf{q}) = \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij} \log(p_{ij}/q_{ij})$$

where $\mathbf{p} = \{p_{ij} : i = 1, \cdots, m; \, j = 1, \cdots, n\}$ could be the joint probability distribution of malicious traffic and $\mathbf{q} = \{q_{ij} : i = 1, \ldots, m; \, j = 1, \ldots, n\}$ could be the joint probability distribution of non-malicious traffic. $D(\mathbf{p} \| \mathbf{q})$ is viewed as a measure of divergence.

2 Related Literature

Supervised learning techniques have been proposed by many to detect malicious traffic in industrial control systems networks. In (Eigner [10]), the authors presented a prototype implementation of an anomaly-detection approach based on Naive Bayes. They ran simulated attacks - Denial of Service (DOS) and Man-in-the-Middle (MITM) - and did a preliminary analysis using the Bayesian classifier. The results were preliminary, and did not justify why and how a Bayesian classifier will be the most appropriate for an ICS network. In (Anthi [1]), the authors used a supervised learning approach to detect attacks. Their proposed attack detection system not only detects malicious packets but also classifies them as specific attack types. A limited number of features from the Modbus data was selected for classification. In (Goh [12]), the authors proposed to use Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) to detect a sequence of patterns for anomaly detection. It was used as a predictor to model normal behavior and then the cumulative sum method was used to identify malicious behavior. The authors tested the approach with data from only one process, limiting its scope and applicability. In (Valdes [16]), the authors used pattern-based and flow-based anomaly detection techniques to identify malicious communications. The pattern-based approach used patterns of communicating hosts to identify normal communication, whereas the flow-based approach used network flow to identify traffic patterns. These approaches typically work well in ICS environments as process related communications are static and predictable; however, they sometimes tend to be resource intensive as the network scales up. In (Feng, [11]), the authors proposed a two-stage anomaly detection technique using a packet level signature analysis and a time-series analysis using Long Short Term Memory Neural Network (LSTM -NN). In the signature analysis stage, a signature database for normal behavior was constructed using communication patterns among devices. The database was then passed through a Bloom filter to detect anomalous behavior. The time series analysis involved using LSTM-NN to learn the most likely package signatures from previously seen network packages. The authors used a SCADA dataset obtained from a gas pipeline to test their proposed technique. In (Caselli, [7]), the authors proposed a sequence-aware intrusion detection system that relied on pattern identification of ICS network events, extraction of semantic meanings, and modeling known behaviors over time. They used discrete time Markov chains to describe several ICS-specific operations and proposed a detection mechanism based on the computation of weighted distance among Markov chain states. In (Yang [17]), the authors proposed an anomaly detection technique using the auto-associative kernel regression model and statistical probability ratio test and applied the technique on a simulated SCADA network.

Not much work has been done using unsupervised learning to isolate malicious from non-malicious traffic. One persistent problem that remains is threshold selection. In (Almalawi [2]), the authors proposed an approach called global anomaly threshold to unsupervised detection (GATUD) that is used as an add-on component to improve the accuracy of unsupervised intrusion detection techniques. They used K-means clustering to initially learn two labeled small datasets from the unlabeled data; each dataset represents either normal or abnormal behavior. Then, a set of supervised classifiers were trained to produce an ensemble-based decision-making model that can be integrated into both unsupervised anomaly scoring, and clustering-based intrusion detection approaches to find a global and efficient anomaly threshold.

The work that came close to our proposed approach is in (Berezinski [4]). The authors proposed an entropy-based network anomaly detector, abbreviated as ANODE, to detect anomalies from network traffic capture. Although the proposed work was not ICS-specific, it provides a very good overall analysis on using entropy for detecting anomalous behavior and is the main motivation behind our approach.

Our paper presents univariate, relative, and bivariate joint entropy analysis on selected features using datasets obtained from an HVAC system. Initial findings from our analysis were reported in (Day [8]). As ICS sensors and actuators are resource constrained, and the OT system itself needs more real time monitoring and response, a computation intensive intrusion and attack detection method may not be suitable. We start from a fundamental question: whether entropy analysis on Modbus-over-TCP data using selected TCP/IP features, without going into Modbus traffic, gives information about an anomaly in the network. In the following sections, univariate entropy analysis and univariate relative entropy using Kullback-Leibler divergence are discussed and results are presented from our initial analysis of the data. Following that, bivariate joint entropy analysis is discussed and results are also presented from our bivariate joint entropy analysis.

3 Entropy Analysis for Modbus Over TCP/IP

3.1 Univariate Entropy Analysis

For the univariate entropy analysis, we based our analysis on a day's worth of Modbus over TCP/IP data collected from an HVAC system at the University of Alabama Huntsville research lab. The data had both malicious and non-malicious traffic representing various stages of the HVAC operations (auto cool, auto heat, normal, and random). Of the various attacks that were simulated, we chose the top three that have been discussed in the literature, namely, Denial of Service (DOS), Man-in-the-middle (MITM), and Reconnaissance.

One of the important considerations in any anomaly detection technique is feature selection. We based our preliminary analysis on only TCP/IP data with the simple objective of answering a fundamental question: whether preliminary entropy analysis on Modbus-over-TCP data using only few TCP/IP features gives us any information about an anomaly in the network. To deduce what features would be relevant in our initial entropy analysis, a correlation matrix was run against the TCP/IP data. The correlation matrix is presented in Table 1.

Three features packet size, inter-packet delay, and packet process time were chosen as they had the most significant

Table 1: Correlation matrix between features

	Packet Size	Inter-Packet Delay	Packet Process Time	Protocol Overhead	Protocol Efficiency	Throughput
Packet Size	1.00	-0.03	-0.44	-0.95	0.95	0.19
IP Delay	-0.03	1.00	0.07	0.02	-0.02	-0.01
PP Time	-0.44	0.07	1.00	0.23	-0.23	-0.04
Protocol Overhead	-0.95	0.02	0.23	1.00	-1.00	-0.20
Protocol Efficiency	0.95	-0.02	-0.23	-1.00	1.00	0.20
Throughput	0.19	-0.01	-0.04	-0.20	0.20	1.00

differences in their relationships with one another. Shannon's entropy, which is a direct measure of the bits needed to store the data in a given variable, was used for entropy computation for each feature for both non-malicious and malicious traffic (DDOS, MITM, and Recon). Table 2 summarizes the results. All three features inter-packet delay, packet process time, and packet size have higher entropy values under Denial of Service (DOS) and Man-In-The-Middle (MITM) attacks. Reconnaissance did not have much impact on entropy for packet size and packet process time but has a small increase for the inter-packet delay. This can be explained by the nature of reconnaissance, where probes are sent with a varying time lag. From Table 2, it can be seen that entropy can be a potential indicator to alert the system of some anomaly, although it will require further investigation to detect the actual nature of the anomaly as such.

3.2 Relative Entropy and Kullback-Leibler Divergence

One persistent question in anomaly detection is how much deviation or change is acceptable. To investigate that, we used relative entropy, or Kullback-Leibler divergence, which is a measure of the deviation of one probability distribution from another and is reflective of a realistic threshold that needs to be set to indicate an anomaly in network traffic. We used Kullback-Leibler divergence to measure deviation of malicious traffic distribution from non-malicious traffic distribution. Before analysis, however, it is important to note that these entropy levels are impacted by the volume of the attack; 45.3%, or nearly half, of the data, was of the attack type MITM, or Man in the Middle; Reconnaissance attacks made up 1.2% of the data; and DOS, or Denial-Of-Service attacks, made up 2.1% of the data. Table 3 summarizes the results from the Kullback-Leibler divergence computation. Divergence was computed for each of the malicious traffic categories from non-malicious traffic. Man-In-The-Middle (MITM) attacks had the largest divergence from non-malicious traffic, especially for packet size. This is not unusual given the nature of MITM attacks and the goals they want to accomplish. Reconnaissance traffic had the lowest divergence, which can be because of the very low volume of reconnaissance traffic within the sample (1.2%). DOS attacks had a similar effect on the system as MITM, however divergence for inter-packet delay was much less compared to the other two features. This could be a result of how DOS attacks flood a system and increase packet process times significantly.

It can be inferred from the results that attacks like MITM and DOS can be detected initially by looking at the entropy values of selected features. In contrast, reconnaissance may be

Table 2: Entropy values for three selected features against three attack types

	Packet Size	Inter-Packet Delay	Packet Process Time
Normal	1.987	2.332	2.957
Normal + MITM	2.173	2.488	3.006
Normal + Recon	1.987	2.393	2.957
Normal + DOS	2.066	2.891	3.008

Table 3: Relative entropy using Kullback-Leibler divergence

	MITM	Recon	DOS
Packet Size KL Divergence	3.176	0	0.952
Inter-Packet Delay KL Divergence	0.949	0.157	0.139
Packet Process Time KL Divergence	0.731	0	0.663

undetectable by initial entropy analysis. However, a more intricate joint entropy may be effective in detecting reconnaissance in the network, hence bivariate joint entropy analysis is done next. It is important to note that the percentage of each attack in the total flow of recorded traffic plays a significant role in the change of entropy; the more infected traffic within the data, the more likely it is to notice changes compared to non-malicious traffic regardless of what attack is being studied.

3.3 Bivariate Joint Entropy Analysis

To gain further insight on detecting anomalies within the ICS traffic, bivariate joint entropy analysis was computed using pairs of features. Bivariate joint entropy was computed for each pair of attributes: packet size, inter-packet delay and packet process time, for non-malicious traffic, malicious traffic, non-malicious + MITM, non-malicious + Recon and non-malicious + DOS. Further, instead of focusing on just a single day's worth of data, this analysis was extended to include eight days' worth of data. Tables 4 and 5 show 8-day average entropy and standard deviation values for each feature pair for non-malicious and malicious traffic, respectively. Results show that bivariate joint entropy analysis with packet process time and inter-packet delay can indicate presence of malicious activities, while the other feature pairs do not convincingly point towards that direction.

Table 4: Entropy averages and standard deviation for each pair of selected features for non-malicious traffic

	PS + IPD	PS + PPT	IPD + PPT
Average Entropy	4.705	4.609	4.839
Standard Deviation	0.063	0.090	0.087

Table 5: Entropy averages and standard deviation for each pair of selected features for malicious traffic

	PS + IPD	PS + PPT	IPD + PPT
Average Entropy	4.425	4.292	5.274
Standard Deviation	0.128	0.194	0.109

Breaking down the analysis into individual attack types, Table 6 shows 8-day average entropy and standard deviation values for each feature pair for non-malicious+MITM traffic. Comparing the results with the values for non-malicious traffic shown in Table 4, it can be inferred that bivariate joint entropy with any two of the selected features can convincingly indicate presence of man-in-the-middle attack in the network. Table 7 shows results for non-malicious+DOS traffic, indicating that bivariate joint entropy with packet size and inter-packet delay as well as with inter-packet delay and packet process time have some indication of denial-of-service attack, while bivariate joint entropy with packet size and packet process time does not indicate any such anomalous activity. Table 8 shows eight days average entropy and standard deviation values for each feature pair for non-malicious+recon traffic. Compared to non-malicious traffic in Table 4, it can be inferred that none of the feature pairs indicates presence of reconnaissance. This confirms with our analysis with univariate entropy computation presented earlier.

Table 6: Entropy averages and standard deviation for each pair of selected features for non-malicious+MITM traffic

	PS + IPD	PS + PPT	IPD + PPT
Average Entropy	4.930	4.847	5.041
Standard Deviation	0.066	0.104	0.083

Table 7: Entropy averages and standard deviation for each pair of selected features for non-malicious+DOS traffic

	PS + IPD	PS + PPT	IPD + PPT
Average Entropy	4.776	4.597	4.897
Standard Deviation	0.065	0.095	0.094

Table 8: Entropy averages and standard deviation for each pair of selected features for non-malicious+recon traffic

	PS + IPD	PS + PPT	IPD + PPT
Average Entropy	4.700	4.598	4.846
Standard Deviation	0.064	0.092	0.091

4 Conclusion and Future Directions

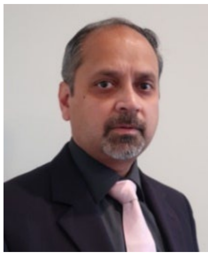
We presented an initial entropy analysis on an industrial control system network using selected features with datasets obtained from an HVAC system. We acknowledge that the initial entropy analysis only provides a starting point in asking several questions and investigating relevant issues that will lead to optimal system design and implementation. We started from the fundamental question: whether a preliminary univariate entropy analysis on Modbus-over-TCP data using only a few TCP/IP features, without going into Modbus traffic, gives us information about an anomaly in the network. For the univariate entropy analysis, we based our work on a day's worth of Modbus over TCP/IP data collected from an HVAC system at the University of Alabama Huntsville research lab. The data had both malicious and non-malicious traffic representing various stages of the HVAC operations (auto cool, auto heat, normal, and random). Of the various attacks that were simulated, we chose the top three that have been discussed in the literature, namely, Denial of Service (DOS), Man-in-the-Middle (MITM), and Reconnaissance. We also used Kullback-Leibler divergence to measure the univariate relative entropy of selected features over non-malicious traffic for each of the malicious activities. Initial analysis showed some promising results using univariate entropy and divergence. To gain further insight on detecting anomalies within the ICS traffic, we extended our work to bivariate joint entropy analysis using pairs of features with more data extending it to eight days-worth of traffic data. Initial analysis of bivariate joint entropy also showed some promising results, but as in the univariate analysis, none of the feature pairs indicated a presence of reconnaissance.

However, there are several questions that need to be addressed as below.

1. Do these entropy and divergence values give us a realistic threshold for anomaly detection? These need to be analyzed with more days' worth of data.
2. Does the percentage of malicious traffic have a causal effect on the entropy values? These would also need to be analyzed with more data and more malicious traffic.
3. What would be the entropy values for the ModBus traffic features? Do those features give us a more holistic view of an anomaly in the network traffic?
4. Would multivariate joint entropies with both Modbus and TCP/IP features give us a better understanding of network anomaly? A multivariate analysis with more features would be needed to answer this question.

References

1. E. Anthi, L. Williams, P. Burnap, and K. Jones, "A Three-Tiered Intrusion Detection System for Industrial Control Systems," *Journal of Cybersecurity,* 7(1):1-10,2021, doi:10.1093/ cybsec/tyab006, 2021.
2. A. Almalawi, A. Fahad, Z. Tari, A. I. Khan, N. Alzahrani, and AS. T. Bakhsh, and S. Qaiyum, "Add-On Anomaly Threshold Technique for Improving Unsupervised Intrusion Detection on SCADA Data," *Electronics,* 9(6):2-20, 1017. doi:10.3390/electronics9061017, 1017.
3. J. Angseus and R. Ekbom, *Network-Based Intrusion Detection Systems for Industrial Control Systems,* Master's Thesis, in Computer Science, Chalmers University of Technology, Gothenburg, Sweden, 2017.
4. P. Bereziński, B. Jasiul, and M. Szpyrka, "An Entropy-Based Network Anomaly Detection Method," *Entropy,* 17(4):2367-2408. doi:10.3390/e17042367, 2015.
5. S. Caltagirone, "Industrial Control Threat Intelligence," *Dragos*, 2018.
6. A. Cardenas, S. Amin, B. Sinopoli, A. Giani, A. Perrig, and S. Sastry, "Challenges for Securing Cyber Physical Systems." *Proc. of Workshop in Cyber Physical Systems*, pp. 9-25, July 23, 2009.
7. M. Caselli, E. Zambon, and F. Kargl, Sequence-aware Intrusion Detection in Industrial Control Systems, *Proc. of the 1st ACM Workshop on Cyber-Physical System Security,* ACM, pp. 13-24, 2015
8. L. Day, T. Ghosh, S. Bagui, and S. Bagui, "Entropy Analysis for Modbus Traffic over TCP/IP in Industrial Control Systems," *Proc. of Computer and its Applications Conference* 2022, CATA'22, pp. 63-71, 2022
9. Z. Drias, A. Serhrouchni, and O. Vogel, "Taxonomy of Attacks on Industrial Control Protocols" *NTDS*, doi:10.1109/NOTERE.2015. 7293513, 2015
10. O. Eigner, P. Kreimel, and P. Tavolato, "Attacks on Industrial Control Systems - Modeling and Anomaly Detection." *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, pp. 581-588, 2018, doi:10.5220/000675540 5810588, 2018.
11. C. Feng, T. Li, and D. Chana, "Multi-Level Anomaly Detection in Industrial Control Systems via Package Signatures and LSTM Networks," *Proc. IEEE Int. Conf. Depend. Syst. Network*, pp. 261-272, 2017
12. J. Goh, S. Adepu, M. Tan, and Z. S. Lee, "Anomaly Detection in Cyber Physical Systems Using Recurrent Neural Networks," 2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE), doi:10.1109/hase.2017.36, 2017.
13. O. Koucham, "Intrusion Detection for Industrial Control Systems," *Universite Grenoble Aples*. Doi:tel-02108208, 2018.
14. Q. Lin, S. Verwer, R. Kooji, and A. Mathur, "Using Datasets from Industrial Control Systems for Cyber Security Research and Education," *Proc. of the 14th International Conference on Critical Information Infrastructures Security CRITIS*, Linköping, Sweden, pp. 122-133, 23-25 September, 2019.
15. T. Morris and W. Gao, "Industrial Control System Traffic Data Sets for Intrusion Detection Research," doi:10.1007/978-3-662-45355-1_5.
16. A. Valdes and S. Cheung, S. (2009). "Communication Pattern Anomaly Detection in Process Control Systems," *2009 IEEE Conference on Technologies for Homeland Security*. doi:10.1109/ths.2009.5168010, 2009.
17. D. Yang, A. Usynin, and J. Hines, "Anomaly-Based Intrusion Detection for SCADA Systems," *Proc. of the 5th International Topical Meeting on Nuclear Plant Instrumentation Controls, and Human Machine Interface Technology*, Albuquerque, NM, USA, pp. 12-16, 2006.



Tirthankar Ghosh is Professor and Associate Chair in the Department of Computer Science at UWF. He has over seventeen years of experience in cybersecurity education and research in network security, ICS security, anomaly detection and adversary TTPs, and threat intelligence. Dr. Ghosh has received numerous grants from federal and state agencies. He was the co-founder of a state-wide consortium in Minnesota and a regional cybersecurity consortium in St. Cloud. Dr. Ghosh authored a book titled "Security by Practice: Exercises in Network Security and Information Assurance", and several journal papers and book chapters. He serves as an ABET evaluator for Cybersecurity and Computer Science.

Sikha Bagui is Distinguished University Professor and Askew Fellow in the Department of Computer Science, at The University West Florida, Pensacola, Florida. Dr. Bagui is active in publishing peer reviewed journal articles in the areas of database design, data mining, Big Data analytics, machine learning and AI. Dr. Bagui has worked on funded as well unfunded research projects and has 100+ peer reviewed publications, some in highly selected journals and conferences. She has also co-authored several books on database and SQL. Bagui also serves as Associate Editor and is on the editorial board of several journals.

Subhash C. Bagui received his B.Sc. in Statistics from University of Calcutta, M. Stat. from Indian Statistical Institute and Ph.D. from University of Alberta, Canada. He is currently a University Distinguished Professor at the University of West Florida. He has authored a book titled, "Handbook of Percentiles of Non -Central t-Distribution", and published many high quality peer reviewed journal articles. He is currently serving as associate editors/ editorial board members of several statistics journals. His research interests include nonparametric classification and clustering, statistical pattern recognition, machine learning, central limit theorem, and experimental designs. He is also a fellow of American Statistical Association (ASA) and Royal Statistical Society (RSS).

Martin Kadzis (photo not available) is a graduate student of Computer Science at the University of West Florida. He is a recipient of CyberCorps: Scholarship for Service at UWF.

Logan Day (photo not available) is a senior in the Software Development/Computer Science major at the University of West Florida. He is a ROTC cadet at UWF.

Jackson Bare is a senior Cybersecurity major at the University of West Florida (UWF). Mr. Bare has participated in several research projects ranging from the use of drones to achieve political objectives to comparing different methods of growing kale to Industrial Control System entropy analysis. Mr. Bare has presented at the National Center for Undergraduate Research Symposium and the UWF Research Symposium. He is a CyberCorps: Scholarship for Service recipient at UWF.

The Implementation of Content Planner Application with MobileNetV2 Classification for Hashtag Automation

Bevan Christian* and Trianggoro Wiradinata*†
Universitas Ciputra, Surabaya, INDONESIA

Abstract

Social media has become the most common communication channel in the 21st century. This progress opens up job opportunities for content developers, especially social media users who want their content to be managed professionally. A content developer handles several types of social media simultaneously for several clients that must be managed. Data shows that social media content developer workers experience fatigue because they have to manage a lot of content, thus sacrificing post quality if there are no tools that make their work easier.

This study aims to design and develop a content planner application with the feature of making automatic hashtag creation. Identification of app features was made through interview processes with several content developers. Among the interview results, it was found that they expressed difficulties when using applications that were not directly integrated with several popular social media, in addition to complaints about the difficulty of determining captions and hashtags for each post on social media.

This research created an application that helps content developers manage social media posts with helpful features such as implementing image classification using MobileNetV2 for hashtag automation with 72% accuracy. The direct test results found that the content planner application with the automatic hash feature helps content developers to manage posting content easier.

Key Words: Mobile application, iOS framework, social media, image classification, CNN, mobilenetv2.

1 Introduction

Indonesia's social media users have increased rapidly during the last five years. According to Digital 2022: Global Digital Insights, of 277.7 million Indonesians, 191.4 million use social media. Therefore, it can be concluded that the penetration rate is about 68.9% [5]. Having identified the significant potential of social media, some of their benefits to a company are wide scope of communication, minimal fees, brand awareness, and increased sales [4, 8, 10]. Even though many companies have Which translates to an increase in company expenditures. It is realized the benefits of social media, however, managing social media-related activities internally require time from employee social media developers who, although expensive, if done right, may be worth the investment [8].

Such demands give rise to new job positions such as Social Media Managers, Social Media Advertisement consultants, Social Media Celebrities known as Celebgram, TikTok-ers, and many more. Alongside the growing job opportunities is a growing workload challenge for content creators. They are tasked with handling various social media platforms, finding accurate SEOs, handling and avoiding copywriting issues, and creating feed and relevant hashtags. Studies conducted by Laucuka [9] concluded that hashtags cause the public to perceive something as bigger and more versatile.

Handling multiple social media platforms simultaneously means that the content created needs to be adjusted accordingly, which takes up valuable time. This overwhelms content development due to the enormous amount of administration work, including scheduling [6]. Several studies have shown that employees working in the social media area work more than their coworkers from other fields and plan to leave their job in two years due to fatigue [2]. This condition needs dire attention. An app to assist in finishing administrative and scheduling activities is needed by content developers so that they can focus more on creating content.

2 Literature Review

Newspapers and magazines are examples of content platforms that regularly publish a few items on their respective social media accounts and advertise some of them with tailored content (TCA). By publishing articles, content platforms can expand their social media followings and make money from impressions generated by clicks on links in social media posts. To schedule social media posts and TCA in a way that would optimize profitability, content platforms must decide what to post, when to post it, and whether and how much to spend on TCA. Social media managers bemoan this complexity, and the academic literature provides little help [4]. This study is interested in knowing the effects of content scheduling.

Another study by Rokhana, et al. [12] suggests using MobileNetV2 architecture as the basic model for a multi-class image classification that can identify when a face mask is being used correctly. This study also provides a trainable head model for the network, which consists of two fully connected layers, a depthwise convolution layer, and a strong classification

* School of Information Technology,
† Email: twiradinata@ciputra.ac.id.

performance. The experimental findings demonstrate the proposed system's high multi-class classification ability, with accuracy, precision, recall, and F1-score values of 97%, 97%, and 97%, respectively. This study suggested an example of object identification implementation in a real-time application due to its lightweight network architecture.

2.1 Terminology

1) iOS

A mobile operating system developed by Apple, iOS, runs software on devices such as iPhone, iPad, and iPod touch. It was earlier introduced as iPhone OS, but was changed to iOS with the introduction to iPad. Since 2019, iPad has its own OS, called iPadOS [1].

2) Swift

Swift is a robust and intuitive programming language created by Apple to develop applications for iOS, Mac, Apple TV, and Apple Watch. Swift alone could be used to develop games, machine-learning-based applications, and much more [7].

3) Core Data

Apple's macOS and iOS operating systems include Core Data, an object graph, and a persistence mechanism. It debuted in iOS with iPhone SDK 3.0 and Mac OS X 10.4 Tiger. It enables the serialization of data from the relational entity-attribute model into XML, binary, or SQLite databases [3].

4) Core ML

Core ML builds a model by applying a machine learning algorithm to a training data set. Making predictions based on fresh input data requires the usage of a model. A wide range of tasks that would be challenging or impractical to write in code can be carried out using models. With just a few lines of code, developers may use Core ML's four dedicated machine learning APIs to take advantage of machine learning capabilities. The first of these APIs is the Vision Framework, which uses computer vision techniques including object identification, image classification, and activity categorization to analyze photos and videos. (2) The Natural Language Framework can assist developers in the analysis of natural language content, the division of that material into paragraphs, phrases, or words, and the tagging of information about those segments. (3) The Speech Framework allows users of various languages to benefit from speech recognition on real-time or recorded audio. (4) Sound Analysis Framework, used to categorize noises like highway noise or singing birds, analyzes audio [11].

5) MobileNetV2

One of several phone-based Convolutional Neural Network (CNN) Architectures that can accommodate more computing power is MobileNetV2. The most recent version is MobileNetV2. In general, the use of convolution layers distinguishes the MobileNetV2 Architecture from the CNN Architecture. MobileNetV2 uses depthwise separable convolutions, which greatly reduce the number of parameters when compared to convolution layers generally, as seen in Figure 1.

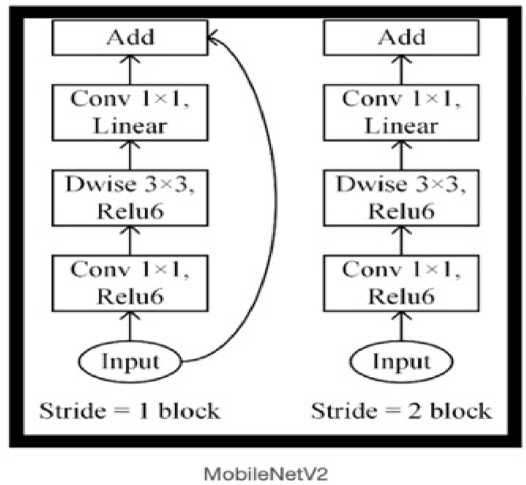


Figure 1: MobileNetV2 architecture

Depending on the input image's thickness, the MobileNetV2 convolution layer applies the appropriate filter thickness. MobileNetV2 makes use of linear bottlenecks, pointwise and depth wise convolutions, as well as shortcut connections between bottlenecks [13].

Furthermore, using the MobileNetV2 has the following benefits:

1. Less memory and power consumption.
2. Transfer learning allows the trained model to be used with a new dataset, hence saving time.
3. Up to 75% accuracy is deemed high.
4. It is perfect for devices with lower-level technology due to its compact model size.
5. MobileNetV2 beats sophisticated real-time detectors in the construction of models for object detection using COCO dataset in terms of accuracy and complexity by 10 times less calculations.

3 Design & Analysis

3.1 Analysis of the Problem

Interviewing content creators to understand the difficulties they encountered at work is the first step in analyzing the application. To find comparable existing applications, the analysis proceeds with a survey of similar apps on the App Store.

Content creators must pay attention to several factors while managing social media, including the date and time the post was made, the description, hashtags, and even the type of content (either video or image). The above are arranged using spreadsheet tools based on the researcher's personal experience. The drawback of this approach is that image files are saved separately from the spreadsheet, taking time to locate and manually submit the image file from wherever it may be. Three content developers with at least six months of experience managing social media were then questioned to acquire more information on the previously described problem.

The purpose of the content developer interview was to learn more about:

1. The scheduling tools utilized for content.
2. Daily routines.
3. Drawbacks related to employing current tools.

Several things could be inferred from the interview's findings, including:

1. Three main tools are used to schedule content. a TikTok draft, Microsoft Excel, and a calendar application.
2. Most content developers rely on two key tools. The first is a calendar application that can just serve as a reminder when material needs to be submitted. The content developers utilize Microsoft Excel to store the information, captions, and hashtags.
3. Content developers must manually copy and paste data from Microsoft Excel.
4. Other schedules and content scheduling are combined by using a calendar program.
5. Content developers frequently overlook posts that were late or hadn't been uploaded.
6. To correctly target consumers based on each social media site, content developers must differentiate the captions, hashtags, and when a post should be published.
7. Hashtags have grown in importance as a means of boosting content popularity but coming up with the best ones may be time-consuming.

A few applications for content scheduling were discovered. There is an app that offers the ability to post and preview feed content automatically. Still, it cannot propose hashtags and requires a relatively high price to enjoy the full features. Another app was found with the capability of content post-time planning. However, it has been discovered to be unreliable regarding timely content scheduling. Finally, there is another app with complete features for content developers, except proposing hashtags.

3.2 Analysis of Needs

Content developers need an application with the following functionalities based on business-related issues:

1. Informs the creator of the material to post in accordance with the content schedule.
2. In order to eliminate the need to switch between applications, one application houses all media content.
3. The capacity to store image and video files.
4. The ability to provide a video or image file preview.
5. The ability to post content with a single click eliminates the requirement for manual posting.
6. A preview feed that is sorted according to social media networks
7. Shows overdue contents.
8. A hashtag generator that automatically creates hashtags based on image recognition.

3.3 Application Architecture

The application architecture diagram is depicted in Figure 2 below. Users first visit a view, often known as the "Home" display, where data is given to the view model for processing each input the content developer adds. For Create-Read-Update-Delete (CRUD) processing, some essential operations will start at the view model and continue in the model or local database. Then, as feedback, either a success or an error some data will be returned. The MobileNetV2 method can be used by the view model to request classification labels for automatic hashtag generation. The findings are returned to the view model to be processed into hashtag suggestions as soon as the categorization process is complete.

Data from the *model* that is sent to the *view model* is handled in accordance with each view rather than being sent directly to the view.

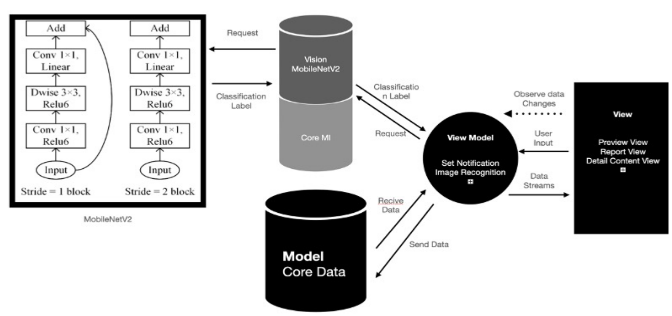


Figure 2: Application architecture

3.4 Usecase Diagram

The content developer, can perform one of the two main activities, as seen in Figure 3. The first is to create content, and the second is to examine post scheduling report data. CRUD operations could be utilized to process the creation of content, particularly in Create User where hashtags are generated automatically and in Read User where content can be disseminated straight to selected social media.

3.5 Application and Design Flow

Based on results from interviews, research, and application comparison, the content will be divided based on social media and for each social media platform, a preview of the feed and the feature to add new content (video / image-based) with *captions* and automatically generated hashtags based on *image recognition.* Another page is dedicated to reporting content that is on track and overdue concerning the content schedule as shown in Figure 4.

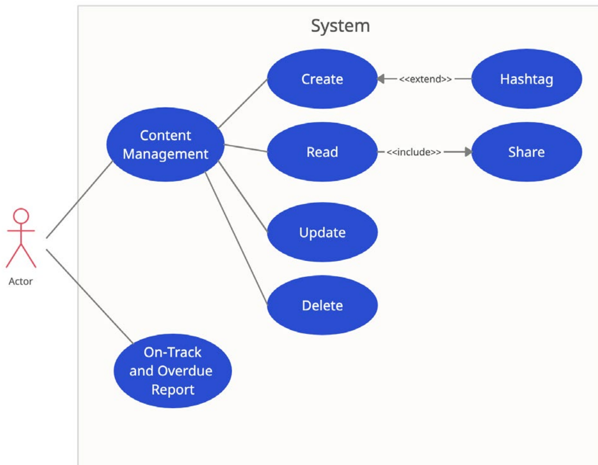


Figure 3: *Use case* diagram

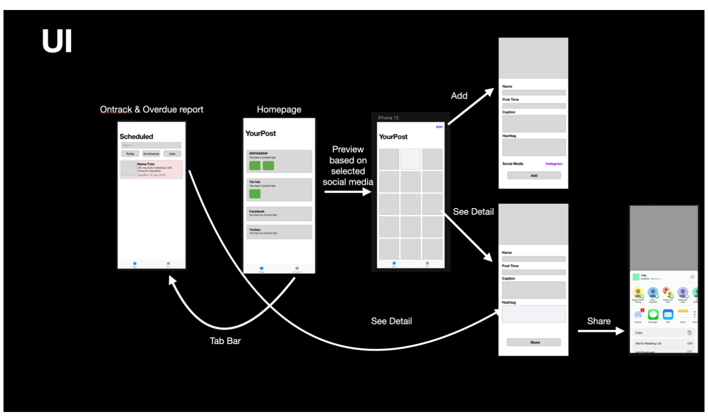


Figure 4: Application flow

4 Implementation

4.1 Automatic Hashtag Generation

To generate automatic hashtags, a deep learning model, in this case an image recognition model, needs to be initialized with the following method.

```
let model = MobileNetV2()
```

Afterwards, carefully selected content will be identified as either video-based or image-based to generate the appropriate hashtag.

```
func append(item: PhotoPickerModel) {
      itemsContentPicOrVid = item
      if  itemsContentPicOrVid.mediaType
== .photo {
           clasifyImage()
      } else {
               clasifyVideo()
         }
     }
```

4.2 UI Display

1) Home page

Four social media platforms chosen for the home page based on their level of popularity are listed based on the interview. Instagram, Tik-Tok, Facebook, and Twitter are the available channels as displayed in Figure 5. Content developers may simply determine how much of their content is past due, thanks to this application's homepage.

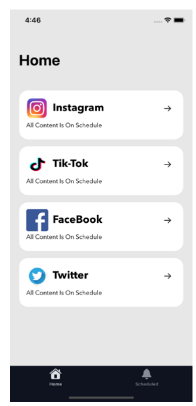


Figure 5: Homepage

2) Preview Feed

The *preview feed* page (Figure 6) is used by content developers to get a preview before posting their feed on social media and which content comes next. Pages could be displayed and ordered based on posts criteria.

3) Adding content.

On this page, (Figure 7) content developers can add new content. From adding an image or video-based content to *captions,* setting when to post, when to be notified, and acquiring an auto-generated hashtag based on the image or video-based content they uploaded and the level of accuracy.

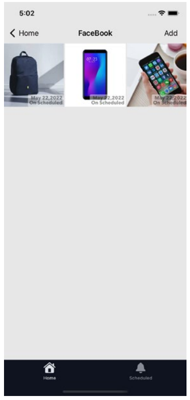


Figure 6: Preview feed page

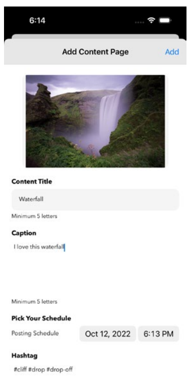


Figure 7: Add content page

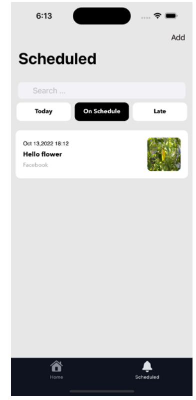


Figure 8: On-track & overdue report display

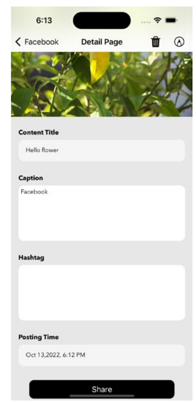


Figure 9: Content detail page

4) On-track & Overdue Report

On this page, content developers can track contents that are overdue, on time, or due that day (Figure 8). This page also allows content developers to add content with the *add* button at the top right and do searches.

5) Content Detail

The caption, image or thumbnail of the video, and hashtags are among the additional information about the posted content that is displayed on this page (Figure 9). The user can share anything on this page to any social media site. The *UIActivityViewController* will be called by this action later to establish a connection with the specified social media. The caption and hashtags will be preserved in the device's clipboard when the content creator shares content, saving time by eliminating the need to redo them.

5 Results and Discussion

User Acceptance Testing (UAT) was chosen as the application testing approach in this study, in the direction of the content developer who created a user persona based on the testing scenario. As a result, the testing scenario has also been modified to reflect the unique traits and environment of each user persona. Additionally, depending on the outcomes of the interviews, the user persona attributes were established. Here are three different types of user personas that were chosen for this study.

5.1 User Persona & Test Case

User persona 1 is an *influencer* who struggles with following the content schedule and posting on time and must manage various social media to support the influencer's work in Table 1.

User persona 2 is business owners responsible for handling business operations, including their businesses social media. This persona struggles with time constraints, handling multiple social media at the same time and posting dense content as shown in Table 2.

Last is *user persona* 3 (Table 3), the *Content Manager* who struggles with determining hashtags for their content and scheduling content.

Table 1: Influencer user persona

Persona Type: *Influencer*		
Pains	**Goals**	**Task**
Handling much content	Ease of sharing content	Sharing video on social media.
Not following content schedule	Enables the user to remember the content schedule easily and receive notification.	Track the content that must be posted on said day and receive notification.
Handling varieties of social media.	Enable the user to manage content folders and scheduling.	Post content on 3 different social media platforms.

Table 2: Business Owner User Persona

Persona Type: *Business Owner*		
Pains	**Goals**	**Task**
Time constraint	Able to save and share content quickly.	Add video-based content and share to desired social media, without writing any captions.
Handling various social media	Easily handle multiple social media at the same time.	Updating and deleting available content.
Personal files mixed with work files.	Able to separate personal files from content for social media.	Add content to various social media.

Table 3: Content manager user persona

Persona Type: Content Manager		
Pains	**Goals**	**Task**
Difficulty creating hashtags and captions	Easily create hashtags	Uploading content to generate hashtags.
Difficulty scheduling content.	Easily schedule content.	Able to interpret schedules in on-track and overdue report.

5.2 Test Results

Table 4: Influencer user persona

Task	**Preconditon**	**Postcondition**	**Result**
Add 3 content for different social media types at different times.	1. User grants access to their gallery. 2. User inputs necessary input for completing content. 3. User presses *add*	User can easily add content without difficulty and free from error.	Passed
Track content that needed to be posted today, overdue, and on-track content.	1. Content has been added beforehand. 2. Three types of content must be added. One to be uploaded today, the second to be posted sometime in the future and the third preceding today.	User can see and understand *segmented control* easily and can interpret in accordance with their respective *segmented control.*	Passed
Sharing a video to social media	1. Video-based content has been added beforehand. 2. Presses one of the social media to be distributed.	Easily share video-based content to social media.	Passed
Receiving notifications on schedule.	1. User has previously saved content. 2 User did not open the app on scheduled day.	A notification will appear according to schedule along with the video / image attachments. Subsequentially the social media destination can be selected.	Passed

Table 5: Business owner user persona

Task	Preconditon	Postcondition	Result
Add video-based content.	1. User grants access to their gallery application. 2. Gallery has a video for selection.	User can add video-based content easily.	Passed
Sharing video to intended social media.	1. User had saved content in video form. 2. User owns social media, capable of sharing videos on their platform.	User could easily share content, captions and hashtags to target social media platforms.	Passed, a suggestion to put the share button at the top.
Add as much content as possible and successfully do CRUD related processing.	1. Grant access to the gallery app. 2. A video is available in the gallery for selection.	User can easily change or erase data.	Passed
Adding image-based content to various social media at the same time.	1. User gives access to gallery application. 2. Gallery contains an image to be uploaded.	User could share pre-made content, captions, and hashtags consistently across social media platforms.	Passed, suggestion to add navigation title based on which social media platform is being accessed and swipe down to refresh feature.
Share image or video - based content without *captions*	1. Has a video file. 2. Captions and hashtags are inputted correctly. 3. Has previously selected a target social media platform.	User could easily share content without having to rewrite captions.	Passed

Table 6: Content manager user persona

Task	Precondition	Postcondition	Result
Adding content to automatical-ly generate hashtags.	1. User grants access to gallery application. 2. Has selected a video or image file.	User could effortlessly receive hashtag recommendati ons based on available content.	Passed, Suggestio n to make social media selection consistent with results.
Able to compre-hend the content schedule made.	1. Has previously added content. 2. Three types of content must be added. One to be uploaded today, the second to be posted sometime in the future and the third one preceding today.	User can see and understand *segmented control* in the on-track & overdue view easily and can interpret it in accordance with their respective *segmented control.*	Passed, Suggestio ns to add time in on-track and overdue reports, a cancel button in the search bar, and adding a haptic touch feature to preview content details.

Out of 11 successful tasks, four tasks received valid suggestions that have been implemented, such as:

1. Adding a cancel button for the search bar.
2. Swipe to refresh the feature
3. Social media selection options have been made consistent.
4. Addition of time in on-track & overdue report.
5. Navigation titles that reflect the social media platform we are currently working on.

The results from UAT align with the previous study by Kanuri et al. [4], in which time was a significant factor in influencing impression and content effectiveness, as evident in the content developer's request to add time in the on-track and overdue report. It was also found that the hashtag generation process relies on objects provided by MobileNetV2, which has an accuracy of 72% over 1000 objects. It leaves the possibility of improperly generating hashtags due to MobileNetV2's limited image recognition capabilities.

6 Conclusion

6.1 Conclusion

The design and development of a content scheduler and reminder app that automatically generates hashtag recommendations were intended to help plan and develop a content scheduling app. The first step was researching related apps and conducting interviews, which revealed content developers struggle with handling content scheduling, content creation, automating hashtag creation, and un-integration with other systems. The next step was utilizing MobileNetV2's classification capabilities to recognize objects in the image posting. Finally, acceptance testing was done with three kinds of user persona in their respective environment. Through this research, a content scheduling and reminder tool that automatically generates hashtags using MobileNetV2's classification skills were developed to assist content developers in managing content with minimal hassle.

6.2 Future Studies

Advice for future studies and app development is listed as follows:

1. The development of this is expected to be more complex and integrated directly with the API from each social media platform to automatically upload content.
2. The development of this application is expected to use a better *image recognition* model than MobilNetV2 with a level of accuracy surpassing 72% and 1000 objects to automatically generate better hashtag recommendations with the aid of alternative *libraries*.

Reference

[1] S. Costello, "The History of iOS, from Version 1.0 to 16.0," *Lifewire*, Jul. 2022. https://www.lifewire.com/ios-versions-4147730, accessed Aug. 21, 2022.

[2] M. Distaso and T. Mccorkindale, "The Career Path of a Social Media Professional," 2020. [Online], Available: https://instituteforpr.org/wp-content/uploads/Challenges-of-a-Social-Media-Professional-Report-v7.pdf, Accessed: Aug. 15, 2022.

[3] P. Hudson, "How to Combine Core Data and SwiftUI - A Free Hacking with iOS: SwiftUI Edition Tutorial," *Hacking with Swift*, May 12, 2022. https://www.hackingwithswift.com/books/iosswiftui/ how -to-combine-core-data-and-swiftui, accessed Aug. 21, 2022.

[4] V. K. Kanuri, Y. Chen, and S. H. Sridhar, "Scheduling Content on Social Media: Theory, Evidence, and Application," *Journal of Marketing*, 82(6):89-108, Oct. 2018, doi: 10.1177/0022242918805411, 2018.

[5] S. Kemp, "Digital 2022: Indonesia," *DataReportal – Global Digital Insights*, https://datareportal.com/reports/digital-2022-indonesia, accessed Aug. 15, 2022.

[6] N. Kesaris, "New Study Finds Many Social Media Managers Work More Than Their Colleagues and Plan to Leave Their Positions in Less Than Two Years," *Institute for Public Relations*, 2020. https://instituteforpr.org/new-study-finds-many-social-media-managers-work-more-than-their-colleagues-and-plan-to-leave-their-positions-in-less-than-two-years/, accessed Aug. 15, 2022.

[7] C. S. Kwang, "What is Swift Language and Advantages and Disadvantages?," *Towards Dev*, 2022, https://towardsdev.com/what-is-swift-language-and-advantages-and-disadvantages-eb8bb57735cb, accessed Aug. 15, 2022.

[8] S. O. Lagrosen and K. Grundén, "Social Media Marketing in the Wellness Industry," *TQM Journal*, 26(3):253–260, 2014, doi: 10.1108/TQM-12-2013-0129/FULL/XML, 2014.

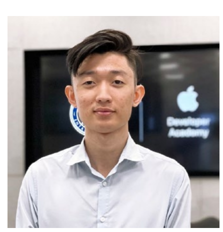
[9] A. Laucuka, "Communicative Functions of Hashtags," *Economics and Culture*, 15(1):56-62, Jun. 2018, doi: 10.2478/JEC-2018-0006, 2018.

[10] G. Meikle, "Social Media: Communication, Sharing and Visibility," *Social Media: Communication, Sharing and Visibility*, pp. 1–170, Feb. 2016, doi:10.4324/9781315884172, 2016.

[11] M. P. Recke and T. G. Pereira, "Core ML Explained: Apple's Machine Learning Framework," Jun. 16, 2021. https://www.createwithswift.com/core-ml-explained-apples-machine-learning-framework/, accessed Aug. 21, 2022.

[12] R. Rokhana, W. Herulambang, and R. Indraswari, "Multi-Class Image Classification Based on MobileNetV2 for Detecting the Proper Use of Face Mask," *International Electronics Symposium 2021: Wireless Technologies and Intelligent Systems for Better Human Lives, IES 2021 - Proceedings*, pp. 636–641, Sept. 2021, doi: 10.1109/IES53407.2021.9594022 2021.

[13] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4510-4520, Jan. 2018, doi: 10.48550/arxiv.1801.04381, 2018.



Bevan Christian is an iOS Developer at Kumparan, a collaborative news media platform. He is an alumnus of the School of Information Technology at Universitas Ciputra Surabaya, as well as an alumnus of Apple Developer Academy. His research interest is in Clean Code, Reactive Programming, and Pattern Architecture. Currently, Bevan has published three applications in Apple's App Store.

Trianggoro Wiradinata is a researcher at the School of Information Technology at Ciputra University. He received a Bachelor's degree in Electrical Engineering (major in Computing) from Petra Christian University, Indonesia, and a Master's degree in Computer Science from University of New South Wales, Australia and completed his Ph.D. program in Information Technology from Assumption University of Thailand. His main research interests are data science, software engineering, technology adoption, and technology-based entrepreneurship. He was actively involved in many large-scale enterprise software development projects before joined as researcher at Ciputra University, Surabaya, Indonesia. Assoc. Prof. Dr. Wiradinata is currently a. member of Association for Computing Machinery (ACM) and Association for Information Systems Indonesia (AISINDO).

Covid-19 Detection Based on Cascade-Correlation Growing Deep Learning Neural Network Algorithm

Soha Abd El-Moamen Mohamed*, Marghany Hassan Mohamed† and Mohammed F. Farghally‡
Assiut University, Assiut, EGYPT

Abstract

COVID-19, is a dangerous disease, that is widely spread among humans by inhalation of the virus, and it harms and may damage the lung. The aim of this paper is to detect COVID-19 using our new algorithm called "Cascade-Correlation Growing Deep Learning Neural Network Algorithm (CCGDLNN)" from Computed tomography (CT) scan images of a patient's chest. We apply the algorithm over 48,260 computed tomography scan images from 377 persons divided into 282 normal persons and 95 patients were infected by COVID-19. Our system is divided into two stages: Firstly, the system removes unclear computed tomography-scan lung images by analyzing them. Secondly, we run our algorithm based on the exception model that begins with a small network without any hidden layers but has input and output layers only. The algorithm after that, adds new neurons and connects them to the last layer or add a new layer with one neuron. Finally, after performing these two stages, the system can be able to detect COVID-19 patients from their lung computed tomography-scan images. We train the data using two different models and compared the results with our model. In the image classification process, our model achieved 98.8% accuracy on more than 7996 test images.

Key Words: Deep learning; constructive deep learning; diagnosis systems; COVID-19; CT scan; automatic medical diagnosis.

1 Introduction

COVID-19 is an infectious disease resulting from the new virus called "severe acute respiratory syndrome Coronavirus 2" (SARS-CoV-2). COVID-19 can be transmitted between persons by contacting directly with a diseased person.

The infection can happen when a patient sneezes, coughs, or even talks close to another person and the infected respiratory droplets are transmitted to the normal person [12, 16, 27].

For Covid-19 diagnosis, scientists used several methods like: Reverse Transcriptase-Polymerase Chain Reaction" (RT-PCR), medical imaging, and medical tests like isothermal nucleic amplification, antibody, and serology [47]. The most popular method to diagnose the viral disease like COVID-19 is the RT-PCR.

However, The RT-PCR requires high experience and many experimentations to develop novel measurements [15]. Also, the shortage of the number of diagnostic tools at some areas around the world forced the scientists and researchers to find an easier way to detect COVID-19. The researchers found that the medical imaging devices like (X-rays and CT scans) are available at many labs and centers, so they used these devises to diagnose COVID-19. COVID-19 virus attacks the patient's lung, so the medical imaging can diagnose the disease from the lung image only. Hence, a chest CT scan has become strong evidence for disease confirmation [26].

Fang et al. noted that the accuracy of CT (98%) was higher than RT-PCR (71%) when diagnosing COVID-19 [14]. Covid-19 appears in the patient's lung using CT scans after at least four days from Covid-19 symptoms [5]. CT scans can help for early COVID-19 diagnosis to prevent the virus to transmit to others, but this method is not recommended at final diagnosis [2]. In [5, 48] some patients who had negative results at their RT-PCR test, were found that their lungs were infected by Covid-19 by using CT scan device. So, when the RT-PCR was repeated twice for these patients, the result converted to positive and confirmed the CT scan examination. However, a manual diagnosis of COVID-19 from chest CT images takes time and it might not be possible to manually check every CT image in emergency situations. So, we need tools that help in diagnosis of COVID-19 automatically by analyzing CT images of patient's lung. Artificial Intelligence (AI) and deep learning can be used to build an AI based tool that accelerates the diagnosis process [39]. The viral infections can be visualized using machine vision and medical imaging. Deep learning is considered as the best method at machine vision [19]. Deep learning has many applications in medicine [31], agriculture [35], economics [18], etc. In [23], the automated method that was used for diagnosing COVID-19 (90%) was better than a manual diagnosis (70%).

All neural networks (NNs) algorithms define the architecture of the network before the beginning of the training process. However, there is another type of NN called constructive (or adaptive) neural network, which allows the structure of the network to be constructed during the training process. The

* Information System Department, Faculty of Computer and Information: Email: sohamoamen@aun.edu.eg.
† Computer Science Department, Faculty of Computer, and Information: Email: marghny@aun.edu.eg.
‡ Information System Department, Faculty of Computer and Information: Email: mfseddik@aun.edu.eg.

constructive algorithm can solve a real-life problem as it is a supervised learning algorithm. The constructive algorithm begins by a small network that might not have any hidden layers but have input and output layers only. The algorithm then adds the neurons and layers gradually during training processes until reaching the optimal result of classification [29].

In this paper as in Figure 1, we used the CCG-DLNN algorithm [29] that we previously developed at [29] for diagnosing COVID-19 cases from the CT scan lung images. This system detects the infected patients using CT scans of these patients.

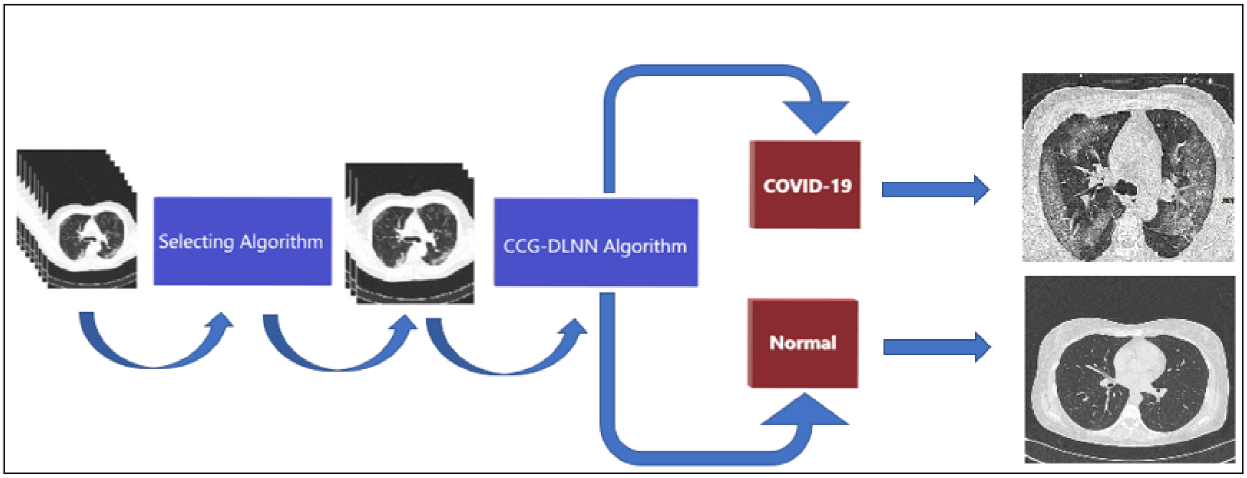


Figure 1: The steps of our proposed system for COVID-19 CT scan chest images classification

The dataset that had been used was COVIDCTset [36], and it was divided into 95 patients (15,589 COVID-19 images) and 282 normal persons (48,260 normal images). Our work has two stages:

1. A selecting algorithm: selecting the most noticeable infected lungs.
2. Run the CCG-DLNN algorithm.

After these two steps, we trained and tested two constructive algorithms other than CCG-DLNN algorithm and compared the three algorithms. Finally, we evaluated CCG-DLNN algorithm on more than 7996 images by single-image classification. Then, we examined the system by testing it on 245 patients and 41,892 images. The COVID-19 infected areas are explored in resulting images using a feature visualization algorithm.

The paper organization is: In Section 2, we will describe the related works. In Section 3, the dataset and the proposed algorithm will be discussed. In Section 4 we will present the experimental results. And finally in section 5, the conclusion of the paper was discussed.

2 Related works

Until the time of writing this paper, many trials have been done and several research studies published to deal with the new pandemic virus COVID-19. There are two types of studies that diagnosed COVID-19 which are: binary classifications or multiple classifications. These studies used chest medical imaging by using X-rays or by CT-scan. This type of studies sometimes used raw data and others used feature extraction.

Almost all studies used convolutional neural network (CNN) and the amount of used data was different from one to another. [30, 45] it was determined COVID-19 was diagnosed using the current deep learning networks and X-ry images of chest. They have a high accuracy on their results. However, [46] developed a new deep learning (DL) system and evaluated it to make a three-class data classification. This study used 5372 cases from many hospitals around China and used their CT scan chest images for the proposed DL. In [20] the X-ray images and VGG19 and DenseNet models also detected COVID-19.

In [23, 34] the Xception architecture and X-ray images of the chest were used to diagnose COVID-19. In [24] the cases were classified into 3 classes: normal, bacterial, and viral pneumonia. In [34] Resnet50v2 networks along with Xception architecture were used to classify 11,302 images into 3 classes: normal, pneumonia, and COVID-19 patients. The accuracy was 99.5 for this classification.

In [25] developed a new network called COVNet. COVNet was evaluated by using 3322 from 3506 chosen CT scan images. In [7] proposed an eXplainable deep learning approach and used 2482 CT scan images of 120 people to classify the COVID-19 infected patients, and the normal persons. The accuracy reached to 97.38% at this study. In [23] a new algorithm was developed called CovidCTNet. They used CT scan chest images to classify 287 patients into three classes: normal people, COVID-19 patients, and community-acquired pneumonia (CAP). The accuracy of the results was 90%. A segmentation of COVID-19 infections has been performed by using CT scans at [44].

The CNN and machine learning techniques have been used by [1, 8, 10, 28, 31-32, 37, 49] using CT scan or X ray chest images. In [36] it achieved 0.9849 accuracy of CNN when they developed a new method by modifying the feature selection pyramid network and used the ResNet50V2 network.

An evolutionary neural network [9] is used to detect COVID-19 automatically using a common pneumonia and pneumonia that causes COVID-19. They used transfer learning to detect different abnormalities with small medical image datasets. Multi-objective differential evolution (MODE) [42] is based on CNN for classifying CT scan chest images to diagnose COVID-19. New deep transfer learning model that was based on DenseNet201 by using CT scan chest images as classification of the COVID-19 patient [22]. A new multi-class segmentation technique called Residual Attention U-Net was proposed in [11]. This new technique can be used to diagnose COVID-19 and its related pneumonia using CT scan chest images. In [3] detected infection areas and the diseased part by using a new proposed network "Auto Diagnostic Medical Analysis". They used X-ray and CT scan chest images. DenseNet network has been used for removing the infected spots at the lung. In [6] two methods: the first method is based on AOCTNet, MobileNet and ShuffleNet CNNs; the second method is based on removing the features at X-ray images and make a classification using many different algorithms to diagnose COVID-19. In [17] COVID-19 was detected by using Bayesian CNN model based on the dropweights and chest X-ray images.

The SqueezeNet model is based on Bayesian optimization and diagnosed COVID-19 using X-ray images in [43]. Five models (VGG16, VGG19, ResNet, DenseNet and InceptionV3) diagnosed COVID-19 by using X-ray images [38] and in [33] used five methods to extract features. Then, they classified features using SVM and two, five, and ten-fold cross-validation methods. COVID-19 was diagnosed by using three models (ResNet, InceptionV3, and Inception-ResNet) [10] and they worked on chest X-ray images. In [4] they used small dataset by developing a new deep neural network (DNN) that was based on diagnostic solutions and Capsule Networks.

All these previous researches diagnosed COVID-19 or classified medical images into two or more classes using CNN models.

CNN is based on a fixed network that has a fixed number of layers and neurons. So, all these researches must choose the most suitable network (number of layers, and number of neurons) manually.

In this paper, the structure of the DL network is determined dynamically based on the number of images and the type of them at each problem. We performed CCG-DLNN algorithm that we previously proposed at [29] to diagnose COVID-19 on CT scan chest images. The following section will describe the proposed system in detail.

3 The Proposed System

3.1 The Dataset

We used COVID-CTset1 [36] dataset in our system. This data was collected by Negin radiology in Sari, Iran. The COVID-CTset1 dataset was collected from March 5th to April 23rd, 2020. These CT images were captured and visualized using a scope model called SOMATOM and software called SYNGO CT VC30-easyIQ. The captured images were at format 16-bit grayscale DICOM, and resolution 512*512 pixels. It converted the format of the resulted images from DICOM to TIFF to remove the patient's private information that attached to each image [18]. COVID-CTset1 dataset has 63,849 images for 377 patients. These images were divided into 15,589 COVID-19 images for 95 COVID-19 patients, and 48,260 normal images for 282 normal people in Tables 1 and 2.

3.2 CT Scans Selection Algorithm

In this paper, we used a selection algorithm that was performed in [18]. The idea of this selection algorithm is to discard the CT scan images of the closed lung. The lung CT scan produces a sequence of images (consecutive frames). At the beginning and the end of consecutive frames of the CT scan lung images, the lung is closed (Figure 2). In these closed lung images, the COVID-19 infection part does not appear. These types of images are useless at classification process. Figure 3 shows the steps of a selection algorithm as discussed in [36].

1. Extracting a region from the middle of each CT scan lung image. This region has pixels numbers [x;y] = [120;240] :[370;340] as in Figure 4.
2. Selecting the dark pixels from the extracting region in step 1 which has values < 300.
3. Return the maximum (Max) and minimum (Min) number of dark pixels of all images in the CT scan lung sequence.
4. Calculating the threshold in equation (1).

$$\text{Threshold} = \frac{\text{Max(dark pixels)}-\text{Min(dark pixels)}}{1.5} \quad (1)$$

5. Comparing the number of dark pixels of each image with the amount of threshold that is calculated in step 4.
6. If the number of dark pixels is less than the threshold, then remove this image from the sequence (closed lung).
7. Finally, we will have a number of images that appear inside of the lung.

Figure 5 shows the deleting images that have a closed lung and the remaining images of the sequence that will be used in the classification process that will be discussed in detail at the next subsection.

3.3 CCG-DLNN Algorithm

Constructive algorithm is used for generating an acceptable network in an automatic manner. In constructive algorithm, the neural network is built gradually during the training process. The network starts with a minimum number of layers (may be input and output layers only). The hidden layers, nodes and connections are added gradually at each step of the training process, until reaching the most suitable model for solving an appropriate problem [40].

Table 1: Number of COVID-19 and normal patients

COVID-19 Patients	Normal People	Total
95	282	377

Table 2: Number of COVID-19 and normal images

COVID-19 Images	Normal Images	Total
15,589	48,260	63,849

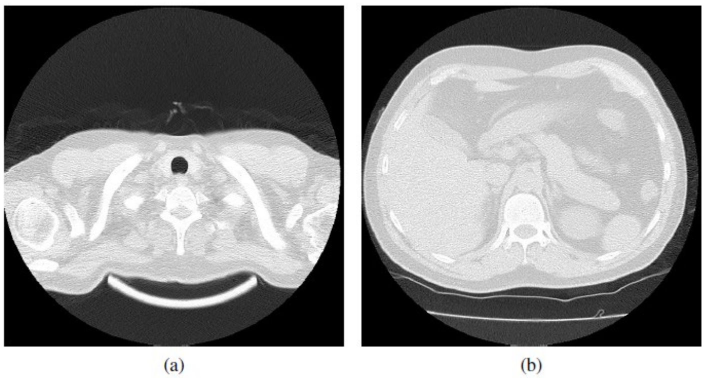


Figure 2: Closed lung at the beginning of CT scan lung sequence (a) and at the end of CT scan lung sequence (b)

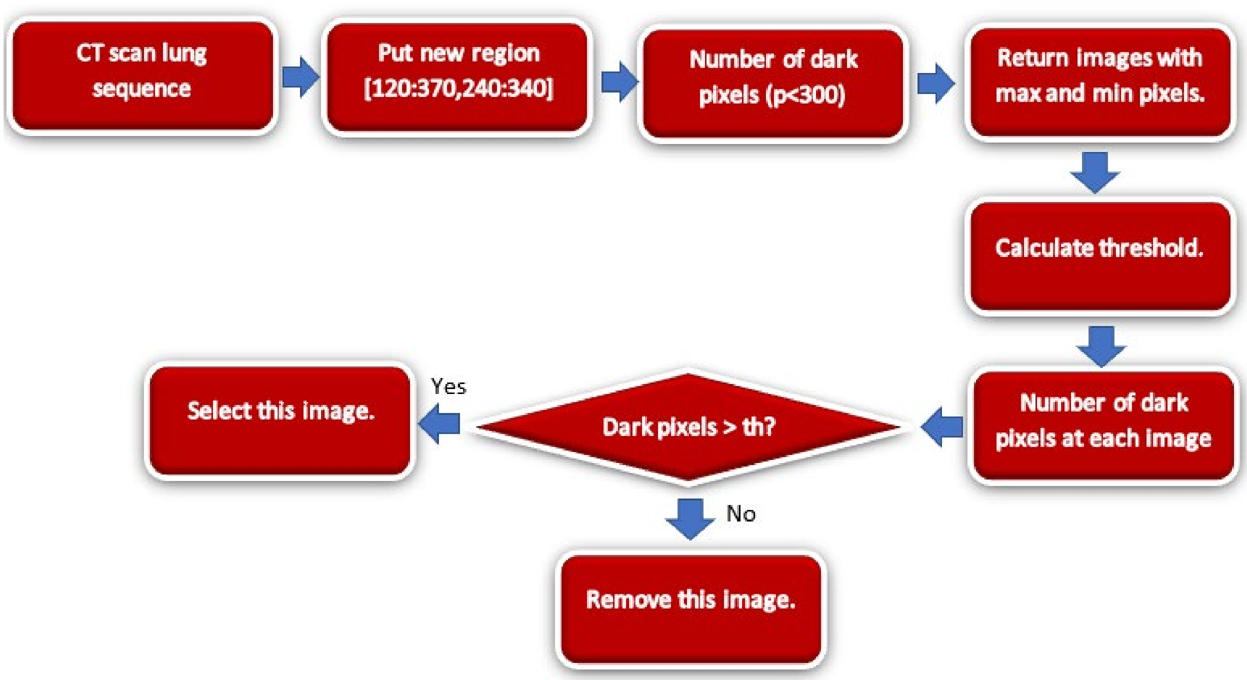
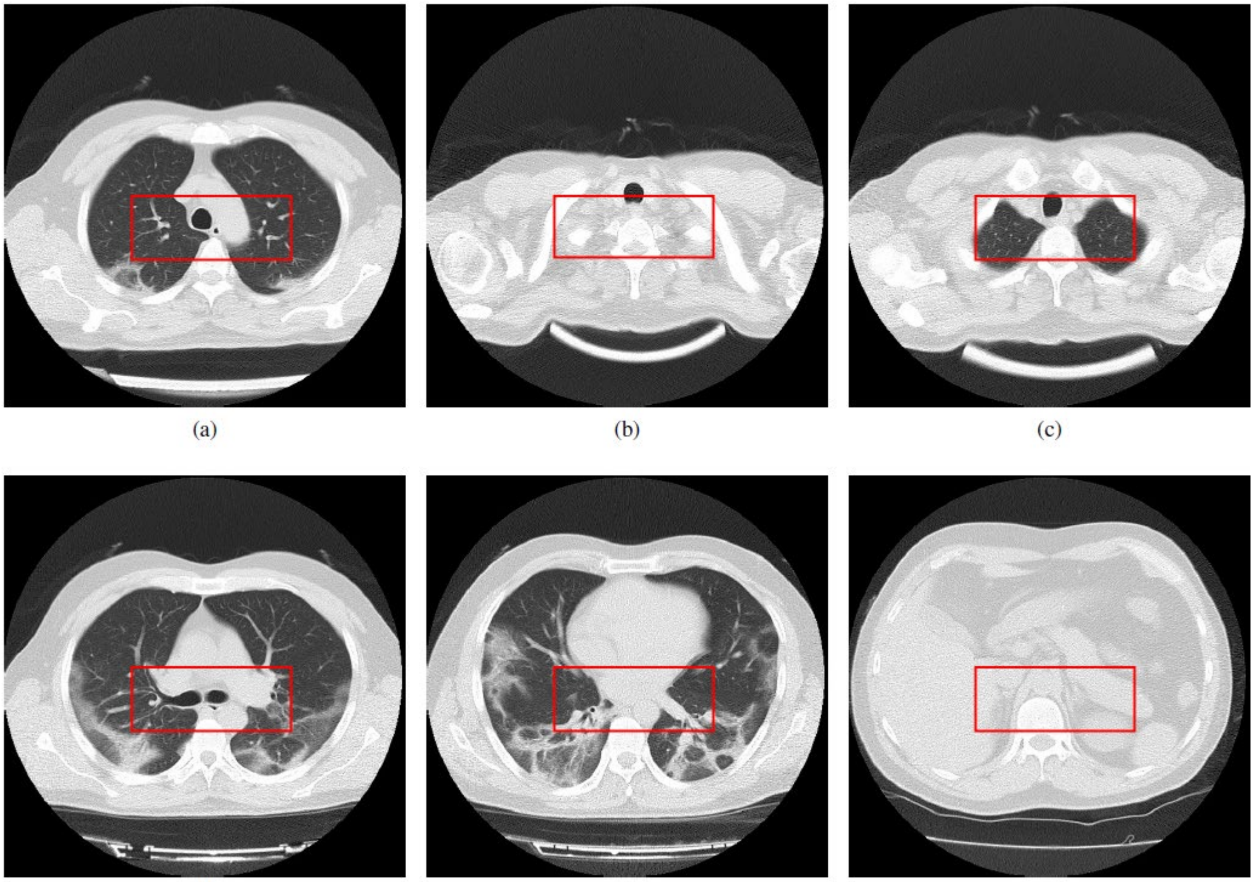


Figure 3: The steps of the selection algorithm

‘



(a) (b) (c)

(d) (e) (f)

Figure 4: Extracting a specific region from the CT scan lung images

Figure 5: CT scan lung sequence of a patient after deleting closed lung images

The classification problems can be solved by the constructive neural network. The constructive neural network has many benefits such as [41]:

1. It requires few numbers of computations.
2. It requires small topology.
3. It learns faster.
4. It performs the classification process with high accuracy.

In this paper, we used the CCG-DLNN algorithm that we previously proposed at [29] to classify the COVID-19 CT scan lung images that resulted from the selection system. The CCG-DLNN algorithm is a constructive algorithm that begins with only input and output layers. Then the algorithm adds new neurons or new hidden layers gradually during the training process [29]. The CCG-DLNN algorithm uses the same technique as Cascade Correlation Neural Network (CCNN). The difference between CCG-DLNN algorithm and CCNN is that our algorithm adds more than one node to each new added hidden layer.

The steps of CCG-DLNN algorithm are:

1. Beginning with a simple neural network with input and output layers only.
2. Training the network and calculating the loss function $L(\phi_t)$.
3. If $L(\phi_t)$>DE, then DE is the desired error. So, we need to do one of the following two options:

   (a) Adding a new neuron to the last hidden layer.
   (b) Adding a new hidden layer with one neuron.

5. Repeat steps from 2 to 4 until $L(\phi_t)$ < DE.

3.4 CT scan COVID-19 Classification System

In Figure 6, we performed the CT scan COVID-19 classification system using the CCG-DLNN algorithm and Xception Architecture [13].

The Xception architecture is a Convolutional Neural Network (CNN) architecture. This architecture is based on separating convolution layers. The Xception architecture extract features the convolutional layers. This convolutional layer consists of 36 layers sorted by 14 modules which connects linearly except the first and last modules. The Xception architecture is a depth wise linear stack based on separate convolutional layers.

The resulting images from Xception model is the input layer of the CCG-DLNN algorithm.

4 Experimental Results

In this section we will run our proposed classification system and compare the results with [36] model and Xception model on the same dataset COVID-CTset.

We performed all models by Python 3.7, 253 Core i7 with CPU 3.60GHZ,NVIDIA GeForce RTX 2070 GPU and RAM size 32.0 GB. We ran the deep network by Keras library and Tensorflow backend. We used CT scan images dataset that were introduced by [36].

We converted all images to TIFF format and with 32-bit float to visualize them easily. We used the same characteristics as [36] for a fair comparison.

We divided the dataset into 5 folds and divided each fold into 3 sets which were training, validation, and testing sets [36]. The number of images at each fold and for the training and test sets shown in Table 3 (this distribution made by [36]).

We used the same parameters used in [36] to train the dataset. We trained the dataset using the CCG-DLNN algorithm based on Xception model, [36] model, and Xception model only (Table 4).

We evaluated the CCG-DLNN algorithm based on Xception, [36], and Xception models using the accuracy metric during the training process. Then we divided the images at the CT scan lung dataset into four parts:

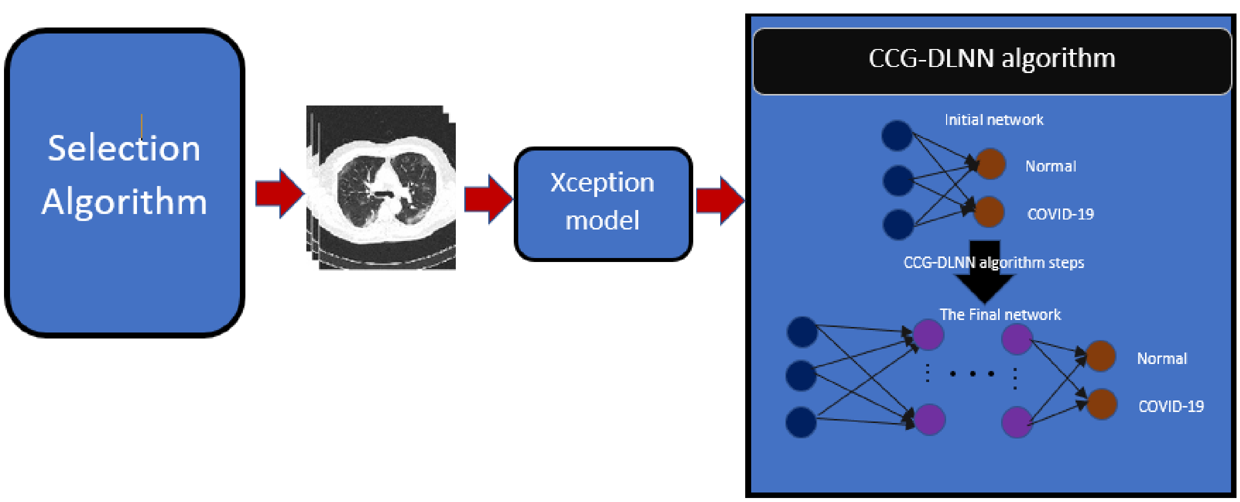


Figure 6: CT scan COVID-19 classification system

1. True Positive (TP): The correct classified images.
2. False Positive (FP): The wrong classified images.
3. False Negative (FN): The images with a wrong class label and are not classified to this wrong class.
4. True Negative (TN): The images with a wrong class label and are classified to this wrong class.

So, we can calculate the accuracy of equation (2), specificity in equation (3), sensitivity in equation (4) and precision in equation (5) using the following equations:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

$$\text{Specificity} = \frac{TN}{TN+FP} \quad (3)$$

$$\text{Sensitivity} = \frac{TP}{TP+FN} \quad (4)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (5)$$

The results of the accuracy for each model and each fold are shown in Table 5. Table 6 shows the evaluating results for the 5 folds and the three models. Figure 7 shows the training accuracy and loss in 50 epochs for the three models. The speed of each model is in Table 7 which shows the training time of each model.

5 Conclusion and Future work

In this paper, we used CCG-DLNN algorithm that we previously developed at [29] for diagnosis COVID-19 cases from the CT scan lung images. We used the Xception model as the backbone of our system. The dataset that we used was COVID-CTset1 dataset that was developed by [36]. This dataset contains 63,849 images divided into 15,589 COVID-19 images for 95 COVID-19 patients, and 48,260 normal images. Our system firstly, performed a selection algorithm that [36] developed. This selection algorithm selects the open lung images from the CT scan lung image sequences, and it removes unclear images that have a closing lung and hard to detect COVID 19 from them. The selection algorithm speeds up the training processes at the next phase. The second phase of our system is to make a classification process by CCG-DLNN algorithm based on Xception model. CCG-DLNN algorithm is a constructive algorithm that begins with only an input and output layers. Then the algorithm adds new neurons or new hidden layers gradually during the training process. It uses the same technique as Cascade Correlation Neural Network (CCNN). The difference between CCG-DLNN algorithm and CCNN is that our algorithm adds more than one node to each new added hidden layer. When we compared our proposed system with [36] and Xception model using the same dataset, the accuracy of our system was the highest at 5 folds with 99.2%, 98.6%, 98.2%, 98.8%, and 99.6% overall accuracy.

Table 3: Number of COVID-19 and normal images in training and testing sets [36]

	Training set		Testing set	
	COVID-19	Normal	COVID-19	Normal
Fold1	1820	1916	462	7860
Fold2	1817	1898	465	7878
Fold3	1836	1893	446	7883
Fold4	1823	1920	459	7856
Fold5	1832	1921	450	7785

Table 4: The Training parameters

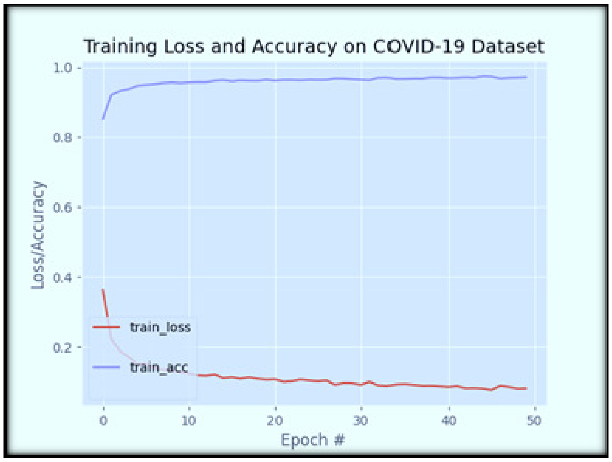
Training Parameters	CCG-DLNN + Xception model	(96) model	Xception model
Learning Rate	1e-4	1e-4	1e-4
Batch size	14	14	14
Optimizer	Nadam	Nadam	Nadam
Loss function	Categorical Cross Entopy	Categorical Cross Entopy	Categorical Cross Entopy
Epochs	50	50	50
Horizontal/vertical flipping	Yes	Yes	Yes
Zoom range	5%	5%	5%
Rotation range	0-360 degree	0-360 degree	0-360 degree
Width/height shifting	5%	5%	5%
Shift range	5%	5%	5%

Table 5: The resulted accuracy for the three models

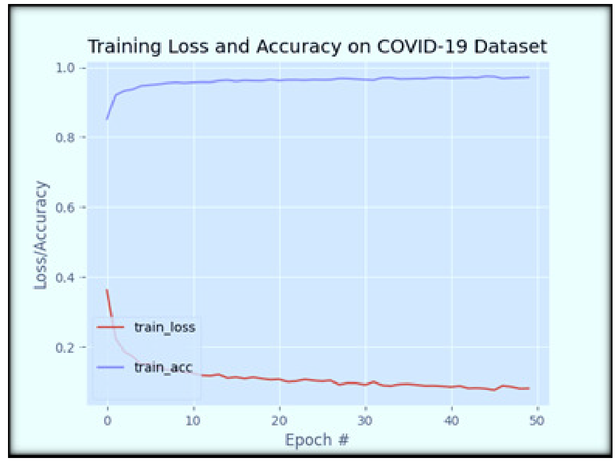
Folds	Network	Overall accuracy	COVID19 accuracy	Normal accuracy
1	CCG-DLNN+Xception	0.992	0.992	0.992
	(96) model	0.987	0.987	0.987
	Xception	0.9811	0.9811	0.9811
2	CCG-DLNN+Xception	0.986	0.986	0.986
	(96) model	0.9847	0.9847	0.9847
	Xception	0.9494	0.9494	0.9494
3	CCG-DLNN+Xception	0.982	0.982	0.982
	(96) model	0.9777	0.9777	0.9777
	Xception	0.9741	0.9741	0.9741
4	CCG-DLNN+Xception	0.988	0.988	0.988
	(96) model	0.9868	0.9868	0.9868
	Xception	0.9446	0.9446	0.9446
5	CCG-DLNN+Xception	0.996	0.996	0.996
	(96) model	0.9886	0.9886	0.9886
	Xception	0.9785	0.9785	0.9785

Table 6: The evaluation results of the three models

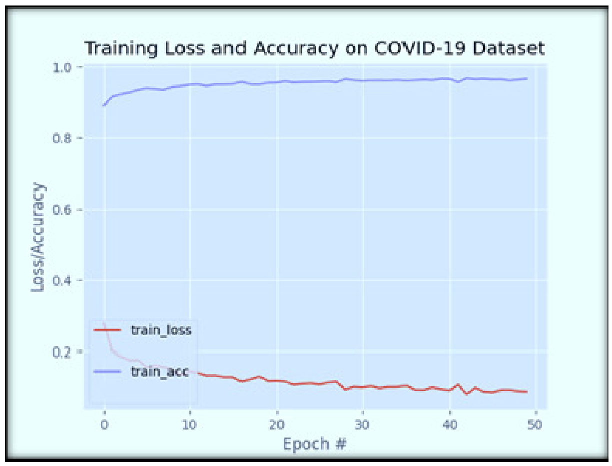
Fold	Network	COVID19 sensitivity	Normal sensitivity	COVID19 specificity	Normal specificity	COVID19 precision	Normal precision
1	CCG-DLNN and Xception	0.98	0.97	0.97	0.98	0.92	0.989
	(96) model	0.9437	0.9896	0.9896	0.9437	0.8417	0.9967
	Xception	0.987	0.9808	0.9808	0.987	0.7512	0.9992
2	CCG-DLNN and Xception	0.98	0.992	0.992	0.98	0.752	0.992
	(96) model	0.9527	0.9865	0.9865	0.9527	0.8069	0.9972
	Xception	0.9849	0.9473	0.9473	0.9849	0.5246	0.9991
3	CCG-DLNN and Xception	0.99	0.997	0.997	0.99	0.86	0.9995
	(96) model	0.9574	0.9788	0.9788	0.9574	0.7189	0.9975
	Xception	0.9731	0.9741	0.9741	0.9731	0.6803	0.9984
4	CCG-DLNN and Xception	0.989	0.989	0.989	0.989	0.92	0.9995
	(96) model	0.963	0.9882	0.9882	0.963	0.8262	0.9978
	Xception	0.9782	0.9426	0.9426	0.9782	0.4989	0.9987
5	CCG-DLNN and Xception	0.985	0.981	0.981	0.985	0.893	0.9996
	(96) model	0.9311	0.9919	0.9919	0.9311	0.8693	0.996
	Xception	0.9778	0.9785	0.9785	0.9778	0.7249	0.9987



(a) Xception Model



(b) [18] model



(c) CCG-DLNN + Xception model

Figure 7: The training loss and accuracy in 50 epochs for the three models (a) Xception model (b) [36] model (c) CCG-DLNN +Xception model

Table 7: The training time in minutes for the three models on NVIDIA GeForce RTX 2070 GPU

Model	Training Time of each Epoch (m)
CCG-DLNN+Xception model	5 m
(96) model	3 m
Xception model	4 m

From all the previous results, we can observe that our system improved the detection of COVID 19 from CT scan lung images.

References

[1] A. A. Abdulmunem, Z. A. Abutiheen, and H. J. Aleqabie, "Recognition of Corona Virus Disease (COVID-19) using Deep Learning Network," *International Journal of Electrical and Computer Engineering (IJECE)*, 11(1):365–374, 2021.

[2] ACR, "ACR Recommendations for the Use of Chest Radiography and Computed Tomography (CT) for Suspected COVID-19 Infection — American College of Radiology," 2020 05, https: //www.acr.org/Advocacy-and-Economics/ACR-Position-Statements/Recommendations-for-Chest-Radiographyand-CT-for-Suspected-COVID19-Infection, Online: accessed 23.06.21, 2020.

[3] N. C. D. Adhikari, "Infection Severity Detection of

COVID-19 from X-rays and CT Scans using Artificial Intelligence," *International Journal of Computer (IJC)*, 38(1):73-92, 2020.

[4] P. Afshar, S. Heidarian, F. Naderkhani, A. Oikonomou, K. N. Plataniotis, and A. Mohammadi, "Covid-caps: A Capsule Network-based Framework for Identification of COVID-19 Cases from X-ray Images," Pattern Recognition Letters, 138:638-643, 2020.

[5] T. Ai, Z. Yang, H. Hou, C. Zhan, C. Chen, W. Lv, Q. Tao, Z. Sun, and L. Xia, "Correlation of Chest CT and RT-PCR Testing for Coronavirus Disease 2019 (COVID-19) in China: A Report of 1014 Cases," *Radiology*, 296(2):E32–E40, 2020.

[6] A. M. Alqudah, S. Qazan, and A. Alqudah, "Automated Systems for Detection of COVID-19 using Chest X-ray Images and Lightweight Convolutional Neural Networks," Research Square, 1:1-26, 2020.

[7] P. Angelov and E. Almeida Soares, "SARS-CoV-2 CT-Scan Dataset: A Large Dataset of Real Patients CT Scans for SARS-CoV-2 Identification," medRxiv, 2020.

[8] I. D. Apostolopoulos, S. I. Aznaouridis, and M. A. Tzani, "Extracting Possibly Representative COVID -19 Biomarkers from X-ray Images with Deep Learning Approach and Image Data Related to Pulmonary Diseases," *Journal of Medical and Biological Engineering*, 40:462–469, 2020.

[9] I. D. Apostolopoulos and T. A. Mpesiana, "Covid-19: Automatic Detection from X-ray Images Utilizing Transfer Learning with Convolutional Neural Networks," Physical and Engineering Sciences in Medicine, 43(2):635-640, 2020.

[10] M. Barstugan, U. Özkaya, and S. Öztürk, "Coronavirus (COVID-19) Classification using CT Images by Machine Learning Methods," arXiv preprint arXiv:2003.09424, 2020.

[11] X. Chen, L. Yao, and Y. Zhang, "Residual Attention U-net for Automated Multi-class Segmentation of COVID-19 Chest CT Images," arXiv preprint arXiv:2004.05645, 2020.

[12] N. Chen, M. Zhou, X. Dong, J. Qu, F. Gong, Y. Han, Y. Qiu, J. Wang, Y. Liu, Y. Wei, et al., "Epidemiological and Clinical Characteristics of 99 Cases of 2019 Novel Coronavirus Pneumonia in Wuhan, China: A Descriptive Study," *The Lancet*, 395(10223):507-513, 2020.

[13] F. Chollet, "Xception: Deep Learning with Depth Wise Separable Convolutions," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1251-1258, 2017.

[14] Y. Fang, H. Zhang, J. Xie, M. Lin, L. Ying, P. Pang, and W. Ji, "Sensitivity of Chest CT for COVID-19: Comparison to RT-PCR," *Radiology*, 296(2):E115–E117, 2020

[15] Genetic Education, "Reverse Transcription PCR: Principle, Procedure, Application, Advantages and Disadvantages," 2020, https://geneticeducation.co.in/reverse-transcription-pcr-principle-procedure applica tions-advantages-and-disadvantages/#Disadvantages, Online; accessed 23.06.21.

[16] I. Ghinai, T. D. McPherson, J. C. Hunter, H. L. Kirking, D. Christiansen, K. Joshi, R. Rubin, S. Morales-Estrada, S. R. Black, M. Pacilli, et al., "First Known Person-to-Person Transmission of Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-COV-2) in the USA," *The Lancet*, 395(10230):1137-1144, 2020.

[17] B. Ghoshal and A. Tucker, "Estimating Uncertainty and Interpretability in Deep Learning for Coronavirus (COVID-19) Detection," arXiv preprint arXiv:2003.10769, 2020.

[18] G. P. Green, J. C. Bean, and D. J. Peterson, "Deep Learning in Intermediate Microeconomics: Using Scaffolding Assignments to Teach Theory and Promote Transfer," *The Journal of Economic Education*, 44(2):142-157, 2013.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778, 2016.

[20] E. E.-D. Hemdan, M. A. Shouman, and M. E. Karar, "COVIDX-NET: A Framework of Deep Learning Classifiers to Diagnose COVID-19 in X-ray Images," arXiv preprint arXiv:2003.11055, 2020.

[21] A. Huemer, D. Elizondo, and M. Gongora, "A Constructive Neural Network for Evolving a Machine Controller in Real Time," Constructive Neural Networks, Springer, pp. 225-242, 2009.

[22] A. Jaiswal, N. Gianchandani, D. Singh, V. Kumar, and M. Kaur, "Classification of the COVID-19 Infected Patients using DenseNet201 based Deep Transfer Learning," *Journal of Biomolecular Structure and Dynamics*, 39:1-8, 2020.

[23] T. Javaheri, M. Homayounfar, Z. Amoozgar, R. Reiazi, F. Homayounieh, E. Abbas, A. Laali, A. R. Radmard, M. H. Gharib, S. A. J. Mousavi, et al., "COVIDCTNET: An Open-Source Deep Learning Approach to Identify COVID-19 using CT Image," arXiv Preprint arXiv:2005.03059, 2020.

[24] A. I. Khan, J. L. Shah, and M. M. Bhat, "Coronet: A Deep Neural Network for Detection and Diagnosis of COVID-19 from Chest X-ray Images," *Computer Methods and Programs in Biomedicine*, 196:105581, 2020.

[25] L. Li, L. Qin, Z. Xu, Y. Yin, X. Wang, B. Kong, J. Bai, Y. Lu, Z. Fang, Q. Song, et al., "Artificial Intelligence Distinguishes COVID-19 from Community Acquired Pneumonia on Chest CT," Radiology, 2020.

[26] X. Li, W. Zeng, X. Li, H. Chen, L. Shi, X. Li, H. Xiang,Y. Cao, H. Chen, C. Liu, et al., "Ct Imaging Changes of Corona Virus Disease 2019 (Covid-19): A Multi-Center Study in Southwest China," *Journal of Translational Medicine*, 18:1-8, 2020.

[27] J. Liu, X. Liao, S. Qian, J. Yuan, F. Wang, Y. Liu, Z.

Wang, F.-S. Wang, L. Liu, and Z. Zhang, "Community Transmission of Severe Acute Respiratory Syndrome Coronavirus 2, Shenzhen, China, 2020," *Emerging Infectious Diseases*, 26(6):1320, 2020.

[28] Krishore Medhi, Md Jamil and Md Iftekhar. Hussain, "Automatic Detection of COVID-19 Infection from Chest X-ray using Deep Learning," medRxiv, 2020.

[29] S. A. E.-M. Mohamed, M. H. Mohamed, and M. F. Farghally, "A New Cascade-Correlation Growing Deep Learning Neural Network Algorithm," *Algorithms*, 14(5):158, 2021.

[30] A. Narin, C. Kaya, and Z. Pamuk, "Automatic Detection of Coronavirus Disease (COVID-19) using X-ray Images and Deep Convolutional Neural Networks," Pattern Analysis and Applications, pp. 1-14, 2021.

[31] U. Özkaya, S¸ Öztürk, and M. Barstuğan, *Coronavirus (COVID-19) Classification using Deep Features Fusion and Ranking Technique," Big Data Analytics and Artificial Intelligence Against COVID-19: Innovation Vision and Approach*, Springer, pp. 281-295, 2020.

[32] S. Öztürk, U. Özkaya, and M. Barstuğan, "Classification of Coronavirus (COVID-19) from X-ray and CT Images using Shrunken Features," *International Journal of Imaging Systems and Technology*, 31:5-15, 2020.

[33] N. S. Punn and S. Agarwal, "Automated Diagnosis of COVID-19 with Limited Posteroanterior Chest X-ray Images using Fine-tuned Deep Neural Networks," Applied Intelligence, 51(5):2689-2702, 2021.

[34] M. Rahimzadeh and A. Attar, "A Modified Deep Convolutional Neural Network for Detecting COVID-19 and Pneumonia from Chest X-ray Images Based on the Concatenation of Xception and Resnet50v2," *Informatics in Medicine Unlocked*, 19:100360, 2020.

[35] M. Rahimzadeh and A. Attar, "Detecting and Counting Pistachios Based on Deep Learning," *Iran Journal of Computer Science*, 5:1-13, 2021.

[36] M. Rahimzadeh, A. Attar, and S. M. Sakhaei, "A Fully Automated Deep Learning-Based Network for Detecting COVID-19 from a New and Large Lung CT Scan Dataset," *Biomedical Signal Processing and Control*, 68:102588, 2021.

[37] P. Saha, M. S. Sadi, and M. M. Islam, "Emcnet: Automated COVID-19 Diagnosis from X-ray Images using Convolutional Neural Network and Ensemble of Machine Learning Classifiers," *Informatics in Medicine Unlocked*, 22:100505, 2021.

[38] K. Sahinbas and F. O. Catak, "Transfer Learning-based Convolutional Neural Network for COVID-19 Detection with X-ray Images," Data Science for COVID-19, Elsevier, pp. 451-466, 2021.

[39] K. Santosh, "Ai-Driven Tools for Coronavirus Outbreak: Need of Active Learning and Cross-Population Train/Test Models on Multitudinal/Multimodal Data," *Journal of Medical Systems*, 44(5):1-5, 2020.

[40] S. K. Sharma and P. Chandra, "Constructive Neural Networks: A Review," *International Journal of Engineering Science and Technology*, 2(12):7847-7855, 2010.

[41] D. Shen, G. Wu, and H.-I. Suk, "Deep Learning in Medical Image Analysis," Annual Review of Biomedical Engineering, 19:221-248, 2017.

[42] Dilbag Singh, Vijay Kumar and Manjit Kaur, "Classification of COVID-19 Patients from Chest CT Images using Multi Objective Differential Evolution-based Convolutional Neural Networks," *European Journal of Clinical Microbiology & Infectious Diseases*, 39(7):1379-1389, 2020.

[43] F. Ucar and D. Korkmaz, "Covidiagnosis-net: Deep Bayes Squeeze Net based Diagnosis of the Coronavirus Disease 2019 (COVID-19) from X-ray Images," Medical Hypotheses, 140:109761, 2020.

[44] A. Voulodimos, E. Protopapadakis, I. Katsamenis, A. Doulamis, and N. Doulamis, "Deep Learning Models for COVID-19 Infected Area Segmentation in CT Images," medRxiv, 2020.

[45] L. Wang, Z. Q. Lin, and A. Wong, "COVID-NET: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-ray Images," *Scientific Reports*, 10(1):1-12, 2020.

[46] S. Wang, Y. Zha, W. Li, Q. Wu, X. Li, M. Niu, M. Wang, X. Qiu, H. Li, H. Yu, et al., "A Fully Automatic Deep Learning System for COVID-19 Diagnostic and Prognostic Analysis," *European Respiratory Journal*, 56(2):1-11, 2020.

[47] Wikipedia, COVID-19 Testing – Wikipedia, https://en.wikipedia.org/wiki/COVID-19 testing, 2020,

[48] X. Xie, Z. Zhong,W. Zhao, C. Zheng, F.Wang, and J. Liu, "Chest CT for Typical Coronavirus Disease 2019 (COVID-19) Pneumonia: Relationship to Negative RT-PCR Testing," *Radiology*, 296(2):E41–E45, 2020.

[49] J. Zhang, Y. Xie, Y. Li, C. Shen, and Y. Xia, "COVID-19 Screening on Chest X-ray Images using Deep Learning based Anomaly Detection," arXiv preprint arXiv:2003.12338, 2020.

Soha Abd El-Moamen Mohamed has completed her Bachelor of Faculty of Computer and Information from Assiut University, Assiut, Egypt and Master of Information System from Assiut University, Assiut, Egypt. Her PhD is from Faculty of Computer and Information, Assiut University in the field of constructive deep neural network. Currently, she is Lecturer in the Department of Information System at Faculty of Computer and Information from Assiut University, Assiut, Egypt. Her research area includes big data, machine learning, and advanced database.

Marghany Hassan Mohamed has completed his Bachelor of Mathematics, Faculty of Science from Assiut University, Assiut, Egypt and Master of Computer Science from Assiut University, Assiut, Egypt. His PhD is in Computer Science from Kyushu University, Japan. Currently, he is a Professor in the Department of Computer Science at Faculty of Computer and Information from Assiut University, Assiut, Egypt. His research area includes data mining, web mining, machine learning, and security.

Mohammed F. Farghally has completed his Bachelor of Faculty of Computer and Information from Assiut University, Assiut, Egypt and Master of Information Systems, December 2009, Faculty of Computers and Information, Cairo university.

His PhD at Computer Science and Applications, College of Engineering, Virginia Tech, VA, USA. Currently, he is Lecturer at Faculty of Computer and Information at Assiut University, Assiut, Egypt. His research area digital education, algorithm visualization, interactive learning environments, pedagogical evaluation, and educational data mining.

A Comparative Study of Classification Algorithms of Moodle Course Logfile using Weka Tool

Iman Al-Kindi* and Zuhoor Al-Khanjari
Sultan Qaboos University, Muscat, SULTANATE OF OMAN

Abstract

Learning Management Systems (LMSs) have been widely used in the deployment of e-learning in higher education institutions. One of the most famous LMS used is Moodle. In Moodle environment, classification has been used for several reasons, including finding students who share similar traits and forecasting student performance. Therefore, this study looks at two classification algorithms that were used on a dataset gathered from a Moodle LMS course logfile. The goal is to conduct a thorough theoretical and experimental examination of classification data mining techniques, as well as a comparison study, to determine which methodology is the best for identifying student performance with the support of their engagement, behavior, and personality during different activities of the course. The algorithms under investigation are Naive Bayes (NB) and Random Forest (RF). The performance of the classification of the two algorithms is compared using the tool Weka "Waikato Environment for Knowledge Analysis" as an open-source software package that includes data preparation, algorithm implementation and visualization tools. According to the study of the comparison results, the classification algorithms with the best accuracy is the Random Forest, with 97.36 % correctly predicted instances. In a Moodle environment, the classification techniques might be used to predict students' performance.

Key Words: Moodle, logfile, classification algorithms, student performance, student engagement.

1 Introduction

Recent developments in education technology are all driving global education reform [3]. The construction of a smart learning environment is also the foundation for altering education and learning techniques. Higher education institutions require personalized and smart learning environments with learning materials to meet the needs of their students, who have a wide range of demands [2, 5]. Learning Management Systems (LMSs) are also becoming increasingly common in universities, schools, and businesses, with individual professors using them to supplement traditional face-to-face sessions with online technology [27]. Modular Object-Oriented Dynamic Learning Environment "Moodle" is an open-source and one of the most popular Learning Management Systems (LMSs) [16]. There is a lot of interest these days in evaluating and mining Moodle interaction data for forecasting students' final grades in blended learning [29]. Moodle log data, particularly that relating to students' interactions with educational materials, may be quite interesting and useful for developing student behavior models [28]. Educational Data Mining (EDM) is a valuable technique for analyzing this data [326]. It is the process of discovering information from LMS datasets [26]. There are several data mining tools available. DBMiner and SPSS are instances of commercial mining tools, whereas Weka and Keel are examples of public domain mining tools [37].

This study expands previously published papers entitled: "Exploring Factors and Indicators for Measuring Students' Performance in Moodle Learning Environment." And "Tracking Student Performance Tool for Predicting Students EBPP in Online Courses," using the same dataset that has been used in the previously published papers in the International Journal of Emerging Technologies in Learning. The first paper used manual analysis to get the results. While the second paper used the tracking tool to get the results. Hence, this paper used different methods to analyze Moodle log file of the same dataset.

The authors distinguish some of the classification algorithms that will be chosen for analyzing a real dataset in order to have a better idea of how students will perform, engage, behave, and treat when dealing with Moodle courses. This is based on the Engagement, Behavior, Personality "EBP" predictive model that has been proposed by the authors previously. The prediction model is built on the course log files, which primarily indicate student's engagement, behavior and personality in the course. The instructor can create patterns of students from those log files to aid in the preparation of customized learning courses tailored to the needs of his/her students and therefore are meant to aid in the monitoring of the student's performance [1, 8].

As a result, in this paper, the authors examine and compare two different classification algorithms. Those algorithms are Naive Bayes (NB) and Random Forest (RF). These algorithms are compared in terms of their time taken to build the model, accuracy, correctly classified and incorrectly classified instances. The key target of this study is the detailed performance analysis of the two classification algorithms selected in the Weka tool (version 3.8.5) and to do a comparison among them, alongside, to assist the instructors in how to

* Email: ir.alkindi@gmail.com.

develop better learning strategies for their students based on their requirements.

The paper is organized as follows: “Background” section discusses the main concepts covered in this study. The section “Review of Relevant Literature” goes over the relevant literature for this study. The data and experiments are described in the section “Method.” The section "Results" contains and examines the steps to apply the selected algorithms in this study. The “Comparison” section highlights the main comparison results between NB and RF algorithms. Also, the section “Discussion” summarizes the main findings. Finally, section “Conclusions” concludes this study.

2 Background

2.1 Moodle Logfile

At Sultan Qaboos University (SQU), the teaching style is largely focused on a blended learning strategy that incorporates both online and traditional learning. Meanwhile, whenever academic year data is available, the student's engagement, behavior, personality, and performance are critical. Students’ successes in completing course assignments, as well as class participation and engagement, are important variables in improving the learning environment [8]. Moodle’s environment not only allows students convenient access to educational resources but also allows higher education institutions to collect massive amounts of data on student activity. This data is useful for assessing student behavior and determining whether there are any trends that contribute to enhanced learning outcomes [4]. Moodle logs maintain tracking of what materials students have accessed, updated, produced, and removed, as well as every click student and teacher make while navigating the site [19]. Table 1 shows an example of logfile in Moodle which consists of nine dimensions; “Time, User Full Name, Affected User, Event Context, Element, Event Name, Description, Origin and IP address”.

2.2 Student Engagement

Engaging in an activity is described as active, deliberate, and prolonged action and it is a hallmark of students’ genuine engagement with academic activities [38]. Moreover, Student engagement with activities of the course may be used to diagnose other crucial motivational processes that are not always obvious [21]. Experts are aware of the value of engagement in learning. Furthermore, empirical research has frequently shown the correlation between engagement and students’ performance [20].

2.3 Student Behavior

Studies of real online student behavior are necessary to discover the behaviors that contribute to online persistence and accomplishment for students and educators [32]. More and more, “student behaviors are significantly associated with many desirable academic and personal development outcomes of college,” as stated by the National Survey of Student Engagement. Standard college experiences for instance involve in collaborative learning activities and the number of hours spent on homework per week are among these behaviors [9]. Further, Zaiane stressed the need for distinguishing between various students' online activities [40]. Cantabella and colleagues advocated using big data technology to try to understand student behavior and draw conclusions about how to improve students’ performance by enhancing their learning process [12].

2.4 Student Personality

“A consistent predictor of student satisfaction, academic motivation, and academic performance” is personality [10]. Personality refers to individual differences in thought, mood, and behavior patterns [13]. McGeown and colleagues identified personality as “A set of fundamental qualities that influence

Table 1: An example of log file of the course in Moodle

Time	Full Name	Affected User	Event Context	Element	Event Name	Description	Origin
20/05/1916:00	SH	-	Forum: Alerts and Circulars	Forum	The course was reviewed	The user with id '13966' viewed the 'forum' activity with course module id '7453'.	web
12/05/1913:50	S H	-	Course: Internet search strategies	System	The course was reviewed	The user with id '13966' viewed the section number '-1' of the course with id '7453'.	web
4/03/1913:33	TA	-	Course: Internet search strategies	System	The course was reviewed	The user with id '26716' viewed the course with id '7453'.	web
19/02/1900:20	T L	-	Course: Internet search strategies	System	The course was reviewed	The user with id '26716' viewed the section number '-1' of the course with id '7453'.	web
……….	……….	……….	……….	……….	……….	……….	……….

how a person generally acts, thinks, and feels" [31].

2.5 Student Performance

Student performance means that the consequences of the teaching and learning process in terms of knowledge and skills obtained by students from schools and colleges are assessed by exams scores [30]. The quality of interpersonal contact in a course, according to Jaggars, has a favorable and meaningful relationship with student performance [22].

2.6 Classification Algorithms

Because data is not always in the best shape for analysis, new techniques for data analysis are needed to turn it into knowledge and information [33]. The authors are going to use classification algorithms, because it has been widely utilized in educational data mining and, also with the aim of predicting the student's performance along with their engagement, behavior and personality. One of the most common research challenges by machine learning researchers is classification. Predicting the value of the class attribute is based on the values of other attributes [36]. When the expected variable is binary or categorical, classification is employed [25]. The following machine learning algorithms were used to classify the students into three categories: Random Forest classifier (RF), Naive Bayes (NB), Logistic Regression (LR) and k-Nearest Neighbors (kNN) [34]. The authors focus in this paper on NB and RF algorithms.

2.7 Weka

In this paper the authors would like to use Weka (Waikato Environment for Knowledge Analysis), which is a set of machine learning algorithms for data mining. It is one of the most frequently used methods for finding knowledge from databases of data [35]. It consists of data preparation, classification, clustering, association rules and visualization tools [39] and built at the University of Waikato, New Zealand in 1997. It is applied in a wide range of applications, including educational, scientific and research purposes.

3 Literature Review

Methods, techniques, and the process of finding information through using Weka from logfile in Moodle LMS are discussed in this literature review as shown in Table 2.

Considering all existing work in Table 2 and based on authors' knowledge, the study described in this paper is the first to analyze student performance, engagement, behavior, and personality simultaneously using data from Moodle course logfiles with support of Weka classification algorithms.

Table 2: Literature review of using Weka to analyze Moodle log file

Author	Objective of the Study	Method	Dataset	Results	Future work/Limitations
[24]	Three classification data mining approaches for the detection of information presentation dimension (visual/verbal) learning style were compared using the Felder-Silverman Learning Style Model and the behavior of students in the Moodle course.	Using Weka an open-source software provides tools for implementation of several algorithms such as J48, Naive Bayes and PART.	Questionnaire and log data of Moodle	The best accuracy was achieved by the Naive Bayes algorithm, which achieved 71.18 % accuracy.	The authors want to incorporate Weka into Moodle so that they may update students' learning styles and adjust Moodle material using the data mining approach discovered in this study on Moodle log data.
[17]	This study used historical data from Moodle logs to preprocess and create machine learning models using Weka to track student performance and reduce the failure rate. Predictor qualities relating to student study behavior, such as Course Viewing Time, Quiz Taken, and so on, were used in this study.	Using Weka of J48, Random Forest, JRip, and OneR algorithms	Moodle Log of five courses	Students' performance was found to be significantly associated with predictive variables such as Activities Completed, Course Views, and Assignment Passed.	Other data mining tools and platforms are recommended to be used for comparative examination of the predictive analytics framework.
[23]	The purpose was to learn more about instructors' behavior and to create clusters based on the activities they did on the platform. The objective of this study is to boost the teaching process by devising particular approaches that will help students achieve greater success.	Weka and Hadoop	Moodle Log	Extracted knowledge from the activities of teachers	New found knowledge was used to enhance the teaching process and develop new instructional approaches in the future.

[18]	Educational Data Mining was applied to Moodle logs in order to see if the level of participation indicated in the amount of time spent using LMS services improved students' academic achievement.	Using machine learning algorithms of clustering and classification from WEKA system tools	Moodle Log	Findings showed that there is a significant relationship between the usage of Moodle resources and students' academic achievement. The findings are beneficial for strategic academic planning at the institution using LMS data.	Suggested that this study could be done with a variety of courses from other disciplines to see if academic disciplines had an impact on students' performance
[14]	Evaluating student prediction performance in Moodle and MOOCs based on their involvement with eLearning activities	Decision Tree, Artificial Neural Network, Support Vector Machine and KNN algorithms using Weka	Moodle log	The rate of interaction with the E-learning environment has a major influence on their performance, according to the study of log files, as students with the highest interactivity on the Moodle tend to do better than those with a low interactivity rate. Also, students spend more time on E-learning Moodle than MOOCs.	Not mention

4 Method

4.1 Materials and Dataset

The authors collected data from Moodle LMS log files for course "Search Strategies on the Internet" of 38 students (Table 1), offered by Department of Information Studies, College of Arts and Social Sciences, Sultan Qaboos University. Students' ability to discover information using multiple techniques such as search engines; library catalogs and online databases was improved via this course. The data taken contained 241896 logs.

4.2 Data Preprocessing

Firstly, the authors downloaded and extracted the Moodle logfile of the course. Secondly, feature selection and data filtering are all chosen during the preprocessing process. This study was carried out utilizing feature selection from the eight attributes listed in Table 3, the authors used Microsoft Excel for performing the preprocessing process. Then the feature selection was made into eight attributes that can be found in Tables 3 and 4, respectively. The next step is to run an algorithm test. The findings are documented and presented in tables and graphs in the next section's results.

The engagement, behavior, personality, and performance traits in numbers for each student are transformed into three categories by dividing the number using the percentile approach (High, Average and Low). It operates by dividing the data into irregular intervals, each pointing to a different category [5], as seen below in Table 4.

Table 5 shows the final feature selection. This file as CVS format will be imported later in Weka tool to compare the NB and RF.

Table 3: Attributes generated from data summarization of Moodle logfile

Attributes	Technical Definition	Type of Data
Engagement	A factor reflects the level of students' interactions with the activities of course in Moodle as getting exams, submit an assignment, etc.	Represent the data in numbers out of 100%
Engagement Category		Shows data in three categories (Low, Average and High)
Behavior	A factor gives the percentage of components that the student interacted with.	Represent the data in numbers out of 100%
Behavior Category		Shows data in three categories (Low, Average and High)
Personality	A factor represents the count of the accessed elements by the student.	Represent the data in numbers out of 100%
Personality Category		Shows data in three categories (Low, Average and High)
Performance	A factor indicates all the marks of students during the course.	Represent the data in numbers out of 100%
Performance Category		Shows data in three categories (Low, Average and High)

Table 4: Range with category division

Range	Category
0.00 - 35	Low
35.1 - 75	Average
75.1- 100.0	High

5 Results

Using the explorer application from Weka interface of Weka, explorer application is one of five available applications that has been used in this study. The authors applied Naive Bayes Algorithm and Random Forest Algorithm as displayed in Figures1 and 2 below.

Table 5: Nine attributes of the study after pre-processing [5]

Student ID	Engagement	Engagement Category	Behavior	Behavior Category	Personality	Personality Category	Performance	Performance Category
ST1	43.27	Average	64.7	Average	61.52	Average	69.5	Average
ST2	89.35	High	82.4	High	84.59	High	85.25	High
ST3	29.13	Low	70.6	Average	84.59	High	52.75	Average
ST4	61.24	Average	70.6	Average	76.9	High	86	High
ST5	62.44	Average	76.5	High	84.59	High	86.25	High
ST6	32.28	Low	70.6	Average	76.9	High	61	Average
ST7	67.46	Average	76.5	High	84.59	High	65.25	Average
ST8	66.78	Average	58.8	Average	69.21	Average	96.75	High
ST9	66.27	Average	64.7	Average	69.21	Average	88.75	High
ST10	68.82	Average	58.8	Average	69.21	Average	75.75	High
ST11	61.50	Average	64.7	Average	69.21	Average	76.25	High
ST12	53.32	Average	82.4	High	84.59	High	74.5	Average
ST13	52.39	Average	64.7	Average	69.21	Average	72.75	Average
ST14	100.00	High	76.5	High	76.9	High	83.5	High
ST15	58.52	Average	82.4	High	92.28	High	85	High
ST16	38.84	Average	64.7	Average	61.52	Average	56.75	Average
ST17	58.26	Average	88.2	High	99.97	High	86	High
ST18	26.75	Low	64.7	Average	61.52	Average	27.75	Low
ST19	79.81	High	82.4	High	76.9	High	85	High
ST20	92.50	High	70.6	Average	76.9	High	89.25	High
ST21	92.08	High	76.5	High	84.59	High	81	High
ST22	88.93	High	82.4	High	84.59	High	83	High
ST23	37.39	Average	58.8	Average	61.52	Average	68.25	Average
ST24	71.64	Average	76.5	High	84.59	High	87	High
ST25	45.74	Average	76.5	High	84.59	High	74.5	Average
ST26	46.17	Average	64.7	Average	69.21	Average	67.25	Average
ST27	58.94	Average	82.4	High	92.28	High	51.75	Average
ST28	55.11	Average	88.2	High	92.28	High	76.75	High
ST29	71.04	Average	70.6	Average	76.9	High	77.25	High
ST30	83.30	High	82.4	High	92.28	High	96.25	High
ST31	70.78	Average	76.5	High	84.59	High	88	High
ST32	53.92	Average	70.6	Average	76.9	High	75	Average
ST33	90.03	High	82.4	High	92.28	High	98.75	High
ST34	65.50	Average	82.4	High	99.97	High	97	High
ST35	48.98	Average	76.5	High	92.28	High	73.5	Average

ST36	55.20	Average	76.5	High	92.28	High	89.5	High
ST37	58.26	Average	70.6	Average	76.9	High	89.25	High
ST38	43.10	Average	70.6	Average	84.59	High	58	Average

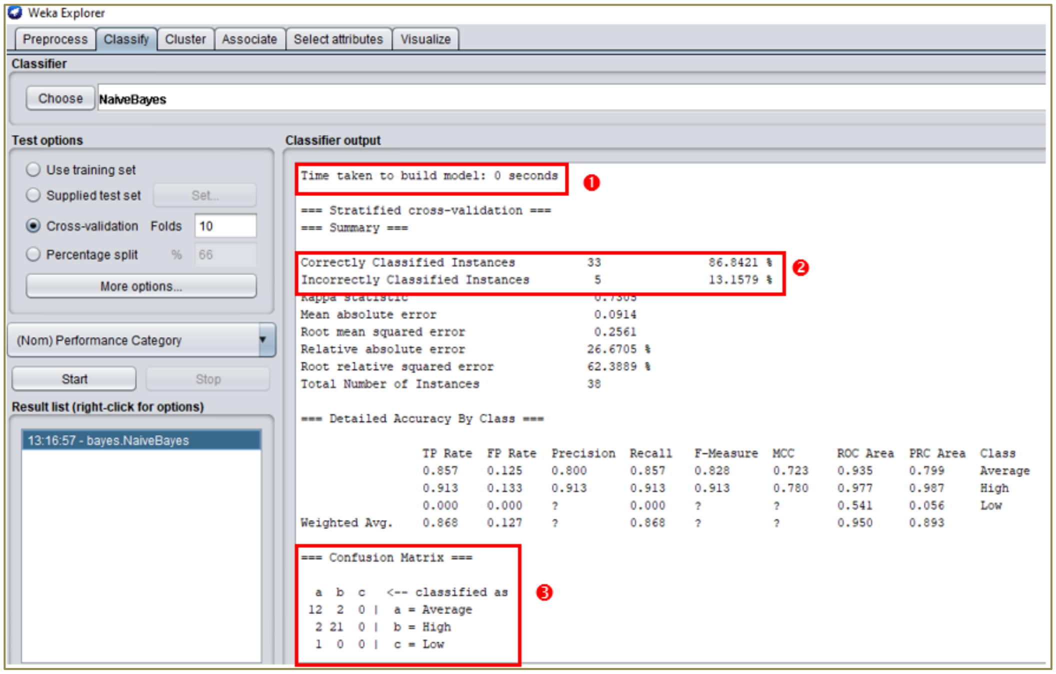


Figure 1: The results of conducting Naïve Bayes algorithm on the dataset

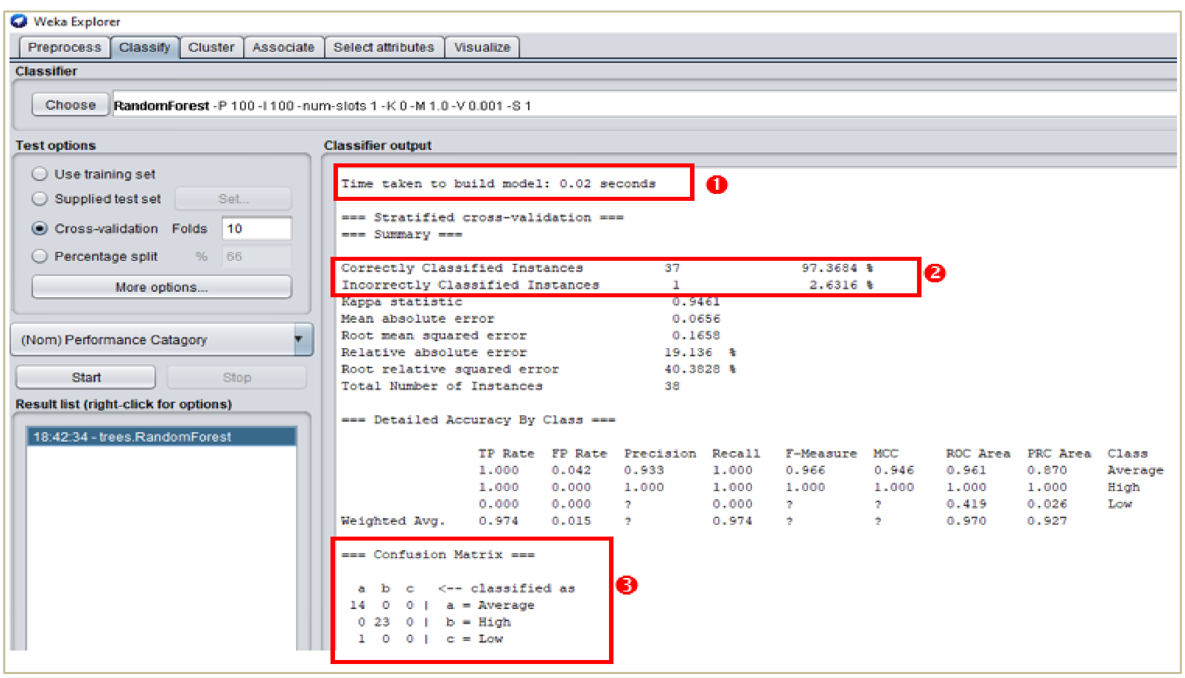


Figure 2: The results of conducting random forest algorithm on dataset

In general, accuracy indicates how often the variable is correct [11]. In Weka, each class has an alphabetical letter as a,b,c. In our experiment, Weka gives “a to Average,” “b to High,” and “c to Low.”

6 Comparison

The previous section discussed using the Weka tool to examine each of the two classification techniques on the “Search Strategies on the Internet” course dataset, which consists of eight attributes and contains data of 38 students. The dataset was classified using two classification algorithms included in the Weka tool: Naive Bayes (NB) and Random Forest (RF). Finally, the findings are as follows:

- **In terms of time taken to build the model**: NB shows 0 second. While RF shows 0.02 second.
- **In terms of accuracy**: The accuracy of NB algorithm is 86.84% compared to the accuracy that we got when we applied RF which is 97.36.%.
- **In terms of confusion matrix**: Using the Confusion Matrix, Naive Bayes classified 12 students correctly as Average and 21 as High which are 33. While 2 students have an average performance and are classified incorrectly as High. In addition, 2 students have a high performance and are classified incorrectly as Average. Lastly, the only student with low performance is classified incorrectly as Average. On the other side, Using the Confusion Matrix, Random Forest classified 14 students correctly as Average and 23 as High which are 37. Lastly, only 1 student also with low performance is classified incorrectly as Average.
- **In terms of correctly classified and incorrectly classified instances**: NB shows 33 correctly classified students from 38 while incorrectly classified 5 students. While RF shows 37 correctly classified students from 38 although incorrectly classified 1 student.

Overall, as referred to Table 6, RF performs the best compared to NB.

7 Discussion

As illustrated in Figure 3, the following features of a comparative analysis of two classification algorithms available in the Weka tools across the selected dataset as a case study in this paper have been examined: Time taken to build the model, accuracy, correctly classified and incorrectly classified instances. As a result of this investigation, the following conclusion may be drawn. We can analyze that the RF algorithm achieves accuracy greater than the NB algorithm. On the other hand, the NB is faster than RF.

Overall, the RF is better than NB because the accuracy is more important than time, especially the difference between the time of two algorithms is very small. In addition, the time of RF is due to the process of building many decision trees to select the best one of them. Based on the training instances and testing, each of them uses the same method which is 10-fold. This method trains all the instances and then tests the algorithm with the same data. This makes the performance more accurate and eliminates any overfitting of results. In addition, RF algorithm was able to correctly classify instances that reach 37 instances out of 38 instances which is the number of all students included in the dataset. In contrast, NB algorithm achieves 33 instances classified correctly. From the result, we see that time to build the RF model is more than using NB and correctly classifying instances are more and prediction accuracy is also greater in RF than the other. Hence it is concluded that RF performed better. To summarize, data mining algorithms are extremely beneficial for examining logfile data within the Moodle LMS to assess students' progress during any course.

8 Conclusions

There is a crucial need to monitor student engagement, behavior and personality in online courses and knowing how to respond to it, which is shown to improve student performance [1]. In such online courses, it is critical for instructors to be able to grasp the needs of each student [5]. Researchers can use good analytic techniques to intelligently examine students’ logfiles in educational systems [15]. Nevertheless, knowledge of the dataset and the type of analysis needed, all work to gather in selecting the best algorithm [33]. With the addition of new knowledge in E-learning, online courses, and the educational environment, such an analysis can allow more academics to dig deeper into the Moodle log file using any approach [6]. It allows for the generation of new data on a student's activity based on their digital profile [7]. So, using the log files of students in a Moodle course, the authors examined two classification data mining methods for forecasting student performance in this paper. The data used were gathered from Moodle log file of 38 students. The algorithms that were used in this study were: NB and RF. The Random Forest algorithm, which obtained 97.36%, had the best accuracy.

Table 6: Comparison of NB and RF algorithms results using the Weka tool

Algorithms	Instances	Attributes	Time Taken to build Model	Accuracy	Correctly Classified Instances	Incorrectly Classified Instances
Naive Bayes (NB)	38	8	0	86.84%	33	5
Random Forest (RF)	38	8	0.02	97.36%.	37	1

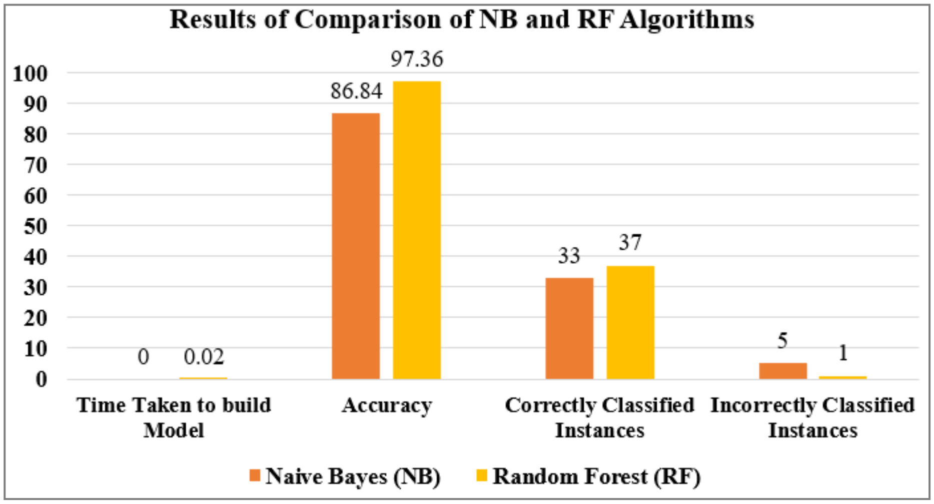


Figure 3: Results of comparison of NB and RF algorithms

Acknowledgement

The authors wish to thank Sultan Qaboos University, College of Science, and the Department of Computer Science. This work is under Prof. Zuhoor Al-Khanjari's supervision supported as a part of a scholarship of Doctoral Program from the Sultan Qaboos University.

References

[1] Z. Al-Khanjari and I. Al-Kindi, "Proposing a Systematic Framework for SQU-Smart Learning Management System (SQU-SLMS)", *International Journal of Computing and Digital Systems*. 10:1-13, https://doi.org /10.12785/ijcds/100169, 2021.

[2] I. Al-Kindi and Z. Al-Khanjari, "The Smart Learning Management System (SLMS)," *4th Free & Open-Source Software Conference* (FOSSC'2019-OMAN), ISSN: 1813-419X, https://fossc.om/2019/wp-content/uploads/ 2019/03/Proceedings.pdf#page=32, 2019.

[3] I. Al-Kindi and Z. Al-Khanjari, "A Novel Architecture of SQU SMART LMS: The New Horizon for SMART City in Oman," *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, IEEE, pp. 751-756, https://doi.org/10.1109/ICSSIT48917.2020. 9214141, 2020.

[4] I. Al-Kindi and Z. Al-Khanjari, "Proposing the EBP Smart Predictive Model Towards Smart Learning Environment," *Journal of Talent Development and Excellence*, 12(2s):2422-2438, https://iratde.com/index.php/jtde/ article/view/959, 2020.

[5] I Al-Kindi and Z. Al-Khanjari, "Exploring Factors and Indicators for Measuring Students' Performance in Moodle Learning Environment," *International Journal of Emerging Technologies in Learning*, 16(12):169-184, https://doi.org/10.3991/ijet.v16i12.22049, 2021.

[6] I. Al-Kindi and Z. Al-Khanjari, "NVivo to Analyze the Definition of Student EBP Qualitatively," *International Conference on IOT Based Control Networks and Intelligent Systems* (ICICNIS 2020), Available at SSRN: https://ssrn.com/abstract=3768217, http://dx.doi.org/ 10.2139/ssrn.3768217, 2021.

[7] I. Al-Kindi and Z. Al-Khanjari, "Tracking Student Performance Tool for Predicting Students EBPP in Online Courses," *International Journal of Emerging Technologies in Learning*. 16(23):140-156, https://doi.org/10.3991/ijet.v16i23.25503, 2021.

[8] I. Al-Kindi, Z. Al-Khanjari, and J. Al-Salmi, "Managing the Triangular Bond of the EBP for SQU Students Through the Proposed Test Model," *International Journal of Engineering and Advanced Technology (IJEAT)*, ISSN: 2249-8958, 10(1):391-402, October 2020, https://doi.org/10.35940/ijeat. A1914.1010120, 2020.

[9] R. D. Axelson and A. Flick, "Defining Student Engagement, Change: The Magazine of Higher Learning," 43(1):8-43, https://doi.org/10.1080/ 00091383. 2011.533096, 2010.

[10] K. K. Bhagat, L. Y. Wu, and C. Y. Chang, "The impact of personality on students' perceptions towards online learning," *Australasian Journal of Educational Technology*, 35(4), https://doi.org/10.14742/ajet 4162, 2019.

[11] Jason Brownlee, "A Gentle Introduction to K-Fold Cross-Validation," Retrieved from: https://machine learningmastery.com/k-fold-cross-validation/. Retrieved on: 18/8/2021, 2020.

[12] M. Cantabella, R. Martínez-España, B. Ayuso, J. A. Yáñez, and A. Muñoz, "Analysis of Student Behavior in Learning Management Systems Through a Big Data Framework," *Future Generation Computer Systems*. 90:262-272, https://doi.org/10.1016/j.future.2018.08.003, 2019.

[13] S. Caswell, J. P. Ambegaonkar, and A. M. Caswell, "Examination of Personality Traits in Athletic Training

Students," *International Journal of Athletic Therapy and Training*, 15(6):37-40, https://doi.org/10.1123/att.15.6.37, 2010.

[14] R. Chweya, S. M. Shamsuddin, S. S. M Ajibade, and S. Moveh, "A Literature Review of Student Performance Prediction in E-Learning Environment," *Journal of Science, Engineering, Technology and Management*, ISSN: 2795-3955, 2(1):1441-1453, Retrieved from https://www.jsetm.org/index.php/jsetm/article/ view/23, 2020.

[15] Data School, "Simple Guide to Confusion Matrix Terminology," Retrieved from: https://www.data school.io/simple-guide-to-confusion-matrix-terminology/, Retrieved on: 18/8/2021, 2014.

[16] M. Dougiamas and P. Taylor, "Moodle: Using Learning Communities to Create an Open-source Course Management System," *Proceedings of ED-MEDIA 2003--World Conference on Educational Multimedia, Hypermedia & Telecommunications*, pp. 171-178, Honolulu, Hawaii, USA: Association for the Advancement of Computing in Education (AACE). https://www.learntechlib.org/primary/p/13739/, 2003.

[17] E. D. Evangelista, "Development of Machine Learning Models using Study Behavior Predictors of Students' Academic Performance through Moodle," *International Journal of Innovative Technology and Exploring Engineering* (IJITEE), 8:22-27, https://www.ijitee.org/wpcontent/uploads/papers/v8i6s3/F10050486S319.pdf, 2019.

[18] U. O. Ezekiel and M. Mogorosi, "Educational Data Mining for Monitoring and Improving Academic Performance at University Levels," (IJACSA) *International Journal of Advanced Computer Science and Applications*. 11(11):570-581, https://doi.org/10.14569/IJACSA.2020.0111171, 2020.

[19] I. M. Félix, A. P. Ambrósio, P. S. Neves, J. Siqueira, and J. D. Brancher, "Moodle Predicta: A Data Mining Tool for Student Follow Up," *Proceedings of the 9th International Conference on Computer Supported Education (CSEDU 2017),* 1:339-346, https://doi.org/10.5220/0006318403390346, 2017.

[20] J. D. Finn and K. S. Zimmer, "Student Engagement: What is it? Why Does it Matter?" *Handbook of Research on Student Engagement,* Springer, Boston, MA, pp. 97-131, https://doi.org/10.1007/978-1-4614-2018-7_5, 2012.

[21] C. Furrer, G. Kelly, and E. Skinner, "Can Teachers use Children's Emotions in the Classroom to Diagnose and Treat Underlying Motivational Problems," *Poster Presented at the Biennial Meetings of the Society for Research in Child Development*, Tampa, FL, https://doi.org/10.1037/a0012840, 2003.

[22] S. Jaggars and D. Xu, "Predicting Online Student Outcomes from a Measure of Course Quality," *Journal of Research in Personality*, 35:78-90, https://doi.org/10.7916/D8N29TZH, 2013.

[23] L. Janevski, A. Velinov, and Z. Zdravev, "Analyzing Teachers Behavior Using Moodle Data and Big Data Tools," *Balkan Journal of Applied Mathematics and Informatics*, 2(1): 39-44. https://eprints.ugd.edu.mk/id/eprint/22199, 2019.

[24] A. Kika, L. Leka, S. Maxhelaku, and A. Ktona, "Using Data Mining Techniques on Moodle Data for Classification of Student? S Learning Styles," *Proceedings of International Academic Conferences* (No. 9211567), International Institute of Social and Economic Sciences, https://doi.org/10.20472/IAC.2019.047.010, 2019.

[25] R. G. Leppan, *A Model for Using Learners' Online Behavior to Inform Differentiated Instructional Design in MOODLE*, PhD Thesis, Faculty of Engineering, the Built Environment, and Information Technology, Nelson Mandela University, https://core.ac.uk/download/pdf/270043831.pdf, 2019.

[26] M. P. P. Liyanage, L. G. KS, and M. Hirakawa, "Detecting Learning Styles in Learning Management Systems using Data Mining," *Journal of Information Processing*, 24(4):740-749, https://doi.org/10.2197/ipsjjip.24.740, (2016).

[27] J. López-Zambrano, J. A. Lara, and C. Romero, "Towards Portability of Models for Predicting Students' Final Performance in University Courses Starting from Moodle Logs," *Applied Sciences*, 10(1):354, https://doi.org/10.3390/app10010354, 2020.

[28] J. López-Zambrano, J. A. Lara, and C. Romero, "Improving the Portability of Predicting Students' Performance Models by using Ontologies," *Journal of Computing in Higher Education*, pp. 1-19, https://doi.org/10.1007/s12528-021-09273-3, 2021.

[29] J. M. Luna, C. Castro, and C. Romero, "MDM Tool: A Data Mining Framework Integrated into Moodle," *Computer Applications in Engineering Education*, 25(1):90-102, https://doi.org/10.1002/cae.21782, 2017.

[30] J. H. Maganga, "Factors Affecting Students' Academic Performance: A Case of Public Secondary Schools in Ilala District," *Dar es Salaam (Doctoral Dissertation, The Open University of Tanzania),* http://repository.out.ac.tz/1732/1/JAMILLAH__MAGANGA-Dissertation_14-10-2016-Final.pdf, 2016.

[31] S. P. McGeown, D. Putwain, E. G. Simpson, E. Boffey, J. Markham, and A. Vince, "Predictors of Adolescents' Academic Motivation: Personality, Self-efficacy and Adolescents' Characteristics," *Learning and Individual Differences*. 32:278-286, https://doi.org/10.1016/j.lindif.2014.03.022, 2014.

[32] L. V. Morris, C. Finnegan, and S. S. Wu, "Tracking student behavior, persistence, and achievement in online courses*". The Internet and Higher Education*, 8(3):221-231, https://doi.org/10.1016/j.iheduc.2005.06.009, 2005.

[33] C. Pradana, S. S. Kusumawardani, and A. E. Permanasari, "Comparison Clustering Performance Based on Moodle Log Mining," *IOP Conference Series:*

Materials Science and Engineering, 722(1):012012), IOP Publishing. https://doi.org/ 10.1088/1757-899X/722/1/012012, 2020.

[34] G. S. Ramaswami, T. Susnjak, and A. Mathrani, “Capitalizing on Learning Analytics Dashboard for Maximizing Student Outcomes,” *2019 IEEE Asia-Pacific Conference on Computer Science and Data Engineering* (CSDE), pp. 1-6, IEEE. https://doi.org/10.1109/CSDE48274.2019.9162357, 2019.

[35] B. Rawat and S. K. Dwivedi, “An Architecture for Recommendation of Courses in E-learning System” *Int. J. Inf. Technol. Comput. Sci*. (IJITCS), 9(4):39-47, https://doi.org/10.5815/ijitcs.2017.04.06, 2017.

[36] C. Romero, S. Ventura, P. G. Espejo, and C. Hervás, “Data Mining Algorithms to Classify Students,” *Educational Data Mining 2008*, https://www.researchgate.net/profile/Michel-Desmarais-2/publication/221570516_Adaptive_Test_Design_with_a_Naive_Bayes_Framework/links/09e41512380f4e4bc1000000/Adaptive-Test-Design-with-a-Naive-Bayes-Framework.pdf#page=8, (2008).

[37] C. Romero, S. Ventura, and E. García, “Data Mining in Course Management Systems: Moodle Case Study and Tutorial,” *Computers & Education*, 51(1):368-384, https://doi.org/10.1016/j.compedu.2007.05.016, 2007.

[38] E. A. Skinner and J. R. Pitzer, “Developmental Dynamics of Student Engagement, Coping, and Everyday Resilience,” *Handbook of Research on Student Engagement*, Springer, Boston, MA, pp. 21-44, https://doi.org/10.1007/978-1-4614-2018-7_2, 2012.

[39] “Weka 3: Machine Learning Software in Java,” Retrieved from: https://www.cs.waikato.ac.nz/ml/weka/, Retrieved on: 22/8/2021

[40] O. Zaiane, “Web Usage Mining for a Better Web-based Learning Environment,” Available online at https://era.library.ualberta.ca/items/0a182195-ce39-4b5d-a1c1-291ed91a0f36/view/336c2a34-b149-4d9d-95a1-ad78c08ee35c/TR01-05.pdf, 2001.

Iman Al-Kindi is a PhD graduate student from the Department of Computer Science, College of Science at Sultan Qaboos University, Sultanate of Oman. She received her BSc in software Engineering from Higher College of Technology, Sultanate of Oman, and MSc in Computer Science from Sultan Qaboos University, Sultanate of Oman. She has worked as a visiting lecturer for more than one year at Sultan Qaboos University, Sultanate of Oman.

Zuhoor Al-Khanjari is a Professor in Software Engineering. She worked as the HOD of the Department of Computer Science, College of Science at Sultan Qaboos University, Sultanate of Oman. She received her BSc in mathematics and computing from Sultan Qaboos University, Sultanate of Oman, MSc and PhD in computer science (software engineering) from the University of Liverpool, UK. Her research interests include software engineering, software testing techniques, database management, e-learning, m-learning and mobile computing. Currently, she is the coordinator of the software engineering group in the Department of Computer Science, Sultan Qaboos University, Sultanate of Oman. Also, she is coordinating e-learning facilities in the same department. She is a member of the editorial board of the International Arab Journal of Information Technology (IAJIT) and a member of the executive committee of the International Arab Conference on Information Technology (ACIT).

Journal Submission

The International Journal of Computers and Their Applications is published four times a year with the purpose of providing a forum for state-of-the-art developments and research in the theory and design of computers, as well as current innovative activities in the applications of computers. In contrast to other journals, this journal focuses on emerging computer technologies with emphasis on the applicability to real world problems. Current areas of particular interest include, but are not limited to: architecture, networks, intelligent systems, parallel and distributed computing, software and information engineering, and computer applications (e.g., engineering, medicine, business, education, etc.). All papers are subject to peer review before selection.

A. Procedure for Submission of a Technical Paper for Consideration

1. Email your manuscript to the Editor-in-Chief, Dr. Ajay Bandi. Email: ajay@nwmissouri.edu.
2. Illustrations should be high quality (originals unnecessary).
3. Enclose a separate page (or include in the email message) the preferred author and address for correspondence. Also, please include email, telephone, and fax information should further contact be needed.
4. **Note**: Papers shorter than 10 pages long will be returned.

B. Manuscript Style:

1. **WORD DOCUMENT**: The text should be **double-spaced** (12 point or larger), **single column** and **single-sided** on 8.5 X 11 inch pages. Or it can be single spaced double column.

   LaTex DOCUMENT: The text is to be a double column (10 point font) in pdf format.
2. An informative abstract of 100-250 words should be provided.
3. At least 5 keywords following the abstract describing the paper topics.
4. References (alphabetized by first author) should appear at the end of the paper, as follows: author(s), first initials followed by last name, title in quotation marks, periodical, volume, inclusive page numbers, month and year.
5. The figures are to be integrated in the text after referenced in the text.

C. Submission of Accepted Manuscripts

1. The final complete paper (with abstract, figures, tables, and keywords) satisfying Section B above in **MS Word format** should be submitted to the Editor-in-Chief. If one wished to use LaTex, please see the corresponding LaTex template.
2. The submission may be on a CD/DVD or as an email attachment(s). **The following electronic files should be included:**
   - Paper text (required).
   - Bios (required for each author).
   - Author Photos are to be integrated into the text.
   - Figures, Tables, and Illustrations. These should be integrated into the paper text file.
3. Reminder: The authors photos and short bios should be integrated into the text at the end of the paper. All figures, tables, and illustrations should be integrated into the text after being mentioned in the text.
4. The final paper should be submitted in (a) pdf AND (b) either Word or LaTex. For those authors using LaTex, please follow the guidelines and template.
5. Authors are asked to sign an ISCA copyright form (http://www.isca-hq.org/j-copyright.htm), indicating that they are transferring the copyright to ISCA or declaring the work to be government-sponsored work in the public domain. Also, letters of permission for inclusion of non-original materials are required.

Publication Charges

After a manuscript has been accepted for publication, the contact author will be invoiced a publication charge of **$500.00 USD** to cover part of the cost of publication. For ISCA members, publication charges are **$400.00 USD** publication charges are required.

Revised 2020