International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Enhancing Forensic Accuracy with IFDenseNet-138 from the Sensor Data Analysis using Evidence Collector (EC) Mobile Application

Sukhada Aloni [1], Divya Shekhawat [2]

Submitted: 28/12/2023 **Revised**: 06/02/2024 **Accepted**: 12/02/2024

Abstract: In this research, we present the development of an open-source mobile application called Evidence Collector (EC) for collecting sensor data from mobile devices. The EC app was built using the Thunkable platform and is available for public use. In addition, we propose a deep neural network algorithm, IFDenseNet- 138, which is designed to analyze the sensor data collected by the EC app. The IFDenseNet-138 algorithm is capable of accurately detecting up to 10 different types of occurrences at a crime scene. An IFDenseNet-138 model achieved an accuracy of 96.32% with high precision and recall scores, indicating its effectiveness in performing 10-class classification. The primary objective of this research is to aid forensic teams in solving mystery cases where the victim is deceased but their mobile device data is available. The utilization of such data can help forensic teams in reconstructing the events leading to the crime, identifying suspects, and gathering evidence for legal proceedings.

Keywords: *Forensic analysis; Mobile application; Sensor data; IFDenseNet-138; Crime scene investigation*

1. Introduction

Approximately Approximately 7.5 lakh police cases are dismissed annually in India due to the absence of evidence. Mystery cases in India remain unsolved or unprosecuted due to a lack of evidence (Mangoli and Tarase, 2010) . The Indian criminal justice system has been criticized for its low conviction rate, which is partly attributed to the lack of proper investigation and inadequate collection and preservation of evidence (Dash, 2020) . Factors such as underfunding, understaffing, and lack of training in forensic science and technology in law enforcement agencies has resulted in poor evidence collection and preservation practices, leading to difficulty in solving the cases and low conviction rates (Senek et al., 2022) . Mobile device forensics can solve this issue. Mobile device forensics is a branch of digital forensics that deals with the recovery of digital evidence or data from mobile devices in a forensically sound manner (Omeleze and Venter, 2013) . The increasing popularity of smartphones and other digital devices among consumers has led to a growing demand for forensic examination of these devices, which cannot be effectively addressed by traditional computer forensics techniques (Bennett, 2012) .

The mobile forensics process is a method of recovering and analyzing digital evidence from mobile devices in a forensically sound manner (Sathe and Dongre, 2018) . It is typically divided into three main stages: seizure, acquisition, and examination/analysis. The seizure stage involves the physical collection of the mobile device(s) and any associated peripheral devices or storage media. It is important that the devices are collected in a manner that preserves their integrity and authenticity, and that any chain of custody is clearly established and documented. The acquisition stage involves the creation of a forensic image of the mobile device(s) and any associated storage media. This process involves making a bit-by- bit copy of the entire storage area of the device, preserving all data including deleted files, slack space, and unallocated space. The examination/analysis stage involves the analysis of the forensic image(s) to recover and interpret the digital evidence. This process can include the use of forensic software tools, manual examination, and the application of specialized knowledge and techniques to extract, analyze and interpret the data (Servida and Casey, 2019) .

Mobile phone data can be used as evidence in legal proceedings in India under the Indian Evidence Act, 1872 and the Indian Information Technology Act, 2000. Admissibility of mobile phone data as evidence in court requires the relevance of the data, authenticity and integrity of the data. However, the use of mobile phone data as evidence is subject to certain conditions and may require technical experts to authenticate the data. Additionally, collecting, obtaining, and using mobile phone data as evidence must comply with data privacy and data protection laws, and must be in accordance with the Indian Constitution and the laws of the land.

Forensic experts may use mobile phone data to help solve a case in a variety of circumstances (Ribaux et al., 2006) . For example, in criminal investigations, mobile phone data can be used to track the movements and activities of a

[1] *Research Scholar, Pacific University, Udaipur, 313024, Rajasthan, India, E-mail: sukhada.aloni@gmail.com*
2 Assistant Professor,Pacific University, Udaipur, 313024, Rajasthan, India, E-mail: divya.shekhawat23@gmail.com

suspect, and can be used to establish a suspect's alibi or link them to a crime scene. In personal injury cases, mobile phone data can be used to determine the cause of an accident by analyzing the phone's GPS data and call logs. In fraud investigations, mobile phone data can be used to uncover fraudulent activity by analyzing call logs, text messages, and other electronic communications. In digital forensics, mobile phone data can be used to recover deleted or hidden data, such as text messages, call logs, and photos. In domestic investigations, mobile phone data can be used to uncover infidelity, abuse or other forms of misconduct by analyzing call logs, text messages, and other electronic communications. In cybercrime investigations, mobile phone data can be used to track the activities of cybercriminals by analyzing the phone's internet history, apps, and other digital traces. However, it's worth noting that the use of mobile phone data as forensic evidence is subject to certain conditions and may require the involvement of technical experts to authenticate the data and establish its integrity.

Deep learning is a subfield of machine learning that is inspired by the structure and function of the brain's neural networks (Voican, 2021) . It involves training artificial neural networks on a large dataset, allowing the network to learn and make decisions or predictions on its own. Deep learning algorithms consist of multiple layers of interconnected "neurons" that process and analyze the input data, allowing the network to learn and improve over time. These layers of neurons can be thought of as a hierarchy, with the early layers recognizing simple features and the later layers recognizing more complex features and patterns.

Deep learning is needed for crime investigation with mobile data as it allows the analysis of large and complex datasets that would be difficult or impossible to process using traditional methods (Chen et al., 2021) . Handling large amounts of data is one of the main advantages, as mobile data can include a vast amount of information such as call logs, text messages, GPS data, and social media activity. Deep learning algorithms can handle this large amount of data and extract relevant information from it. Identifying patterns and connections is another advantage as it can identify patterns and connections in the data that may not be immediately apparent to a human investigator, such as analyzing call logs to identify connections between suspects and possible accomplices, or use GPS data to track a suspect's movements over time. Additionally, deep learning can be used to analyze images and videos from CCTV cameras and mobile phones, which can be crucial for identifying suspects or reconstructing the events of a crime (Xiao et al., 2019) . It can also be used to analyze text messages, social media posts, and other written communications to understand the intent of the person and uncover any criminal activities. Furthermore, it can perform real-time analysis which can be important for identifying and responding to criminal activity as it occurs.

DenseNet is a type of convolutional neural network (CNN) that is designed to improve upon the performance of traditional CNNs. It is a variation of the ResNet architecture and it is based on the idea of feature reuse. DenseNet uses "dense blocks" which connect each layer to every other layer in a feed-forward fashion. This allows DenseNet to efficiently reuse features from previous layers, reducing the number of parameters and computational resources required to achieve the same level of performance as a traditional CNN.The main advantage of DenseNet is that it addresses the problem of vanishing gradients, which can occur in deep neural networks. This problem occurs when the gradients of the error function become very small as they are backpropagated through the layers, making it difficult for the network to learn. DenseNet addresses this issue by connecting each layer to all previous layers, allowing the gradients to flow more easily through the network.

2. Related Research

Al et al. (Al-Dhaqm et al., 2020) conducted a review of Mobile Forensics Investigation Process Models (MFIPMs) in the field of Mobile Forensics (MF), with a focus on recovering Potential Digital Evidence (PDE) from mobile devices using forensic techniques. They aim to uncover the transitions in the MF field and identify open and future challenges. They reviewed 100 MFIPMs and proposed a Harmonized Mobile Forensic Investigation Process Model (HMFIPM) to unify and structure the investigation processes in the MF field

Teing et al. (Teing et al., 2018) address the challenges of cloud forensic efforts due to the significant increase in the volume, variety, and velocity of data. They stress on the importance of digital forensic practitioners having up-to-date knowledge of relevant data artifacts that can be recovered from the cloud product under investigation. They specifically studied CloudMe, a popular cloud storage service and described the types and locations of artifacts related to the installation and uninstallation of CloudMe client application, logging in and out, and file synchronization events from the computer desktop and mobile clients. The findings from this research will also help inform future development of tools and techniques such as data mining techniques for cloud-enabled big data endpoint forensic investigations.

Kim et al. (Kim et al., 2018) propose the mobile forensic reference set (MFReS) as a mobile forensic investigation procedure and a tool for mobile forensics. The MFReS is designed to be a collection of repositories, databases, and services that can easily retrieve data from a database for effective classification of meaningful data related to a

crime among numerous data types in mobile devices. The MFReS is able to effectively analyze information from installed applications and user behavior through system data, application data, and multimedia data. The tool developed by the team is useful for investigators as it can analyze log files of all applications and analyze behavior based on timeline, geodata, and other characteristics. The research contributes to the study of mobile forensic support systems and suggests the direction of mobile data analysis tool development.

Mumba et al. (Mumba and Venter, 2014) examine the challenges of mobile forensic investigations in light of the fast-developing technology of mobile devices. The authors acknowledge that while mobile devices have become a vital source of digital evidence, performing data acquisition in a forensically sound manner from mobile devices can be a challenge. The paper aims at testing the performance of a Harmonized Digital Forensic Investigation Process (HDFIP) as stipulated in the ISO/IEC 27043 draft international standard through the extraction of potential digital evidence from mobile devices, with the help of a case study. The goal is to evaluate the effectiveness of the HDFIP in mobile forensic investigations and to identify areas that need improvement.

3. Methodology

Figure 1 IFDenseNet-138 Conceptual Diagram for Classifying Mobile Phone Sensor Data. With 138 layers of dense connections, the model accurately assigns the input data obtained from mobile phone sensors, represented as a 2-D array, to one of the 10 classes. Each row in the 2-D array represents individual sensors and each column represents a time series. The data is then processed through a 32x3 convolutional layer with ReLU activation for 138 times. To reduce the dimensions, a max pooling layer and a flattened layer was used. Finally, the data is classified using the Softmax function.

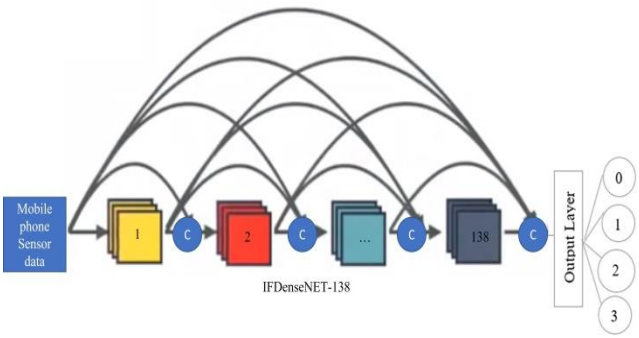

Fig 1

We have used Samsung S22 Ultra for collection of results. The Samsung S22 Ultra model of mobile phone has several sensors, including an under display ultrasonic fingerprint sensor, an accelerometer, a gyroscope, a proximity sensor, a compass sensor, and a barometer sensor. The fingerprint sensor is used for biometric authentication, the accelerometer and gyroscope are used to detect changes in the device's orientation, the proximity sensor is used to detect when the device is close to the user's face, the compass sensor is used to detect the device's orientation in relation to magnetic north and the barometer sensor is used to measure the atmospheric pressure. Additionally, the Samsung S22 Ultra has Samsung DeX, Samsung Wireless DeX (desktop experience support), Bixby natural language commands and dictation, Samsung Pay (Visa, MasterCard certified) and Ultra Wideband (UWB) support.

The goal of the research was to develop an app capable of collecting data from mobile phones for use in investigations or legal cases. The first step was to collect data for the training dataset, which was done by using trained actors to simulate 10 real-life scenarios, such as a teenager running for survival, a man getting dragged into a car during a kidnapping, and a woman calling for help during a kidnapping. The names of the 10 output classes include "Teenager Running for Survival," "Male Sitting in Car Just Before Accident," "Kids Fighting," "Woman Sitting in Moving Vehicle Just Before Car Crash," "Women Calling for Help During Kidnapping," "Woman Struggling Against Attackers," "Throwing of Mobile Phones to Protect Evidence," "Women Running to Save Herself," "Man Getting Dragged into Car During Kidnapping," and "Man Enters Vehicle by Forcibly Pushing Woman onto Back Seat."

Figure 2 Block coding of a mobile application for evidence collection. Button 1 triggers data collection from mobile sensors, which is then sent to a cloud variable. The app stores data on the device's local memory when there is no internet connection. Panel a of the interface displays start and stop buttons, current sensor readouts, and a loading symbol during data upload. Block code consisted of the initialization of the cloud variable and do-while loop that determines whether the app is in sleep mode or collecting sensor data. When collecting, the app loads gyroscope readings into the cloud variable and stores data in table 1. The app variable A is set to 1 when data is being recorded, and 0 when the stop button is pressed or data is already stored. The app remains in an infinite loop until the button is pressed again.

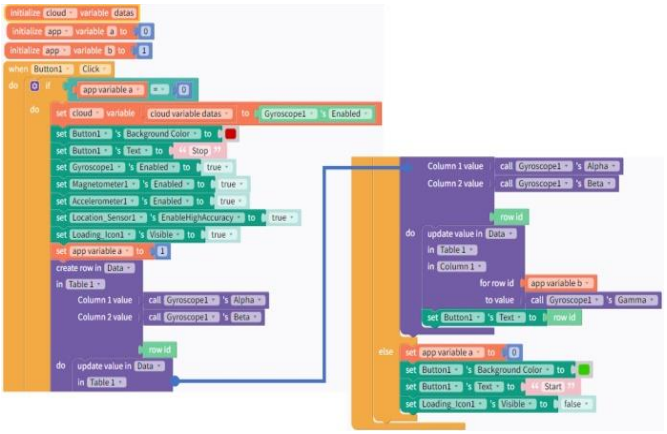

Fig 2

Figure 2 shows the block coding of a proposed mobile application designed for collecting evidence. When the user clicks button 1, the application starts gathering data from the gyroscope, magnetometer, accelerometer, and location sensor. This data is then sent to a cloud variable that is periodically updated. If there is no internet connection, the data is stored on the mobile device's local memory.

Figure 3 Panel a shows the user interface display with the blue start button, indicating that the program has just started and has no data in either the cloud or local memory. When the app is collecting data from all the mobile sensors, the interface shows the red stop button (Fig. 3b), and the current readout from each sensor is displayed above the button. The rotating loading symbol is just a visual representation displayed, while data is uploaded to the cloud. The evidence collector app with the green start button indicates that the data is already stored in local memory and has also been uploaded to the cloud.

Figure 2 shows the block coding of the proposed evidence collector app. The red blocks represent the initialization of the cloud variable data S with two local app variables, A and B, where A is initialized to 0, and B is initialized to 1. The app has a single button called button 1 in the front end, which is initially blue (Fig. 3a). When the user presses the button, the code goes into a do-while loop, where app variable A's value decides whether all sensor readings are being stored, or the app is in sleep mode. In sleep mode, the app variable A value is reset, and the button background color is set to green (Fig. 3c), the button text is set to Start, and the loading icon's visibility is set to false. If the app variable A is zero, then the 'if' loop is entered.

In this loop, the cloud variable is loaded with the gyroscope's readings, and the button's background color is set to red (Fig. 3b), while the button's text is set to Stop. All four sensors (gyroscope, magnetometer, accelerometer, and location sensor) are enabled, and the loading icon's visibility is set to true. A new variable data is created in table 1, which holds the values of alpha, beta, and gamma. Alpha and beta are stored as separate columns, and variable b is created to store the gamma into the data variable of table 1 in column 1, just below the value of alpha.

The app variable A is set to 1 just before storing the gyroscope values. If the app variable value is already 1, meaning that the data is recorded or the stop button is pressed, the app variable is reset to zero, and the background color of button 1 is set to green (Fig. 3c). Additionally, button 1's text is set to Start, and the loading icon's visibility is set to false. The code remains in an infinite loop and waits for the button press.

For analysis of the proposed network following formulas were used.

The precision formula in terms of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) is:

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

The formula for recall in terms of True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN) is:

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

The accuracy can be represented in terms of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

4. Results

Figure 3 displayed the front end of the Evidence Collector (EC) mobile collection app, which had three different states: initialization (Fig. 3a), data collection (Fig. 3b), and sleep mode (Fig. 3c). The initialization state was entered either when the user first launched the app or when the app had been closed and reopened. In this state, the EC app's front panel displayed a blue button, which indicated that no data had been collected. The front panel provided no indication of data collection or upload, and the user could proceed to the next state by pressing the blue start button.

In the data collection state, the front panel displayed a red button, which indicated that the EC app was actively collecting data. The current readout from each sensor was

displayed above the button, and a rotating loading symbol was displayed while data was uploaded to the cloud. The app collected data from the gyroscope, magnetometer, accelerometer, and location sensor, and this data was sent to a cloud variable that was periodically updated. If there was no internet connection, the data was stored on the mobile device's local memory.

Figure 3 Front end states of the EC mobile collection app. (a) Initialization state, indicated by a blue button, showing that no data has been collected yet. (b) Data collection state, indicated by a red button and display of current sensor readings, while data is being collected. (c) Sleep mode state, indicated by a green button, which shows that the data is already stored in local memory and has also been uploaded to the cloud.

Fig 3

Figure 4 The confusion matrix showing the performance of the proposed IFDenseNet-138 , which depicts the accuracy of its 10-class classification. The model achieved an accuracy of 96.32%, with an average precision of 96.36% and an average recall of 96.4%.

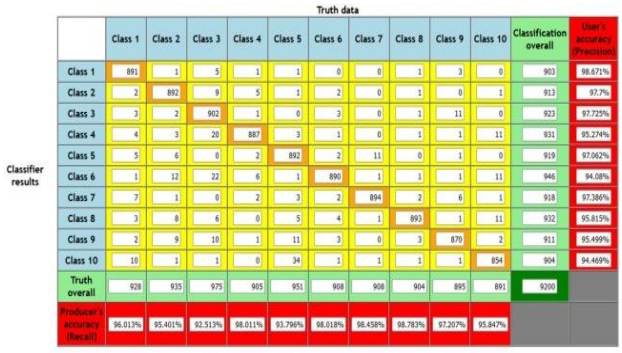

Fig 4

by pressing the red button, and if they did so, the app entered the sleep mode state.

The IFDenseNet-138 model was able to correctly classify the majority of the data, achieving an overall accuracy of 96.32%. Furthermore, the model's precision and recall scores were both quite high, with an average precision of 96.36% and an average recall of 96.4%. These scores indicate that the model was able to effectively identify the various classes and minimize both false positives and false negatives.

Overall, these results suggest that the proposed IFDenseNet-138 model is highly accurate and effective in performing 10-class classification. The high precision and recall scores demonstrate the model's ability to correctly identify each class while minimizing the likelihood of misclassification.

4.1. Discussions

The results of this study demonstrate the potential for using the IFDenseNet-138 model to enhance the accuracy of forensic analysis from sensor data collected by the Evidence Collector (EC) mobile application. By achieving high precision and recall rates for identifying specific events, such as "Teenager Running for Survival," "Male Sitting in Car Just Before Accident," "Kids Fighting," "Woman Sitting in Moving Vehicle Just Before Car Crash," "Women Calling for Help During Kidnapping," "Woman Struggling Against Attackers," "Throwing of Mobile Phones to Protect Evidence," "Women Running to Save Herself," "Man Getting Dragged into Car During Kidnapping," and "Man Enters Vehicle by Forcibly Pushing Woman onto Back Seat.", this model can aid in the investigation and reconstruction of crime scenes.

One potential application of this technology is in the analysis of data collected from wearable devices, which are increasingly common in modern society. By analyzing the data collected from these devices, investigators may be able to gain insight into the activities of suspects and victims, helping to build a more comprehensive picture of events leading up to a crime.

However, there are also limitations to the use of this technology. For example, the IFDenseNet-138 model was trained on a specific dataset, and its performance may not generalize to other datasets or real-world scenarios. Additionally, the accuracy of the model may be impacted by factors such as sensor placement and environmental conditions.

Furthermore, while the use of technology in forensic analysis has the potential to increase accuracy and efficiency, it is important to consider ethical and legal implications. For example, there may be concerns around privacy and data security when analyzing data collected from wearable devices or other sources. Additionally, it is important to ensure that the use of this technology does not lead to biases or inaccuracies in the investigation process.

The use of the IFDenseNet-138 model in forensic analysis shows promise for enhancing accuracy and aiding in the investigation of crimes. However, further research is needed to validate its performance in real-world scenarios and to address potential ethical and legal concerns.

5. Conclusions

In conclusion, this research presents the development of an

open-source mobile application called Evidence Collector (EC) which is designed to collect sensor data from mobile devices. The app was built using the Thunkable platform and is available for public use. The authors also propose a deep neural network algorithm, IFDenseNet-138, which is capable of analyzing the sensor data collected by the EC app and accurately detecting up to 10 different types of occurrences at a crime scene. The objective of the research is to assist forensic teams in solving mystery cases where the victim is deceased but their mobile device data is available.

The utilization of such data can help forensic teams in reconstructing the events leading to the crime, identifying suspects, and gathering evidence for legal proceedings. The development of the EC app and IFDenseNet-138 algorithm can greatly aid in the field of forensic investigation and provide a new tool for forensic teams to use in solving complex cases. This research has significant implications for the field of forensic science and may pave the way for further developments in this area.

5.1. Conflicts of interest or competing interests:

Authors declare that there is no Conflict of interest or competing interests.

5.2. Data and code availability:

Data and code will be made available on reasonable request to the Authors.

5.3. Supplementary information:

Not applicable.

5.4. Ethical approval:

All the ethics approval was taken by an institutional review board or equivalent ethics committee.

5.5. Funding statement:

This research has no funding associated with it.

5.6. Competing interests:

Authors declare that there is no competing interests.

5.7. Availability of data and materials:

Data will be made available on reasonable request to corresponding author.

5.8. Ethics statements:

All authors consciously assure that the manuscript fulfills the following statements:

1) This material is the author's original work, which has not been previously published elsewhere.
2) The paper is not currently being considered for publication elsewhere.
3) The paper reflects the author's own research and analysis truthfully and completely.
4) The paper properly credits the meaningful contributions of co-authors and co-researchers.
5) The results are appropriately placed in the context of prior and existing research.

References

[1] Al-Dhaqm, A., Razak, S. A., Ikuesan, R. A., Kebande, V. R., and Siddique, K. (2020). 'A review of mobile forensic investigation process models'. IEEE access, Vol 8, pp. 173359–173375.

[2] Bennett, D. (2012). 'The challenges facing computer forensics investigators in obtaining information from mobile devices for use in criminal investigations'. Information Security Journal: A Global Perspective, Vol 21, No 3, pp. 159–168.

[3] Chen, K., Zhang, D., Yao, L., Guo, B., Yu, Z., and Liu, Y. (2021). 'Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities'. ACM Computing Surveys (CSUR), Vol 54, No 4, pp. 1–40.

[4] Dash, P. P. (2020). 'Rape adjudication in India in the aftermath of Criminal Law Amendment Act, 2013: findings from trial courts of Delhi'. Indian Law Review, Vol 4, No 2, pp. 244–266.

[5] Kim, D., Lee, Y., and Lee, S. (2018). 'Mobile forensic reference set (MFReS) and mobile forensic investigation for android devices'. The Journal of Supercomputing, Vol 74, No 12, pp. 6618–6632.

[6] Mangoli, R. N. and Tarase, G. M. (2010). 'A study of human rights violation by police in India'. International Journal of Criminology and Sociological Theory, Vol 3, No 2,.

[7] Mumba, E. R. and Venter, H. S. (2014). 'Mobile forensics using the harmonised digital forensic investigation process'. In 2014 Information Security for South Africa, pp. pages 1–10.

[8] Omeleze, S. and Venter, H. S. (2013). 'Testing the harmonised digital forensic investigation process model-using an Android mobile phone'. In 2013 Information Security for South Africa, pp. pages 1–8.

[9] Ribaux, O., Walsh, S. J., and Margot, P. (2006). 'The contribution of forensic science to crime analysis and investigation: forensic intelligence'. Forensic science international, Vol 156, No 2-3, pp. 171–181.

[10] Sathe, S. C. and Dongre, N. M. (2018). 'Data acquisition techniques in mobile forensics'. In 2018 2nd international conference on inventive systems and control (icisc), pp. pages 280–286.

[11] Senek, M., Robertson, S., Taylor, B., Wood, E., King, R., and Ryan, T. (2022). 'Consequences of understaffing on type of missed community care-a cross-sectional study'. International Journal of Nursing Studies Advances, Vol 4, pp. 100075.

[12] Servida, F. and Casey, E. (2019). 'IoT forensic challenges and opportunities for digital traces'. Digital Investigation, Vol 28, pp. 22–29.

[13] Teing, Y.-Y., Dehghantanha, A., and Choo, K.-K. R. (2018). 'CloudMe forensics: A case of big data forensic investigation'. Concurrency and Computation: Practice and Experience, Vol 30, No 5, pp. e4277.

[14] Voican, O. (2021). 'Credit Card Fraud Detection using Deep Learning Techniques'. Informatica Economica, Vol 25, No 1,.

[15] Xiao, J., Li, S., and Xu, Q. (2019). 'Video-based evidence analysis and extraction in digital forensic investigation'. IEEE Access, Vol 7, pp. 55432–55442.