Journal of Data Science and Cyber Security, ISSN: 2584-0010, Volume 1 Issue 1 June 2023

Malware Detection Using Machine Learning and Deep Learning

Arun Venkat S J[1], John Chacko[2] and Micheal Olaolu Arowolo[3,*]

[1,2]School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, 600127, Tamilnadu, India

[3] Landmark University – Omu-aran, Kwara State, Nigeria arowolo.olaolu@gmail.com

***Corresponding Author:** Micheal Olaolu Arowolo, arowolo.olaolu@gmail.com

Abstract

Malicious software, generally known as "malware," is becoming alarmingly more prevalent, and some of it uses various techniques to conceal itself on your system. To keep computers and the Internet secure, malware must be identified before it infects numerous systems. Malware is harmful software that sneaks onto your computer by pretending to be a legitimate application. It can be installed in many ways, but phishing emails, fake installers, infected attachments, and phishing links are the most common. Hackers show malware to users so they will install it. In many cases, users are unaware that the program is malware. Basically, this is how malware gets installed on your computer. Malware hides in various folders on your computer after installation. Advanced types of malwares have direct access to the operating system. Then it starts encrypting files and recording personal information. A malware detection process is created to detect malware. Malware detection is essential in the spread of malware over the internet as it acts as an early warning system to keep your computer safe from malware and cyberattacks. Keep hackers away from your computer and prevent your information from being compromised. But no method can find all malware in the real world. This shows that it is very hard to come up with effective ways to find malware and that there are big gaps in new research and methods.

Keywords

Malwares, Machine Learning, Deep Learning

1. INTRODUCTION

Malware detection is the process of looking for malware on your computer and in your files. It works well to find malware because it uses a variety of tools and methods. This is a very complicated process that goes both ways. Antivirus software simply looks for signatures in each programme to figure out if it contains malware (scanning). Malware Detection is like an early warning system for your PC that lets you know you're on a safe platform. Hackers can't get into your computer or share your personal information if you have malware detection. Before you can learn how to spot malware, you need to know what it is. Recent research shows that the number of mobile malware is growing. McAfee's Mobile Threat Report shows that mobile devices are getting a lot more backdoors, fake apps, and banking Trojans. Malware assaults against cryptocurrency, cloud computing, healthcare, social media, and the Internet of Things (IoT) are also on the rise [1, 2]. Malware must be identified for genuine people and organizations to be protected from it.

In the realm of cybersecurity, malware detection utilising machine learning and deep learning has grown in importance [15, 16]. Traditional signature-based techniques are no longer adequate to identify and stop malware attacks due to the rising amount and complexity of malware. As a result, methods for deep learning and machine learning have been developed to increase the precision and effectiveness of malware identification.

The procedures below are usually taken in order to identify malware using machine learning and deep learning:

- Data Collection: To train the machine learning or deep learning model, the initial step is to gather a sizable dataset of malware samples.
- Feature Extraction: The features from the malware samples must then be extracted as the next stage. These characteristics can be static (such file size, file format, and hash values) or dynamic (such as system calls and API calls).
- Model Training: The dataset of malware samples and their accompanying attributes serves as the training ground for the machine learning or deep learning model. Based on the retrieved attributes, the model develops the ability to discern between good and bad code. So, in current settings, ML and DL are utilised to detect malware in an effective and accurate method.

2. PROBLEM STATEMENT

The malware industry has grown so quickly that criminal groups have spent a lot of money on ways to get around traditional security measures, and anti-malware groups and communities have made stronger software to find and stop these attacks [3, 4]. Finding out if a file or piece of software is malware is the most important part of keeping your computer safe from malware attacks. As a result, we constructed a model to detect potential malware in this project and employed various machine learning and deep learning approaches to locate malware [5]. The fact that there are so many files and data that need to be examined for potentially harmful items is one of the main issues with anti-malware today. A real-time detection anti-malware software from Microsoft, for instance, is installed on more than 160 million PCs worldwide and scans more than 700 million systems each month. This amounts to tens of millions of data points per day that must be examined for malware. One of the main causes of the wide variety of files is that those who create malware have incorporated polymorphism to their dangerous components to make them harder to detect [6, 7]. This implies that dangerous files belonging to the same "family" of malware are constantly being altered to perform the same negative action and/or disguised in various ways. increase. You must be able to classify them into groups and determine which family they belong to in order to evaluate and sort through such a massive quantity of documents. This sort of grouping can also be used to determine whether newly downloaded files on your computer are a part of a specific family of harmful programmes.

3. DATASET

The dataset contains both the Train dataset and the Test dataset. The 400GB collection of data consists of both byte files and ASM files. ASM stands for "assembly language code file," and these files only have 0s and 1s in them. This file has keywords, registers, opcodes, and prefixes. From which some characteristics are taken. Bytes files have hexadecimal values, and if the value is between 00 and FF, the file is a byte file.

Training and Testing Dataset

We have downloaded the dataset (for training) for our work from www.kaggle.com.

4. WORKFLOW IMPLEMENTATION OF THE SYSTEM

The following workflow will be used in this work (refer figure 1):

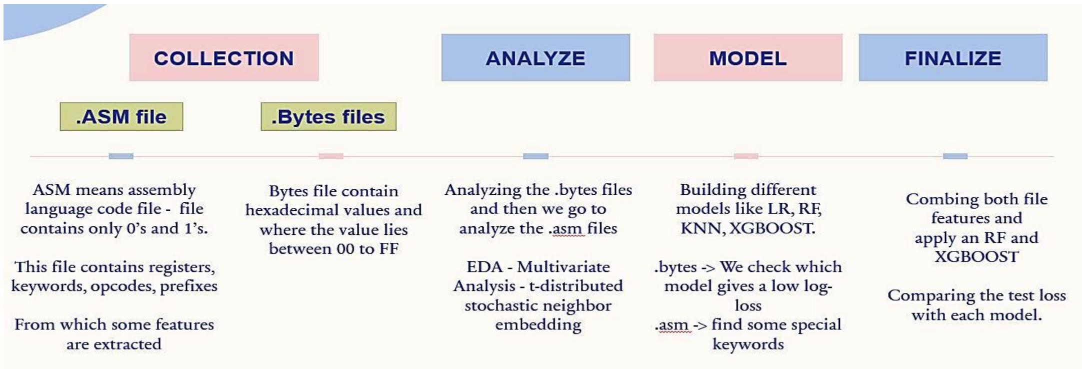

COLLECTION
ANALYZE
MODEL
FINALIZE
.ASM file
.Bytes files
ASM means assembly language code file - file contains only 0's and 1's.
This file contains registers, keywords, opcodes, prefixes
From which some features are extracted
Bytes file contain hexadecimal values and where the value lies between 00 to FF
Analyzing the .bytes files and then we go to analyze the .asm files
EDA - Multivariate Analysis - t-distributed stochastic neighbor embedding
Building different models like LR, RF, KNN, XGBOOST.
.bytes -> We check which model gives a low log-loss
.asm -> find some special keywords
Combing both file features and apply an RF and XGBOOST
Comparing the test loss with each model.

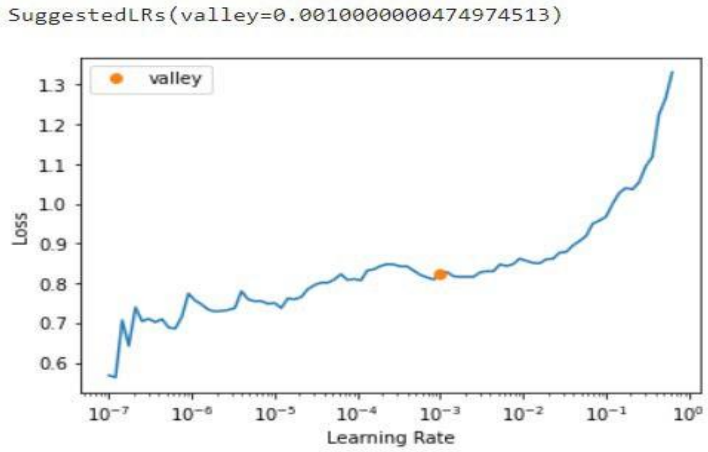
Fig. 1: Workflow of the system

Now the process of implementing the system is explained as:

- First, we imported the necessary libraries.
- We extract features from ASM files and byte files and store it's as a csv file we use this to train the model.
- One of the metrics we use to measure performance is multiclass log loss. We also use precision and recall metrics, which tell us where loss and learning rate are at their best.
- We used this model to look at the malware dataset:
    - Multilabel classifier

- ○ LightGBM
- ○ XGBoost
- ○ CatBoost

Multilabel Classifier:
Multi-label classification is a form of classification problem in machine learning where more than one label can be applied to each instance (see picture 2). A generalisation of multi-class classification is multi-label classification. In a single-label issue known as multiclass classification, instances are assigned to one of more than three classes. Labels in a multi-label problem don't have to be mutually exclusive, and an instance can belong to as many classes as it likes. When there are multiple classes and the data you wish to categorize may not belong to any class or to all classes at once, this is used.



```
SuggestedLRs(valley=0.0010000000474974513)
```

Figure 2: Multiple classifier

Stacking: Stacking [7, 8] will be performed on the train set using each model, and the predictedvalue will be made into new learning data.
LGBM: LightGBM is a framework for gradient boosting that uses algorithms for learning from trees. Used for ranking, sorting, and many other tasks related to machine learning.
XGBoost: The gradient boosting algorithm has been made better with the XgBoost algorithm [9, 10]. Its main objective was to make machine learning models run faster and perform better. In addition, the XGBoosting package offers a gradient boosting framework for several different languages, including R, Python, and Java.CatBoost: CatBoost is a decision tree algorithm that employs the gradient boosting technique. Fast, scalable, high-performance decision tree libraries that are used for ranking, classification, regression, and other machine learning tasks are created using gradient boosting. In addition to regression and classification, CatBoost can be used for ranking, recommendation mechanisms, forecasts, and even personal assistants. CatBoost is primarily used for the native processing of categorical features, fast GPU training, visualisation, and tools for model and feature analysis. It may also be used to quickly execute algorithms by employing forgetting or symmetric trees, to get over obstacles, and for other purposes. It is preferred over other gradient boosting techniques due to its capabilities. Concat is used to combine the models and forecast results.

5. MODELING

We use c code for the required model:

a. LGBM

```
[1]: model = LGBMClassifier(learning_rate= 0.025, n_estimators = 850,␣
        ↪min_child_weight = 1, boosting_type = "gbdt",␣
        ↪min_child_samples=68,random_state = 62,objective = "multi-class",metric =␣
        ↪"multi_logloss")
     model.fit(X,y)
     lgbm_pred = model.predict_proba(test.drop("Id",axis=1)) lgbm_pred = pd.DataFrame(lgbm_pred)
```

```
lgbm_pred.columns = ␣
  ↪["lgbm_Prediction1","lgbm_Prediction2","lgbm_Prediction3","lgbm_Prediction4","lgbm_Predict
  i
```

```
[2]: lgbm_pred.head()
```

```
[3]:    lgbm_Prediction1   lgbm_Prediction2   lgbm_Prediction3   lgbm_Prediction4   \
0       0.003597           0.000252           0.005012           0.002451
1       0.000348           0.000161           0.000223           0.001490
2       0.001270           0.001107           0.002490           0.005066
3       0.000655           0.001156           0.000284           0.000416
4       0.000005           0.000007           0.999875           0.000009

        lgbm_Prediction5   lgbm_Prediction6   lgbm_Prediction7   lgbm_Prediction8   \

0       0.980478           0.001998           0.001956           0.003985
1       0.994039           0.000552           0.001321           0.001779
2       0.823208           0.142466           0.012082           0.011727
3       0.006406           0.140785           0.000496           0.849639
4       0.000008           0.000032           0.000020           0.000030

        lgbm_Prediction9
0       0.000271

1       0.000087
2       0.000583
3       0.000164
4       0.000012
```

b. XGB

```
[4]: model = XGBClassifier(booster="gbtree",eta = 0.0975, min_child_weight = ␣
       ↪2,random_state = 62, objective = "multi:softproba",eval_metric="logloss") model.fit(X,y)
     xgb_pred = model.predict_proba(test.drop("Id",axis=1)) xgb_pred = pd.DataFrame(xgb_pred)
     xgb_pred.columns = ␣
       ↪["xgb_Prediction1","xgb_Prediction2","xgb_Prediction3","xgb_Prediction4","xgb_Predictio
       n5",
```

```
[5]: xgb_pred.head()
```

```
[6]:     xgb_Prediction1    xgb_Prediction2    xgb_Prediction3    xgb_Prediction4    \
0        0.018997           0.001637           0.005957           0.002948
1        0.003198           0.002105           0.000723           0.003809
2        0.003135           0.001500           0.000802           0.002041
3        0.001627           0.004977           0.000636           0.002888
4        0.000032           0.000035           0.999595           0.000033

         xgb_Prediction5    xgb_Prediction6    xgb_Prediction7    xgb_Prediction8    \

0        0.942978           0.007844           0.002205           0.016478
1        0.964947           0.006124           0.002711           0.015430
2        0.919743           0.042212           0.002658           0.026975
3        0.010097           0.232425           0.000726           0.744402
4        0.000044           0.000119           0.000045           0.000053

         xgb_Prediction9
0        0.000954

1        0.000953
2        0.000934
3        0.002221
4        0.000044
```

c. CBC

```
[7]: model = CatBoostClassifier(verbose=0) model.fit(X,y)
       cat_pred = model.predict_proba(test.drop("Id",axis=1)) cat_pred = pd.DataFrame(cat_pred)
       cat_pred.columns = ␣
         ↪["cat_Prediction1","cat_Prediction2","cat_Prediction3","cat_Prediction4","cat_Predictio
         n5",
```

```
[8]:     cat_Prediction1    cat_Prediction2    cat_Prediction3    cat_Prediction4    \
0        0.007862           0.000708           0.000061           0.000581
1        0.000318           0.000238           0.000015           0.000233
2        0.000994           0.000763           0.000064           0.001256
3        0.002163           0.003951           0.000097           0.005573
4        0.000006           0.000005           0.999622           0.000001

         cat_Prediction5    cat_Prediction6    cat_Prediction7    cat_Prediction8    \

0        0.971659           0.004077           0.000899           0.013638
1        0.995652           0.001113           0.000095           0.002232
2        0.981506           0.009225           0.000307           0.005662
3        0.013718           0.146473           0.000456           0.826299
4        0.000001           0.000123           0.000006           0.000221

         cat_Prediction9
```

Model Building - Multi Label Classifier

It is used when there are two or more classes and the data we want to classify may belong to noneof the classes or all of them at the same time [11, 12], e.g. to classify which traffic signs are contained on an image (refer figure 3).

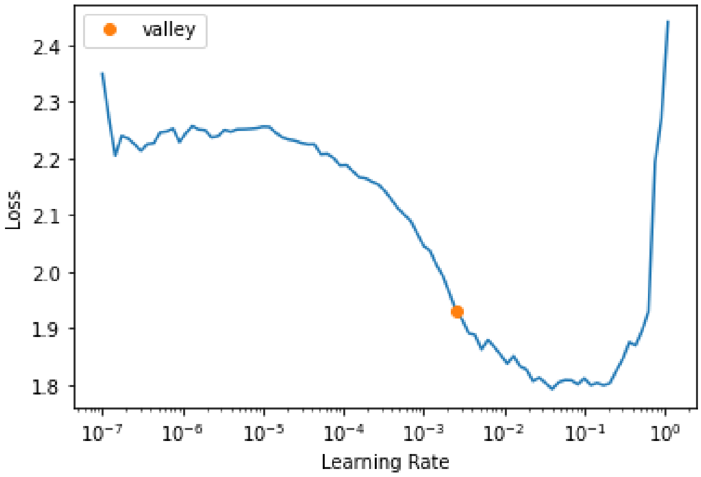


Figure 3: Loss with respect to learning rate

6. RESULTS AND DISCUSSION

The purpose of this research is to identify the type of malware in a given file by using Microsoft malware data to train the model [13,14]. The model is trained so that it can tell if a file is malware and what kind of malware it is. Below is the prediction values obtained by Multi label classifiers. Above is the prediction model concated with XGB, LGBM and CATBOOSTprovide the best model to detect malware instead of taking them as separate models for detecting malware. Further, readers are suggested to refer article [17-34] to know more about futuristic technologies and their importance to other useful sectors.

7. CONCLUSION

Since malware continues to constitute a serious danger to the security of computer systems and networks, malware detection is a crucial field of cybersecurity study. Deep learning and machine learning techniques have produced encouraging results in the identification and classification of malware. In conclusion, malware may be found and identified using machine learning and deep learning algorithms. These methods have the advantage of being able to learn from data, allowing them to improve over time and adapt to new kinds of malware. The requirement for significant amounts of labelled data, the existence of adversarial assaults, and the complexity of feature extraction are still issues that must be resolved. More study is required to keep enhancing the precision and efficacy of these techniques in the quickly evolving field of machine learning and deep learning for malware detection. So, the primary goal of this project was to identify the most accurate model for locating malware files based on the annual malware statistics given by Microsoft, which also enabled users to defend their systems from dangerous software and files.

REFERENCES

[1]. Pinhero, Anson, M. L. Anupama, P. Vinod, Corrado Aaron Visaggio, N. Aneesh, S. Abhijith, and S. AnanthaKrishnan. "Malware detection employed byvisualization and deep neural network." Computers & Security 105 (2021): 102247.

[2]. Ab Razak, Mohd Faizal, Nor Badrul Anuar, Rosli Salleh, and Ahmad Firdaus."The rise of "malware": Bibliometric analysis of malware study." Journal of Network and Computer Applications 75 (2016): 58-76.

[3]. Kolosnjaji, Bojan, Ambra Demontis, Battista Biggio, Davide Maiorca, Giorgio Giacinto, Claudia Eckert, and Fabio Roli. "Adversarial malware binaries: Evading deep learning for malware detection in executables." In 2018 26th European signal processing conference (EUSIPCO), pp. 533-537. IEEE,

2018.

[4]. Yuxin, Ding, and Zhu Siyi. "Malware detection based on deep learning algorithm." Neural Computing and Applications 31, no. 2 (2019): 461-472.

[5]. Yan, Jinpei, Yong Qi, and Qifan Rao. "Detecting malware with an ensemble method based on deep neural network." Security and Communication Networks2018 (2018).

[6]. Ronen, Royi, Marian Radu, Corina Feuerstein, Elad Yom-Tov, and MansourAhmadi. "Microsoft malware classification challenge." arXiv preprint arXiv:1802.10135 (2018).

[7]. Sharma, Sanjay, C. Rama Krishna, and Sanjay K. Sahay. "Detection of advanced malware by machine learning techniques." In Soft computing:Theories and applications, pp. 333-342. Springer, Singapore, 2019.

[8]. Kolosnjaji, Bojan, Ambra Demontis, Battista Biggio, Davide Maiorca, GiorgioGiacinto, Claudia Eckert, and Fabio Roli. "Adversarial malware binaries: Evading deep learning for malware detection in executables." In 2018 26th European signal processing conference (EUSIPCO), pp. 533-537. IEEE, 2018.

[9]. Watson, Michael R., Angelos K. Marnerides, Andreas Mauthe, and David Hutchison. "Malware detection in cloud computing infrastructures." IEEE Transactions on Dependable and Secure Computing 13, no. 2 (2015): 192-205.

[10]. Santos, Igor, Yoseba K. Penya, Jaime Devesa, and Pablo Garcia Bringas. "N-grams-based file signatures for malware detection." ICEIS (2) 9 (2009): 317-320.

[11]. Yin, Heng, Dawn Song, Manuel Egele, Christopher Kruegel, and Engin Kirda. "Panorama: capturing system-wide information flow for malware detection andanalysis." In Proceedings of the 14th ACM conference on Computer and communications security, pp. 116-127. 2007.

[12]. Narudin, Fairuz Amalina, Ali Feizollah, Nor Badrul Anuar, and Abdullah Gani."Evaluation of machine learning classifiers for mobile malware detection." SoftComputing 20, no. 1 (2016): 343-357.

[13]. Firdausi, Ivan, Alva Erwin, and Anto Satriyo Nugroho. "Analysis of machine learning techniques used in behavior-based malware detection." In 2010 second international conference on advances in computing, control, and telecommunication technologies, pp. 201-203. IEEE, 2010.

[14]. Kim, TaeGuen, et al. "A multimodal deep learning method for android malwaredetection using various features." IEEE Transactions on Information Forensics and Security 14.3 (2018): 773-788.

[15]. I. Mapanga, V. Kumar, W. Makondo, T. Kushboo, P. Kadebu and W. Chanda, "Design and implementation of an intrusion detection system using MLP-NN for MANET," 2017 IST-Africa Week Conference (IST-Africa), Windhoek, Namibia, pp. 1-12, 2017.

[16]. S. Rani, K. Tripathi, Y. Arora and A. Kumar, "Analysis of Anomaly detection of Malware using KNN," 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM), Gautam Buddha Nagar, India, 2022, pp. 774-779, doi: 10.1109/ICIPTM54933.2022.9754044.

[17]. Rekha, G., Tyagi, A.K., Anuradha, N. (2020). Integration of Fog Computing and Internet of Things: An Useful Overview. In: Singh, P., Kar, A., Singh, Y., Kolekar, M., Tanwar, S. (eds) Proceedings of ICRIC 2019 . Lecture Notes in Electrical Engineering, vol 597. Springer, Cham. https://doi.org/10.1007/978-3-030-29407-6_8

[18]. Tibrewal, I., Srivastava, M., Tyagi, A.K. (2022). Blockchain Technology for Securing Cyber-Infrastructure and Internet of Things Networks. In: Tyagi, A.K., Abraham, A., Kaklauskas, A. (eds) Intelligent Interactive Multimedia Systems for e-Healthcare Applications. Springer, Singapore. https://doi.org/10.1007/978-981-16-6542-4_17

[19]. Tyagi, Amit Kumar, Building a Smart and Sustainable Environment using Internet of Things (February 22, 2019). Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur - India, February 26-28, 2019. http://dx.doi.org/10.2139/ssrn.3356500

[20]. Tyagi, Amit Kumar and M, Shamila, Spy in the Crowd: How User's Privacy Is Getting Affected with the Integration of Internet of Thing's Devices (March 20, 2019). Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur - India, February 26-28, 2019.

[21]. Amit Kumar Tyagi, N. Sreenath, “Preserving Location Privacy in Location Based Services against Sybil Attacks”, International Journal of Security and Its Applications (ISSN: 1738-9976 (Print), ISSN: 2207-9629 (Online)), Volume 9, No.12, pp.189-210, December 2015.

[22]. Gillala Rekha, Amit Kumar Tyagi, and V. Krishna Reddy, “A Wide Scale Classification of Class Imbalance Problem and its Solutions: A Systematic Literature Review”, Journal of Computer Science, Vol.15, No. 7, 2019, ISSN Print: 1549-3636, pp. 886-929.

[23]. A. K. Tyagi, T. F. Fernandez and S. U. Aswathy, "Blockchain and Aadhaar based Electronic Voting System," 2020 4th International Conference on Electronics, Communication and Aerospace

Technology (ICECA), Coimbatore, India, 2020, pp. 498-504, doi: 10.1109/ICECA49313.2020.9297655.

[24]. S. U. Aswathy, Amit Kumar Tyagi, Shabnam Kumari, “The Future of Edge Computing with Blockchain Technology: Possibility of Threats, Opportunities and Challenges”, in the Book "Recent Trends in Blockchain for Information Systems Security and Privacy", CRC Press, 2021.

[25]. Rekha, G., Tyagi, Amit Kumar, and Krishna Reddy, V. ‘Solving Class Imbalance Problem Using Bagging, Boosting Techniques, with and Without Using Noise Filtering Method’. 1 Jan. 2019 : 67 – 76.

[26]. B. Gudeti, S. Mishra, S. Malik, T. F. Fernandez, A. K. Tyagi and S. Kumari, "A Novel Approach to Predict Chronic Kidney Disease using Machine Learning Algorithms," 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2020, pp. 1630-1635, doi: 10.1109/ICECA49313.2020.9297392.

[27]. Amit Kumar Tyagi, Meenu Gupta, Aswathy SU, Chetanya Ved, “Healthcare Solutions for Smart Era: An Useful Explanation from User’s Perspective”, in the Book "Recent Trends in Blockchain for Information Systems Security and Privacy", CRC Press, 2021.

[28]. Tyagi, A.K., Kumari, S., Fernandez, T.F., Aravindan, C. (2020). P3 Block: Privacy Preserved, Trusted Smart Parking Allotment for Future Vehicles of Tomorrow. In: , et al. Computational Science and Its Applications – ICCSA 2020. ICCSA 2020. Lecture Notes in Computer Science(), vol 12254. Springer, Cham. https://doi.org/10.1007/978-3-030-58817-5_56

[29]. Kumar, A., Tyagi, A.K., & Tyagi, S.K. (2014). Data Mining: Various Issues and Challenges for Future A Short discussion on Data Mining issues for future work.

[30]. Amit Kumar Tyagi, G. Rekha, “Challenges of Applying Deep Learning in Real-World Applications”, Book: Challenges and Applications for Implementing Machine Learning in Computer Vision, IGI Global 2020, p. 92-118. DOI: 10.4018/978-1-7998-0182-5.ch004

[31]. Amit Kumar Tyagi, N. Sreenath, Cyber Physical Systems: Analyses, challenges and possible solutions, Internet of Things and Cyber-Physical Systems, Volume 1, 2021,Pages 22-33,ISSN 2667-3452,https://doi.org/10.1016/j.iotcps.2021.12.002.

[32]. Nair, Meghna Manoj; Tyagi, Amit Kumar "Privacy: History, Statistics, Policy, Laws, Preservation and Threat Analysis", Journal of Information Assurance & Security . 2021, Vol. 16 Issue 1, p24-34. 11p.

[33]. S. Mishra and A. K. Tyagi, "Intrusion Detection in Internet of Things (IoTs) Based Applications using Blockchain Technolgy," 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2019, pp. 123-128, doi: 10.1109/I-SMAC47947.2019.9032557.

[34]. M. Shamila, K. Vinuthna and A. K. Tyagi, "A Review on Several Critical Issues and Challenges in IoT based e-Healthcare System," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), Madurai, India, 2019, pp. 1036-1043, doi: 10.1109/ICCS45141.2019.9065831.

[35]. S. Chaudhary et al., “Design and development of gesture based gaming console,”, RJET. A & V Publications, 12(2), pp. 51-56, 2021 [doi:10.52711/2321-581X.2021.00009].

[36]. A. Jatain et al., “Cloud storage architecture: Issues, challenges and opportunities. International Journal of Innovative Research in Computer Science & Technology (IJIRCST) ISSN,” vol. 5552, p. 2347, 2021.

[37]. T. Joy et al., “Computer vision for color detection,” Int. J. Innov. Res. Comput. Sci. Technol. ISSN, vol. 5552, p. 2347, 2021.