Selected Computing Research Papers

Volume 10

June 2021

Dr. S. Kendal (editor)

Published by
the
University of Sunderland

Edited, typeset and printed by
Dr. S Kendal
University of Sunderland
David Goldman Informatics Centre
St Peters Campus
Sunderland
SR6 0DD

Tel: +44 191 515 2756
Fax: +44 191 515 2781

Contents Page

Critical Analysis of Current Automated Scalable Application Deployment on Cloud

Adam Felner

Abstract

Cloud applications offers a lot of advantages for rapid elasticity, redundancy and flexibility. Multiple solutions exist to deploy the applications on Cloud, although a lot of them offer only limited usability. In this paper, three different methods and approaches of scalable automatic deployment such as legacy and containerized component applications with DevOps approach, are evaluated and analysed. The purpose was to investigate advantages and disadvantages of each approach. Conclusions and recommendations were made through this paper regarding comparison, future work and real-world usage of the optimal use case for the method.

1 Introduction

Cloud computing is a phenomenon of the last 10 years however, to fully use the potential of cloud computing, there is a very difficult challenge. Automated deployment of applications on the Cloud is one of the main goals of DevOps teams all around the world. The scalability issue is what makes the goal even more problematic. Approaching the same goal with different methodologies often leads to inconsistency (Herden S. et.al., 2010). It is crucial to have one approach applicable to the highest usage whilst still being able to offer a scalable solution.

Herden S. et.al. (2010) investigate approaches to automate the deployment with scalable applications on the Cloud by declaring 7 principles. Through producing architectural specification, they conducted implementation which fulfilled the expectations towards scalable deployment model of applications.

The principles declared by Herden S. et.al. (2010) are the backbone for all current methods, however, the approach can vary in multiple ways.

Hao Wei, Joaquin Salvachua Rodriguez (2018) proposed a policy-based approach, which is nowadays still just in an early phase of development, and there is more work to be done. However, in theory, it offers more dimensions of expansion.

Ehrlich M. et.al. (2019) have implemented and showed the usage of multiple frameworks inside of the automated application deployment with a focus on scalability and flexibility, thanks to the Tosca framework profile with Ansible playbooks. Extending the base model of Tosca to reach the needed specification profile in Yaml allows almost perfect expansion in all dimensions.

This survey paper will evaluate current research papers focused on improved scalability of automated applications deployment with multiple frameworks and multiple Cloud providers, based on their experiments and outcomes. That will have a significant impact on the community dealing with the subject of automated scalable deployments, where it will offer a brief understanding of current possibilities and the way further research should lead.

2 Current methods of application deployment on Cloud

The section reviews multiple approaches and methods used in application deployment with legacy frameworks, containerized, DevOps solutions, and deployment of component-based applications carried out by different researchers. Methods and validity of the experiments as well as the implementation in the real world will be discussed in this paper.

2.1 Legacy application deployment method

Sagar N.Deshmukh and Khandagale H.P.(2017) proposed a method based on a master-slave model of servers, where the application is being deployed. However, their method is tight up together with SaltStack technology, which can get a better functionality of multiple servers than the traditional master-slave model.

SaltStack technology, based on the master server, creates a controller, which can deploy and provision applications on multiple pool servers. As Sagar N.Deshmukh and Khandagale H.P.(2017) show, there is a simple architecture showing this approach with other server pools implemented as the SaltStack is able to control multiple different environments.

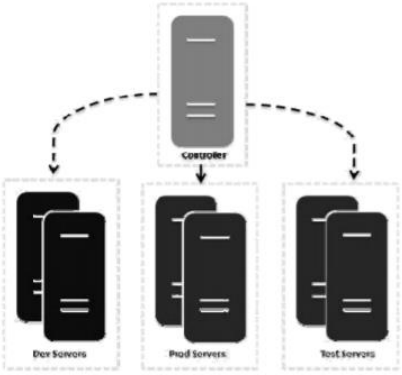


Figure 1 Different types of environments (Sagar N.Deshmukh and Khandagale H.P. 2017)

The whole setup is configured through formula in vice or state files saved in the pillar method to ensure easy data flow and secured data for formula creation of the new environment. (Sagar N.Deshmukh and Khandagale H.P.2017)

The formula to install an apache server in the experiment on the slave machine in the server pool is shown in Figure 2.

```
install-apache:
  pkg.installed:
    - name: apache2

run apache:
  service.running:
    - name: apache2
```

Figure 2 Installation example (Sagar N.Deshmukh and Khandagale H.P.2017)

The installation will be provided from the pkg file, the service will start after the installation.

With this approach and SaltStack technology, they claim to provide scalable and flexible applications on a bigger scale with 90% savings in time to set up and provide easy and secure data flow as Sagar N.Deshmukh and Khandagale H.P.(2017) concluded. In the future, they also plan to extend to applications as Tomcat or Ruby on Rails what can help to improve the deployment procedure.

The authors of the solution show the structure of the SugarCRM system, however, there are multiple factors, which could have an impact on the validity of claims, as bandwidth of network or configuration differentiation, which are not provided in this paper. Therefore, there is a need for more work to gain more measurement and configuration data to fully justify the method.

Göttsche M. et.al. (2015) proposed a method with 3 layers method, which can be implemented in several ways as they are showing. Based on a framework structure build upon execution layer with 3 steps as provision, configure, and application management, with the usage of the cluster for backbone framework and cluster for the deployed application.

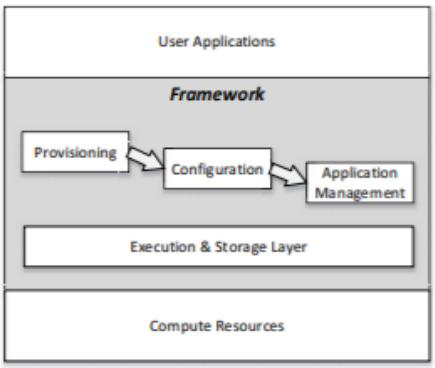


Figure 3 Architecture layers (Göttsche M. et.al.2015)

However, they further work with solutions is based on the Hadoop framework with Vagrant and Ansible. Furthermore, they use python scripts to enable the needed functionality thanks to multiple APIs. They use OpenStack Cloud with 19 nodes, and for each of them, 2vCPUs and 4GB of RAM. From the 19 nodes 3 nodes will be management nodes for Hadoop, and 16 nodes will run the deployed application.

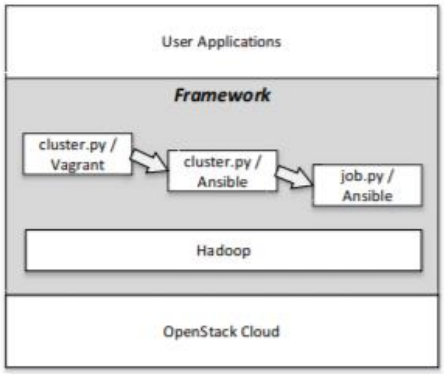


Figure 4 Implementation Layers (Göttsche M. et.al. 2015)

They further describe and provide a test with complete measurements and show a comparison of the proposed method and sequential deployment, which is semi-automated using only a MapReduce application.

Göttsche M. et.al. (2015) show the benefit in the reduction of deployment time in a scalable application using the proposed framework against the native MapReduce on average by 1.15 speedup with multiple tests runs and a lot of side factors mitigated. They also conclude a demonstrated feasibility of the approach. They mention a few issues to enable the full potential of the framework as ties to OpenStack.

Göttsche M. et.al. (2015) show the method in a very detailed form for the implementation in the real world with comparison and test against the native solution. They have provided measurement tables with deployment times compared with the native solution and deployment details needed to prove the validity and repeatability of the tests while mitigating bias by multiple tests being done and clear and structured configuration. Their simulation was in place and showed a clear difference in the same discipline with still counting on multiple aspects. Therefore, the results are justified.

Sagar N.Deshmukh and Khandagale H.P.(2017) showed a simple method, which can be also extended to the level of the second method of Göttsche M. et.al. (2015). However, Sagar N.Deshmukh and Khandagale H.P.(2017) showed weakness to justify the concluded reduction of time by 90% due to missing measurements data of deployment time. Therefore, the results are missing more proof to be concluded as valid. Göttsche M. et.al. (2015) have proposed an extended version with the detail needed as exact configuration with setup details in discipline to evaluate and compare the results.

To conclude, further research should be done by improving the method proposed by Göttsche M. et.al. (2015). Research by them was done in a proper way with sufficient detail and control to be concluded as trustworthy and valid to further development of the method. The method also shows a potential that needs to be unleashed by taking the ties of OpenStack out of the method.

2.2 Containerized solutions

Satish Narayana Srirama et.al. (2020) proposed a method based on a containerized platform for microservices to deploy applications on the Cloud. They show contribution in their method by 4 stages.

The first contribution helps to minimize the cost of running containers by finding the best resource requirements match for microservices. The second contribution is to reuse warm containers, to minimise the deployment time and overall processing cost. The third contribution is to deploy containers on suitable physical machines or virtual machines based on the availability of the resources

to utilise the computing resources efficiently. Figure 5 is showing the average number of active usage of resources.

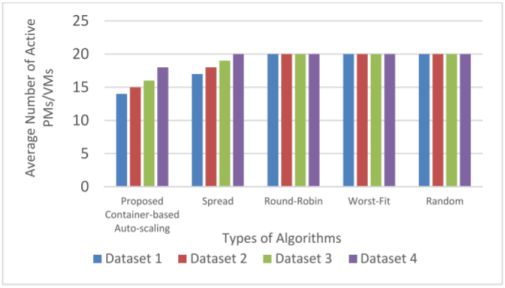


Figure 5 Average number of active physical machines and virtual machines (Satish Narayana Srirama et.al. 2020)

The fourth contribution, based on the availability of the resources and requested resources, is an auto-scale function, which is helping to reduce overall resource wastage.

For the test, Satish Narayana Srirama et.al. (2020) used multiple performance matrices over different real-time Google Cluster traces. They have also done the same tests on the current Docker Swarm strategy. For the evaluation of the proposed method, they compared the results.

The results have shown, the proposed deployment container-based auto-scaling method minimized 12-20% processing cost of the microservices, whilst maximizing the efficient usage of resource by 9-15% and 10-18% respectively, over Docker Swarm strategy.

Based on the results they expressed, the method offers a great fit as a backbone framework for the deployment of the next containerized solutions due to a high efficiency, Satish Narayana Srirama et.al. (2020). However, they have also shown the current limitation of the method tied to Google Cluster Trace Logs provided by Google Cloud.

Satish Narayana Srirama et.al. (2020) experimented using multiple matrices to test and prove the validity of the results. They provided an adequate amount of data to justify the results, and proving the claims by the researchers, which are clear and validated with control over bias, which might happen during the testing multiple application matrices.

Zahra Nikdel et. al. (2017) proposed a solution DockerSim based on iCanCloud, as a buildup layer, such as Docker Engine layer, while modifying existing layers of the iCanCloud. By the addition of the Container Repository layer and Docker engine layer to a model of iCanCloud helped DockerSim to support deployment regimes as within VMs, within containers within VMs, within server-hosted containers and applications spanning any combination of previously named. Figure 6 shows layered model of DockerSim.

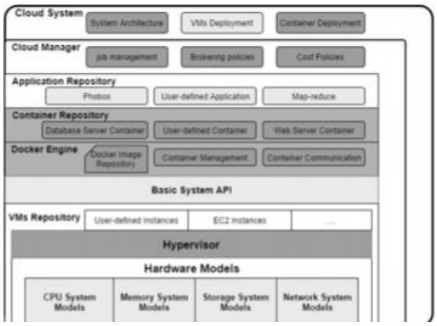


Figure 6 DockerSim Architecture Model (Zahra Nikdel et.al. 2017)

The experiments taken by Zahra Nikdel et. al. (2017) included building a webserver deployed within its own Docker Container for every instance.

They have used computing resources such as Intel Core i5 with 8GBytes of internal memory. Testing was divided into 2 scenarios, whilst the first scenario was simulating the 15,000 requests per hour requiring 10,000MI to service with the input of 1,246KBytes.The second scenario involved a simulation of 100 containers deployed web server, with varying request workload from 0 to 100,000 per hour, with each request size of input 1,246KBytes.

The results obtained in both experiment scenarios showed clear degradation of DockerSim's performance with a richer SaaS system but the ability to achieve a sub 1:1 wall time ratio proofs the ability of DockerSim to support larger-scale more complex scenarios.

Based on their experiments and results, Zahra Nikdel et.al. (2017) concluded that DockerSim has proved the ability to provide: i) a full container deployment and behavioural layer, ii) full packet-level network and protocol behaviours, iii) full multi-layer OS process scheduling behaviours, and iv) a generic queuing network approach to

modelling application layer SaaS deployment behaviours.

Research by Zahra Nikdel et.al. (2017) and experiment data provided are clear and with data measurements and configuration of tests and deployment are valid and justified. Although, there is a very small spectrum of usage in the real world of application as they provided in the experiment. The claims are clear, however, are showing more scalability and flexibility qualitative possibilities than the comparable quantitative results.

As the containerized solutions needs to be flexible in many dimensions, there is more space for usage for Satish Narayana Srirama et.al.(2020) with their contributions to real-world solutions. Other than that, method Zahra Nikdel et.al.(2017) offers a lot of similar, though there is a need to get more experiment data with other kinds of applications as webservers as only provided in their research.

2.3 DevOps approaches

Research done by Wettinger J. et.al. (2018) have shown an approach towards application deployment with middle-ware and application deployment type.

As they say, it is the key enabler in current Cloud deployment, however, there is nothing specific to have a broad function across multiple applications and Cloud providers. Their model is based on the systematic classification of existing application deployment methods; however, they expanded the level of usage of deployment to the level of middleware. This approach allowed them to configure deployment for better reusability, portability, and flexibility.

For their experiments, Wettinger J. et.al. (2018) developed deployment plans of three applications (Taxi App, SugarCRM, and Chat App), which have different requirements for deployment. They deployed the apps with multiple Cloud providers and tested them in qualitative and quantitative dimensions.

The results obtained in the experiments showed improvement in deployment time as still having better reusability, portability, and flexibility as with native application deployment.

Based on the experiments and their results, they concluded 6 lessons learned statements, which are showing the guideline for the deployment of applications. As Wettinger J. et.al. (2018) stated middle-ware application deployment is not suitable for all purposes, such as TaxiApp or SugarCRM, however, it is possible to realise a hybrid solution. Middle-ware deployment best fit for scenarios where conventions for application components are already established, as it can create a more complex deployment plan, although not affect the total execution time or deployment scenario.

Middle-ware application deployment research by Wettinger J. et.al.(2018) provided a test on the application from real-world usage, with qualitative and quantitative measurements to justify the claims. Tests are valid and justified with data provided from multiple application usage and controlled bias by the utilisation of several cloud providers.

Mitesh Soni (2015) proposed a method of end to end automation of application automation lifecycle, with deployment being one part of the process (See Fig.7). The method is building upon the DevOps approach and knowledge and is using Chef as configuration management with deployment plugins or by Shell script. As he said, this method is an outcome of the agility of business and IT collaboration.

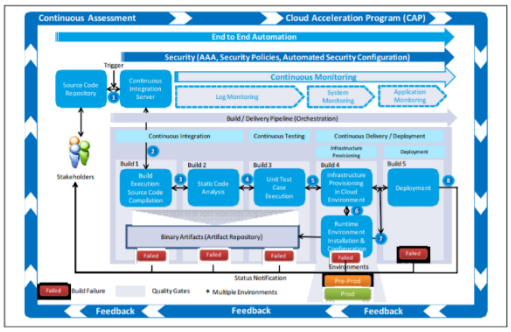


Figure 7 End-to-End Automation (Mitesh Soni 2015)

For test purposes, the Proof-Of-Concept (PoC) solution has been created. He included multiple technologies and methods, such as integration of CI server and Source Code repository, build pipeline for orchestration and quality gates creation, application assessment for the Cloud environment- feasibility analysis or infrastructure provisioning using configuration management tool in cloud environment.

The implemented PoC solution showcased various benefits of end to end automation in application deployment and full delivery lifecycle. Automated deployment with standardized production environments, rapid delivery, automated infrastructure provisioning in the Cloud with integration and configuration across the public, private and hybrid cloud, monitoring of the health of applications.

Based on the results of the implementation of PoC, Mitesh Soni (2015) concluded method of End-to-End automation of application lifecycle using the DevOps knowledge with Cloud technologies gives multiple benefits of application deployment and whole lifecycle, which will result in the running cost and development cost of the solution.

The experiment by implanted PoC solution is valid and justified as it shows qualitative claims to be real in real-world application deployment. The reduction of bias was controlled by multiple tests with different technologies included.

DevOps community has a big impact on current application deployment development, and the research done by Mitesh Soni (2015) showed more usage and bigger application in the present world. The claims and knowledge gained by Wettinger J. et.al. (2018) can be used to further develop the PoC solution of Mitesh Soni (2015).

2.4 Component-based applications deployment

Sébastien Lacour et.al. (2005) proposed a method of deployment component-based applications based on a generic application model (GADE).

The GADE model allowed the deployment of various applications using a common planer. They followed up by the creation of prototype ADAGE, which integrated the GADE model to be able to automatically deploy component-based applications using a common planner.

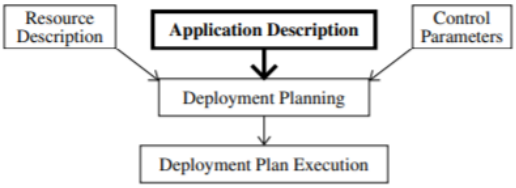


Figure 8 Overview of process to deploy component-based application (Sébastien Lacour et.al. 2005)

For the experiments, Sébastien Lacour et.al. (2005) deployed one application with several methods, such as Corba Component Model or Condor-G.

The results obtained showed better utilization of resources with transparency and reusability of the common planner for various application deployments.

Sébastien Lacour et.al. (2005), based on their research, concluded that the ADAGE prototype is capable of automatically deploying the component-based applications using the common planer. However, they also claim there is a need to understand how application-specific considerations should be handled to reduce the configuration phase.

The prototype method by Sébastien Lacour et.al. (2005) showed multiple qualitative results, which are justified by a controlled experiment. However, there is not control of bias due to the usage of only one application to deploy. There is more work needed to be done to justify the results with multiple application deployments.

Although Tao Shi et.al. (2020) proposed a method based on GA-algorithm to select VMs, even from different Clouds, to even more reduce the cost. This approach also needed them to design a new hidden clustering method to utilize the VMs. These changes with control of computational time resulted in the H-GA method.

The experiments of Tao Shi et.al. (2020), have been provided on multiple Cloud providers, network environments, and business applications in real-world simulations collected multiple datasets, such as performance utilization, scaling time, deployment time, cost.

The data collected from the experiments showed that the H-GA method can achieve up to 8% performance improvement at the same budget with comparison to the current methods.

Tao Shi et.al. (2020) claim that the solution can offer a reduction of cost in multiple component applications, however, they also stated that the current H-GA method would not be efficient on the containerized component applications. Also, it can

cause multiple issues and difficulties, such as solution space and computational difficulty what results in a higher cost.

The experiments by Tao Shi et.al. (2020) are controlled and collected quantitative data, with various factors to justify the results. The claims are built from experimental data and showed the usage in a real-world application. The conclusion is valid.

As the method of Sébastien Lacour et.al. (2005) does not provide enough test data on multiple applications, the method of Tao Shi et.al. (2020) currently shows a better deployment method, with mentioned drawbacks as not suitability for containerized deployments.

3 Conclusions

The majority of current application deployment methods evaluated in this paper had a strong prospect of successful results and well-presented tests but for some, there was a need for more work to provide proof of the results with evidence to justify it. The methods are solving current applications for the need of scalable cloud deployment.

Göttsche M. et.al. (2015) proposed a method based on the 3 layers method, which gives a certain level of scalability, however within the larger deployments may come up to limitations of the solution by design.

Therefore, the other solution proposed by Satish Narayana Srirama et.al. (2020) offers a better model to support a larger deployment model with planning due to the containerized approach.

However, Tao Shi et.al. (2020) mentioned in their method a valid point to develop the method of deployment of applications, however with disadvantages of not suitability for containerized solutions.

Although, the DevOps approach of Mitesh Soni (2015) with Wettinger J. et.al. (2018) showed the most suitable solutions, which are scalable even to large deployment with having the flexibility as with very small solution.

Generally, looking at all the methods and approaches mentioned in this paper, the most promising solution for real-world scalable automatic deployment of applications on the Cloud is the DevOps approach with a combination of containerized solutions. However, there is still missing a few crucial points on how to scale containers based on their needs with exact utilization and suitability of applications to deploy as containerized solutions.

References

Ehrlich M., Trsek H., Gergeleit M., Paffrath J., Simkin K., Jasperneite J., 2019, 'Secure and Flexible Deployment of Industrial Applications inside Cloud-Based Environments',*2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA),* pages 1269-1272

Hao Wei, Joaquin Salvachua Rodriguez, 2018, 'A Policy Based Application Deployment Method in Hybrid Cloud Environment',*IEEE 6th International Conference on Future Internet of Things and Cloud,* pages 93, 95, 98

Herden S.,Zwanziger A.,Robinson P.,2010, 'Declarative Application Deployment and Change Management',*International Conference on Network and Service Management – CNSM,* pages 131-133

Göttsche M.,Fabian Glaser,Steffen Herbold,Jens Grabowski,2015, 'Automated Deployment and Parallel Execution of Legacy Applications in Cloud Environments',*IEEE 8th International Conference on Service-Oriented Computing and Applications (SOCA)* ,Vol. 4

Mitesh Soni, 2015, 'End to End Automation On Cloud with Build Pipeline: The case for DevOps in Insurance Industry',*IEEE International Conference on Cloud Computing in Emerging Markets,* pages 85-89

Satish Narayana Srirama, Adhikari M., Souvik P.,2020, 'Application deployment using containers with auto-scaling for microservices in cloud environment', *Journal of Network and Computer Applications,* Vol. 160

Sagar N.Deshmukh and Khandagale H.P. ,2017, 'A system for application deployment automation

on cloud environment', *Innovations in Power and Advanced Computing Technologies (i-PACT)*, Vol. 1

Sébastien Lacour,Christian Pérez,Thierry Priol,2005, 'Generic Application Description Model: Toward Automatic Deployment of Applications on Computational Grids', *The 6th IEEE/ACM International Workshop on Grid Computing,* pages 284-288

Tao Shi ,Hui Ma ,Gang Chen,Sven Hartmann,2020, 'Location-Aware and Budget-Constrained Service Deployment for Composite Applications in Multi-Cloud Environment', *IEEE Transactions On Parallel And Distributed Systems, Vol. 31, No. 8, pages 1954-1969*

Wettinger J.,Andrikopoulos V.,Leymann F.,Strauch S.,2018, 'Middleware-Oriented Deployment Automation for Cloud Applications', *IEEE Transactions On Cloud Computing,Vol. 6,No. 4*

Zahra Nikdel,Bing Gao,Stephen W.Neville, 2017, 'DockerSim:Full-stack simulation of container-based Software-as-a-Service (SaaS) cloud deployments and environments', *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing(PACRIM)*

Analysis of Distributed Ledger Consensus Protocols for Improving Transactional Throughput and Scalability

James Hastie

Abstract

This research paper evaluated various proposed alternative consensus protocols, commonly used in Distributed Ledger Technology (DLT), focusing on transactional throughput and scalability compared to current DLT consensus protocols. These comparisons highlighted various improvements in the proposed consensus protocols whilst revealing potential limitations of transactional throughput and scalability, as well as challenges when making direct comparisons between protocols. Finally, a conclusion was given based on the proposed consensus protocols, aiming to providing a discussion on future research opportunities drawn from the research paper findings.

1 Introduction

With the advent of DLT in 2008, commonly referred to as blockchain, its' use cases have diversified to encompass a wide range of markets from cryptocurrencies to supply chains.

Distributed Ledgers (DL) are naturally decentralised, consisting of various nodes separated across geographic areas, each having a replication of the ledger and providing consensus for any updates to the DL (He et al. 2018).

Distributed Ledgers implement consensus protocols to validate transactions, however, these protocols vary in efficiency, such as power usage, scalability and throughput. The Proof-of-Work (PoW) protocol, implemented by Bitcoin, has low transactional throughput (reaching a maximum of 7 transactions per second) and high computational power requirements to verify transactions (Mahmoud, Q. and Shahriar Hazari, S., 2020), limiting its effectiveness for large scale applications such as commercial or financial services.

Hassani et al. (2018) pay particular focus on DLT and its potential for use in Big Data and artificial intelligence, stating current protocol research has been focused on alleviating power usage and transactional throughput to increase viability and application usage.

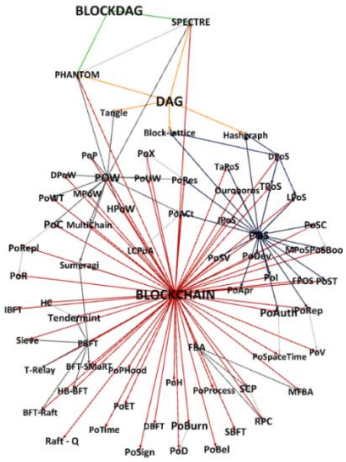


Figure 2: Relationship of various consensus protocols with different architecture-based categories of DLTs. (Hewage et al. 2019)

Various consensus protocols exist, each aiming to address a specific use case or solve limitations of other protocols. Figure 2 shows their relationships based on their DLT architecture, many of these protocols share similar characteristics or evolution from one another (Hewage et al. 2019).

This research paper aims to critically evaluate select proposed consensus models as potential viable alternatives to traditional blockchain DLT, focusing on their suitability to address low transactional throughput and scalability of PoW protocols.

2 Alternative DLT Consensus Protocols

This section discusses select consensus protocols proposed by researchers, specifically CoDAG, Bitcoin-NG and RapidChain. These protocols focus on addressing some or all of the deficiencies of existing protocols by use of alternative protocol methods and design.

2.1 Compacted Directed Acyclic Graph (CoDAG)

Chen et al. (2019) propose a consensus protocol which utilises a Compacted Directed Acyclic Graph (CoDAG) restricted and organised level and width structure, as shown in Figure 3, limiting the number of blocks able to be generated in a round by the system, aiming to improve concurrency and transactional throughput.

Directed Acyclic Graph (DAG) utilise nodes for storing transactions as the foundation, allowing for parallelism and a cooperative consensus model, versus competitive PoW consensus models (Bamakan et al. 2020), with Haq et al. (2018) stating in their research on DAG that it also provides scalability by shifting handling of transactions on to their owner, as well as suitability for microtransactions.

Chen et al. (2019) claim that by restricting the width of the DAG it maintains greater connectivity between honest blocks, improving security against dishonest attacks. They note that DAG width can be altered to provide greater transactional throughput allowing greater flexibility depending on intended use.

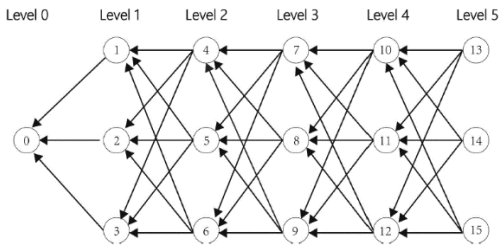


Figure 3: Channel node structure of CoDAG (Chen et al. 2019)

Figure 4 shows the relative throughput against Bitcoin and Etherium based on DAG width and block size, showing that whilst Transactions per Second (TpS) does increase with width they appear equally affected by block size.

Compared to Bitcoin or Etherium, CoDAG claims to achieve 56x and 26x throughput respectively, as well as doubling its own throughput at a width of 20 compared to a width of 1 (Chen et al. 2019).

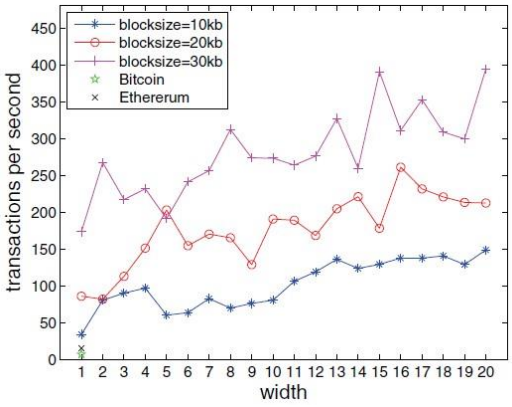


Figure 4: CoDAG width and block size throughput (Chen et al. 2019)

Chen et al. (2019) tested level formation time with a block size of 20kb as a basis for TpS, as shown in Figure 5. Whilst it would be proposed that a decrease in level formation time would correlate to increased TpS, their findings show mixed results.

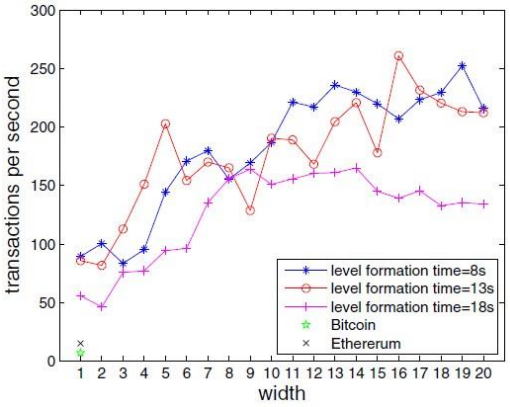


Figure 5: CoDAG level formation time (Chen et al. 2019)

Examination of these results show an increase in TpS based on increase of DAG width, however it does not always scale linearly. The researchers state this is due to saturation of computational

nodes used, limiting the number of transactions possible for each level formation time (Chen et al. 2019). The authors did not provide details on their nodes, making suggestions for improvement of node structure to overcome this limitation difficult, or whether this limitation is intrinsic in the protocol.

Based on results shown for correlation of block size to width, it would prove beneficial for further research with greater widths and block sizes which might show the optimum for each, provide a greater understanding of the limitations of CoDAG and suggest areas for improvement.

Whilst results shown by Chen et al. (2019) are promising, providing evidence their claimed improvements are valid, the researchers did not provide information on test system hardware used or relative power usage, only that they utilised the open-source code of the official Go implementation of Ethereum, making their results challenging to corroborate, reproduce or fully evaluate.

Additionally, no other research was found that evaluated CoDAG, whilst the original researchers proposed protocol claims to improve on other consensus protocols, it has not yet been corroborated by other researchers.

Overall, CoDAG appears to offer improved TpS and scalability compared to the Bitcoin consensus protocol, allowing granular control over the design of the protocol, providing greater flexibility in applying it to applications. Additional testing at greater widths and block sizes would be required to more clearly understand if CoDAG can provide additional increase in TpS and scalability.

2.2 Bitcoin-NG

Eyal et al. (2016) propose Bitcoin-NG, intended as an alternative blockchain protocol which is designed to build upon the Bitcoin blockchain protocol, addressing its scalability and transactional throughput limitations.

The authors claim that by separating blockchain events into leader election and transaction serialisation by use of key blocks and microblocks, as shown in Figure 6, this reduces the bandwidth required to transmit blocks throughout the system, as well as allowing transactions to take place whilst a new leader is elected, reducing latency (Eyal et al. 2016).

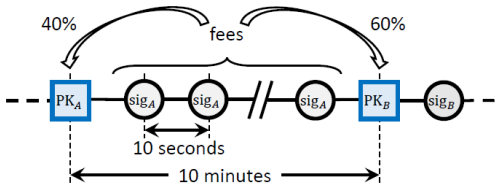


Figure 6: Bitcoin-NG chain structure, showing key blocks (square) and microblocks (circle) (Eyal et al. 2016)

Eyal et al. (2016) conducted latency and block size tests to evaluate the proposed benefits of the Bitcoin-NG protocol, Figure 7 shows results of the latency tests. For each frequency of Bitcoin block size, the corresponding microblock size was chosen such that payload throughput of each protocol was identical.

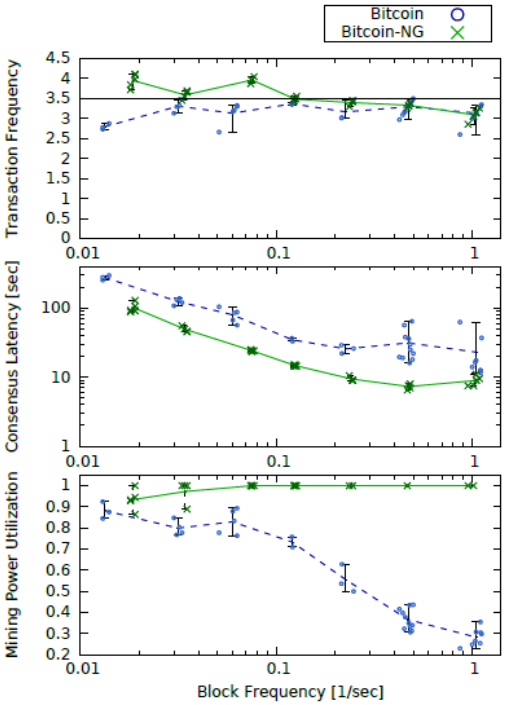


Figure 7: Bitcoin-NG latency reduction test results (Eyal et al. 2016)

The authors claim in the transaction frequency and consensus latency results that, whilst both protocols are similar, the mining power utilisation chart highlights a difference between them. By increasing block frequency/generation Bitcoin-

NG appears to maintain and improve on this metric.

The authors results indicate that as block generation is increased, the rate of discarded blocks increases for some nodes, with only the fastest nodes blocks being accepted as forks become more frequent and require resolution.

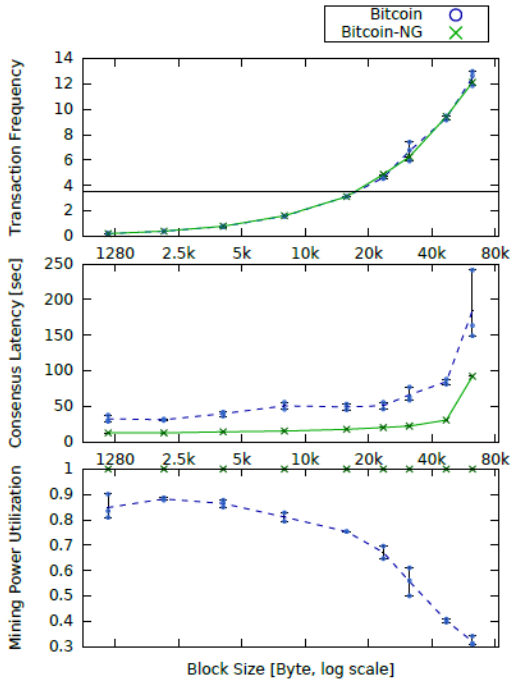


Figure 8: Bitcoin-NG block size test results (Eyal et al. 2016)

Testing of block size was conducted by Eyal et al. (2016), Figure 8 shows comparison between Bitcoin and Bitcoin-NG protocols.

Latency and block size tests claim that whilst both protocols share some characteristics, separation of blockchain transactions and leader election provide notable improvement in scalability, primarily power utilisation, it can be observed that latency is consistently lowered with Bitcoin-NG.

Whilst both protocols scale similarly in transaction frequency, Bitcoin-NG claims to maintain mining power utilisation efficiency at larger block sizes. The authors note this is primarily due to propagation time required for larger blocks with the Bitcoin protocol (Eyal et al. 2016).

Analysis of these results suggest that the Bitcoin-NG protocol appears to utilise mining power more efficiently at higher rates of block generation and block sizes.

Without stating mining power used the authors claims provide limited usefulness as it is unknown whether power usage when testing both protocols is set to be identical, or that one uses more than the other to achieve the claimed TpS or scalability.

Göbel and Krzesinski (2017) conducted independent testing of both protocols, shown in Figure 9, stating that the Bitcoin-NG protocol was capable of PayPal class transaction rates at above 180 TpS with increased block sizes.

Table I: Bitcoin, Ethereum and Bitcoin-NG performance.

1 MB delay	payload 10 mins	TPS MAX	TPS Bitcoin-600	TPS Bitcoin-10	TPS Ethereum	TPS Bitcoin-NG
2 sec	1	2.9	2.9 ± 0.0	2.9 ± 0.0	2.9 ± 0.0	2.9 ± 0.0
	8	23.3	22.9 ± 0.2	22.8 ± 0.1	22.8 ± 0.1	23.5 ± 0.0
	64	186.4	159.1 ± 0.9	158.3 ± 1.1	159.9 ± 1.0	187.4 ± 0.1
10 sec	1	2.9	2.9 ± 0.0	2.9 ± 0.1	2.0 ± 0.0	2.9 ± 0.0
	8	23.3	21.0 ± 0.2	20.9 ± 0.1	21.1 ± 0.6	23.5 ± 0.0
	64	186.4	112.4 ± 0.7	112.9 ± 0.4	112.3 ± 0.6	185.3 ± 0.1
20 sec	1	2.9	2.8 ± 0.0	2.0 ± 0.0	2.8 ± 0.0	2.9 ± 0.0
	8	24.0	19.3 ± 0.2	19.3 ± 0.1	19.3 ± 0.1	23.4 ± 0.0
	64	186.4	117.8 ± 10.4	122.1 ± 11.4	102.3 ± 9.1	182.6 ± 0.1

Figure 9: Bitcoin and Bitcoin-NG performance results (Göbel and Krzesinski 2017)

Their tests were simulated, indicating a best-case scenario rather than actual real-world results, additionally, Göbel and Krzesinski (2017) do not provide precise details of their test systems, whilst their claims appear consistent with Eyal et al. (2016) their results could also be questioned to their validity.

Huang et al. (2019) similarly state that whilst Bitcoin-NG protocol does improve on the transactional throughput of the Bitcoin protocol, it retains some limitations of the Bitcoin protocol, suggesting that the increased throughput of Bitcoin-NG could lead to security risks, such as double spending by malicious block leaders.

Whilst the Bitcoin-NG protocol proposed by Eyal et al. (2016) shows a clear understanding of current limitations, as well as promising scalability results against the Bitcoin protocol, test systems used were not disclosed, only that it was a simulated system, and, alongside the CoDAG protocol, this makes reproduction of results more difficult.

Whilst research by Göbel and Krzesinski (2017) show similar results, again, their research also lacks full disclosure of their test systems.

The testing by Eyal et al. (2016) did not show the achieved TpS, rather they appeared to focus on scalability and latency. It would have been beneficial for the authors to provide TpS results, giving an overall view of their proposed protocol to facilitate comparisons with other protocols.

Lastly, taking into consideration the reservations claimed by Huang et al. (2019) regarding security, these need to be clearly addressed by Eyal et al. (2016), sacrificing security for the claimed improvements would nullify the usefulness of Bitcoin-NG.

2.3 RapidChain

Movahedi et al. (2018) propose an alternative consensus protocol named RapidChain which utilises sharding, dividing the overheads of processing between smaller groups of nodes, claiming RapidChain implements improvements to the sharding consensus protocol, allowing for increased transactional throughput (Movahedi et al. 2018).

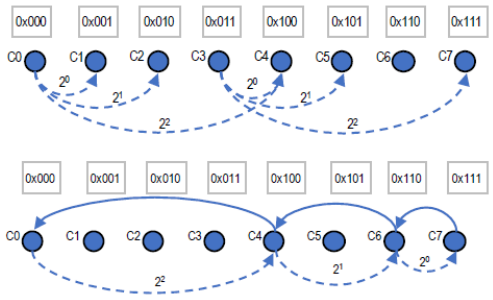


Figure 10: RapidChain routing between committee shards (Movahedi et al. 2018)

Primarily, the researchers claim this improvement is found in optimisation of committee and intra-committee communication, such that updates of transactions are not sent across the entire network, as shown in Figure 10.

The researchers claim that by not requiring the protocol to wait for each node to complete a PoW this removes the latency of this stage of the protocol allowing nodes to complete the PoW offline (Movahedi et al. 2018).

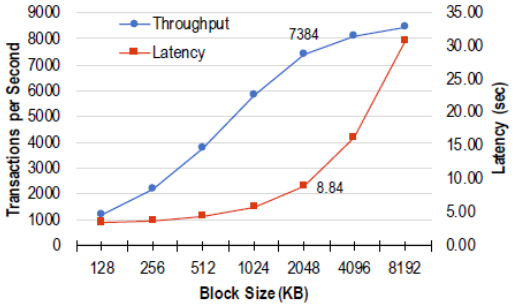


Figure 11: RapidChain TpS based on block size and latency (Movahedi et al. 2018)

The authors tested with 4,000 nodes, Figure 11 shows their claims of how varying block size affects throughput and latency of RapidChain.

Whilst throughput increases shown are promising, it can be seen that by a block size of 4096KB benefits of a larger block size diminish against the increased latency incurred, suggesting an optimal block size is around 2048KB to minimise latency whilst maintaining increased transactional throughput.

Figure 12 shows how the number of nodes affects the TpS of RapidChain, with the researchers claiming that doubling of nodes increases capacity of RapidChain by 1.5x to 1.7x and this increase percentage and TpS can be maintained with additional nodes (Movahedi et al. 2018).

However, it is unclear in the authors research whether this scaling can be maintained indefinitely or whether after a certain number of nodes this increase is negated by the addition of further nodes, limiting the usefulness of the data provided by the authors.

With more than 7,300 TpS in a network of 4,000 nodes RapidChain appears to provide greatly increased TpS compared to CoDAG or Bitcoin-NG, however the limited block size allowed before latency becomes a problem puts it at a disadvantage.

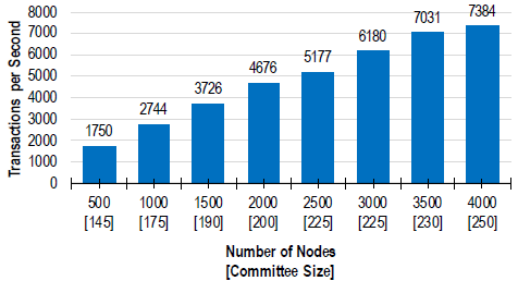


Figure 12: RapidChain TpS based on number of nodes (Movahedi et al. 2018)

Liu et al. (2020) analysed in depth various proposed sharding protocols, claiming that RapidChain appears to provide an efficient model, increasing throughput over alternatives, but does not provide support for smart contracts, limiting its usefulness, and still requiring improvements regarding latency. This research appears to validate conclusions found from analysis of Figure 11.

Movahedi et al. (2018) state their test system and network design but do not provide exact details of hardware, so whilst their findings may be more easily corroborated it would be beneficial to disclose exact details.

The authors claimed consensus protocol appears to provide the greatest increase of TpS and scalability compared to CoDAG or Bitcoin-NG, whilst also providing clear evidence of research in to the problem and the solution that they propose.

However, some reservations are indicated regarding the increase in latency shown in results with larger block sizes, potentially limiting the range of applications to which RapidChain is suited unless the authors can resolve the cause of this.

Of interest would be if the authors can improve block size whilst keeping latency reduced in future versions of RapidChain.

3 Comparison of Protocols

Each proposed consensus protocol utilises greatly differing designs and methods for solving transaction and scalability limitations of current protocols, making direct comparisons challenging.

Eyal et al. (2016) and their Bitcoin-NG protocol builds upon the Bitcoin protocol, providing benefits primarily in mining power utilisation and latency, the TpS is lower than both CoDAG and RapidChain. Bitcoin-NGs limitations appear founded in using the Bitcoin protocol as its basis and it is uncertain if these can be overcome.

CoDAG, proposed by Chen et al. (2019), shows promising increases in TpS compared to Bitcoin-NG, however it cannot match RapidChain. Compared to RapidChain, CoDAG allows for larger block sizes which may be more suitable for use where this is preferred over TpS. The authors of CoDAG did not provide latency results, making it unclear if this is comparative to Bitcoin-NG or RapidChain.

Movahedi et al. (2018) and their RapidChain protocol shows a greater TpS than either CoDAG or Bitcoin-NG. Testing shows linear scaling of TpS with increases of nodes, albeit with some levelling off at greater numbers of nodes. Latency and block size appear to be the greatest drawback to RapidChain, with larger block sizes greatly increasing latency. RapidChain's main limitation appears to be block size, potentially preventing its use in some applications compared to CoDAG's larger block size.

4 Conclusions

This research paper critically evaluated the CoDAG, Bitcoin-NG and RapidChain protocols.

Each consensus protocol shows good science, strong methodology and appear to provide improvements in TpS compared to traditional PoW consensus protocols.

Each protocol appears to offer unique and well-developed solutions to current limitations. Nevertheless, each DLT use case is different, making choice of consensus protocol used be based as much on individual DLT requirements as its potential TpS or scalability.

It appears there is no currently accepted method to directly compare protocols, leading to difficulty ascertaining which consensus protocol provides the greatest increase of TpS or its scalability compared to others. Whilst each evaluated consensus protocol provides improvements over

current protocols, a common testing methodology would be beneficial.

Yang et al. (2019) and their CoDAG protocol provides a promising basis for future work and it would be of interest to see CoDAG improved, refined and evaluated with fully disclosed systems. Its approach to DAG width management allows for granular control over its implementation and appears to scale effectively. Irfan et al. (2018), state that DAG shows a strong use case for DLT in overcoming current limitations. CoDAG appears to be a well implemented protocol based on DAG but with limited testing and independent verification.

Eyal et al. (2016) with Bitcoin-NG is the most mature of the protocols evaluated, other researchers such as Göbel and Krzesinski (2017) have carried out independent verification of the protocol and reached similar conclusions to those shown by Eyal et al. (2016).

Huang et al. (2019) showed concerns over the security of the protocol, a possible limitation of its evolutionary approach from the Bitcoin protocol, at higher transaction throughput rates which would limit Bitcoin-NGs usefulness unless addressed.

RapidChain, proposed by Movahedi et al. (2018), enjoys transactional throughput advantages over both CoDAG and Bitcoin-NG. Their implementation of sharding and network communication alleviates latency and network overheads of other protocols, whilst achieving substantial transactional throughput improvements.

Their research shows further work should be undertaken to address the increase of latency and levelling off of TpS at larger block sizes. Liu et al. (2020) provides validation of RapidChains transactional throughput but do not focus on scalability, limiting somewhat any additional confirmation of the findings of Movahedi et al. (2018).

RapidChain appears to provide the greatest TpS of all consensus protocols reviewed. Latency of RapidChain at larger block sizes appears to be the main drawback which may limit its usefulness for certain applications, CoDAG appears suitable for larger block sizes, although latency results for CoDAG were not provided. For applications requiring block sizes of less than 2048KB the latency drawbacks are reduced, other applications may not be so sensitive to latency and may tolerate the increased time.

Further research is recommended for each proposed consensus protocol, with a focus on providing greater transparency of test systems and broader comparative testing against current consensus protocols.

References

Bamakan, S. M. H., Bondarti, A. B. and Motavali, A., 2020, 'A survey of blockchain consensus algorithms performance evaluation criteria' *Expert Systems with Applications*, volume 154, 15th September, Article 113385.

Chen, Z., Cui, L., Ming, Z., Xu, K., Xu, M. and Yang, S., 2019, 'CoDAG: An Efficient and Compacted DAG-Based Blockchain Protocol' *2019 IEEE International Conference on Blockchain (Blockchain)*, Atlanta, USA, 14th-17th July, pages 314-318, IEEE.

Eyal, I., Gencer, A. E., Sirer E. G., and van Renesse, R., 2016, 'Bitcoin-NG: A Scalable Blockchain Protocol' *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI '16)*, Santa Clara, USA, 16th-18th March, pages 45-59.

Göbel, J. and Krzesinski, A. E., 2017, 'Increased block size and Bitcoin blockchain dynamics' *2017 27th International Telecommunication Networks and Applications Conference (ITNAC)*, Melbourne, Australia, 22nd-24th November, pages 1-6, IEEE.

Haq, I. U., Irfan M. U., Muneeb, M. and Pervez, H., 2018, 'A Comparative Analysis of DAG-Based Blockchain Architectures' *12th International Conference on Open-Source Systems and Technologies (ICOSST)*, Lahore, Pakistan, 19th-21st December, pages 27-34, IEEE.

Hassani, H., Huang, X. and Silva, E., 2018, 'Big-Crypto: Big Data, Blockchain and Cryptocurrency', *Big Data and Cognitive Computing*, volume 2, issue 4, pages 1-15.

He, Y., Liu, J., Si, P., Yu, F. R. and Zhang Y., 2018, 'Virtualization for Distributed Ledger

Technology (vDLT)' *IEEE Access*, volume 6, pages 25019-25028.

Hewage, C., Khan, I., Lidgey, B. and Shahaab, A., 2019, 'Applicability and Appropriateness of Distributed Ledgers Consensus Protocols in Public and Private Sectors: A Systematic Review.' *IEEE Access*, volume 7, pages 43622-43636.

Huang, T., Liu, J., Liu, Y., Xie, J., Xie, R. and Yu, F.R., 2019, 'A Survey on the Scalability of Blockchain Systems', *IEEE Network*, volume 33, no. 5, pages 166-173, IEEE.

Liu, R. P., Ni, W., Wang, X., Yu, G., Yu, K. and Zhang, J. A., 2020, 'Survey: Sharding in Blockchains', *IEEE Access*, volume 8, pages 14155-14181, IEEE.

Mahmoud, Q. and Shahriar Hazari, S., 2020, 'Improving Transaction Speed and Scalability of Blockchain Systems via Parallel Proof of Work' *Future Internet*, volume 12, pages 1-19.

Movahedi, M., Raykova. M., and Zamani, M., 2018, 'RapidChain: Scaling Blockchain via Full Sharding'. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18)*, New York, USA, pages 931–948.

Analytical Evaluation of Current Machine Learning-Based Deepfake Detection Methods Aimed at Enhancing Detection Accuracy

Raúl Alvarado García

Abstract

The appearance of Deepfake technology has risen the untrustworthiness of digital contents leading the research community to develop new techniques to combat it. This paper analysed and critically evaluated state-of-the-art research methods on Deepfake detection based on different machine learning classifiers, including: Random Forest, Support Vector Machine, Convolutional Neural Networks and One-Class classifiers. These methods were compared to determine the most accurate approach, understanding their advantages and drawbacks and introducing further research to enhance current Deepfake detection accuracy. Lastly, additional research was proposed to enhance detection accuracy in other areas that rely on machine-learning image processing as well.

1 Introduction

Over the last years, the appearance of Deepfake technology has risen the untrustworthiness of digital contents. Latest advances in Deepfake generation provide widely available tools able to produce altered content whose integrity cannot be detected by the human eye. As these techniques are becoming more complex, it is increasingly challenging to accurately detect fake content. In consequence, the development of Deepfake detection techniques has become a high priority topic for the research community (Nguyen H.M. and Derakhshani R. 2020).

In recent research, various approaches to combat Deepfake have been explored. While some researchers, like Hasan H.R. and Salah K. (2019), have proposed methods to prevent Deepfake generation misuse, additional research has been conducted on content integrity verification. This is the case of Tursman E. et. al. (2020), who built a system to determine the authenticity of an event detecting face geometry inconsistencies. Similarly, Jung T. et. al. (2020) proposed a detection algorithm analysing significant changes in blinking patterns.

However, these detection approaches do not perform machine learning-based classification and present lower accuracy rates compared to machine learning-based methods.

Hence, the aim of this paper is to analyse and critically evaluate current research on machine learning-based Deepfake detection to find the most accurate technique in the field. The paper will present the conclusions reached from the analysis, evaluation and comparison of the different methods by the respective machine-learning classifier, proposing further research to enhance the detection accuracy.

2 Current Deepfake Detection Methods Using Machine Learning Classifiers

In this section, current research techniques on Deepfake detection will be critically analysed and evaluated. The section presents the different methods according to the applied machine learning classifier.

2.1 Random Forest

Guarnera L. et. al. (2020) proposed a detection method based on the Expectation-Maximization algorithm to extract a discriminative feature vector, named convolutional trace. They analysed the convolutional traces produced by GANs in the Deepfake generation process.

For their experiments, they utilized ten Deepfake generation architectures to produce six datasets focused on altered faces and four not containing faces, each composed of 2000 images. Additionally, they used CELEBA dataset as real-image source. After extracting the convolutional traces, they evaluated K-NN, SVM, LDA and Random-Forest classifiers to verify content authenticity obtaining Table 1 results.

	CELEBA Vs DeepNetworks		
	Kernel Size		
	3x3	5x5	7x7
3-NN	**89.80**	77.38	78.63
5-NN	**90.79**	77.20	77.80
7-NN	**90.44**	76.47	78.39
9-NN	**90.30**	77.20	78.28
11-NN	**89.80**	77.29	77.45
13-NN	**89.73**	77.66	77.69
SVMLinear	**84.14**	76.28	80.28
SVMSigmoid	58.57	61.36	**63.52**
SVMrbf	**91.22**	80.04	80.87
SVMPoly	**88.74**	78.66	78.87
LDA	**83.50**	77.38	78.98
Random Forest	**98.07**	93.81	91.22

Table 1 Classifiers' accuracy (Guarnera L. et. al. 2020)

Furthermore, they conducted experiments on real-world conditions with FaceApp, generating 471 Deepfake images from which their method detected 437 (92.78%).

They tested model's robustness applying transformations to input Deepfakes. The results showed that datasets with bigger images were less affected by attacks.

Lastly, they compared the proposed method with three state-of-the-art CNN-based models, achieving the results shown in Fig.1.

Guarnera L. et. al. (2020) conclude that their method has high discriminative power, robustness and semantics independence. They claim that it is also good at detecting Deepfakes without faces. They present further research, as robustness experiment achieved best accuracy results for images rotated 90º anticlockwise.

In their work, the authors used an ad-hoc dataset based on ten generation architectures that included different contexts, proving method's generalizability. They demonstrated its accuracy in real-world conditions through FaceApp market application. Additional controlled experiments were conducted proving robustness and higher accuracy than CNN-based approaches, successfully mitigating bias applying the same attacks to images and using the same datasets in comparisons. Therefore, as the experiments were conducted properly, obtaining fair and objective results that provide evidence of method's accuracy in different scenarios, the conclusions reached are valid.

Patel M. et. al. (2020) presented a Deepfake video detection technique based on transfer learning. In the proposed pipeline, the face region is firstly detected and extracted as an image using a Multi-Task Cascaded Convolutional Neural Network (MTCNN). Secondly, using pre-trained extractors, features are obtained from the previous face images. These features are used to classify the videos through a Random-Forest algorithm.

The authors used the Deepfake Detection Challenge (DFDC) dataset to extract 7000 images for training and 2000 for testing. They conducted tests using five feature extractors (VGG16, ResNet50, InceptionV3, MobileNet, and DenseNet), which were implemented using transfer learning.

They presented results comparing the feature extractors regarding accuracy, precision, recall and Area Under Curve (AUC) as shown in Table 2.

Feature extractor	Accuracy	Precision	Recall	AUC
VGG16	0.893	0.881	0.91	0.893
ResNet50	0.89	0.873	0.913	0.89
InceptionV3	0.862	0.848	0.881	0.86
MobileNet	**0.902**	**0.893**	0.913	**0.9**
DenseNet	0.897	0.879	**0.922**	0.897

Table 2 Feature extractors results (Patel M. et. al. 2020)

They conclude that, although the overall best results are presented by MobileNet, their preferred extractor is DenseNet as optimizing recall minimises false negatives.

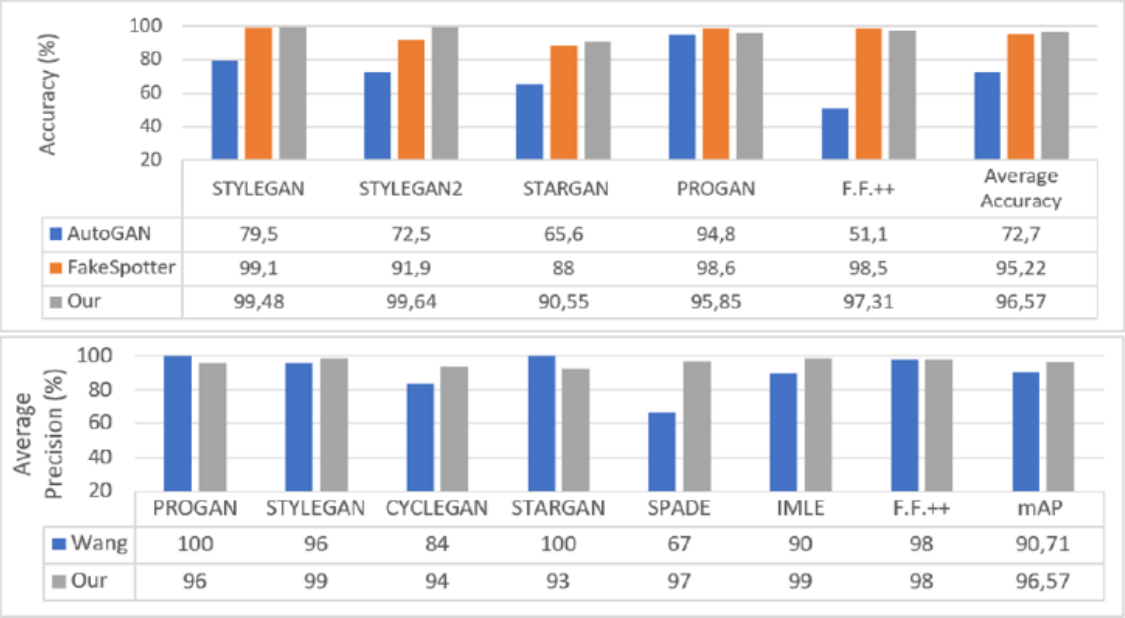


Fig.1 CNN-based methods comparison (Guarnera L. et. al. 2020)

Patel M. et. al. (2020) properly conducted a controlled experiment to evaluate the accuracy of five feature extractors using the same architecture and datasets to avoid biased results. They used DFDC dataset, which is widely used by the Deepfake detection research community due to its good-quality Deepfakes. However, it is not mentioned that DFDC contains audio and video Deepfakes which are labelled equally as "Deepfake", leading to false negatives where audio Deepfakes are classified as real. Therefore, as the proposed model only evaluates images, their experiments could have been conducted on an image-only dataset to achieve more precise results. In consequence, due to the lack in precision, further testing needs to be done to fully prove conclusions validity.

2.2 Support Vector Machine

Yang X. et. al. (2019) proposed a detection method based on the estimation of 3D head poses to reveal errors introduced by the synthesis of different face regions during the Deepfake generation process. They present the 3D pose estimation as an optimization problem that can be solved applying the Levenberg-Marquardt algorithm.

To evaluate their approach, the authors trained a Support Vector Machine (SVM) classifier using UADFV and "DARPA MediFor GAN Image/Video Challenge" datasets. UADFV dataset contains 49 real and 49 Deepfake videos while DARPA gathers 241 real and 252 fake images.

They extracted frames from 35 real and 35 Deepfake videos from UADFV for training while the remaining 14 real and 14 fake videos from UADFV and all the images from DARPA were used for testing.

The results obtained considering frames as unit of analysis showed a 0.89 detection rate for UADFV and 0.843 for DARPA in AUROC metric. Additionally, they presented results using video as unit of analysis for UADFV comparing five types of features, achieving Table 3 scores.

features	frame	video
$\vec{v}_a - \vec{v}_c$	0.738	0.888
$\vec{r}_a - \vec{r}_c$	0.798	0.898
$R_a - R_c$	0.853	0.913
$(\vec{v}_a - \vec{v}_c)$ & $(\vec{t}_a - \vec{t}_c)$	0.840	0.949
$(\vec{r}_a - \vec{r}_c)$ & $(\vec{t}_a - \vec{t}_c)$	0.866	0.954
$(R_a - R_c)$ & $(\vec{t}_a - \vec{t}_c)$	0.890	0.974

Table 3 UADFV results (Yang X. et. al. 2019)

Concluding, Yang X. et. al. (2019) claim that the analysis of inconsistencies in estimated head

poses provides a good approach to identify Deepfake contents. However, they also state that their method presents limitations when input videos are blurry as it hinders facial landmarks extraction.

The authors provide a detailed mathematical explanation of their methodology and properly explain data extraction and distribution processes proving high reproducibility and reliability. They performed a controlled experiment evaluating different features trained and tested using the same dataset, reducing potential bias. The presented results were based on frame and video analysis, enhancing comparability with other models. Therefore, these results are objective and fair, providing evidence to determine conclusions validity.

As an area of improvement, a larger image dataset could have been used to achieve frame-based results that are more generalizable and representative of real-world situations.

2.3 Convolutional Neural Network

Chang X. et. al. (2020) proposed a model to detect Deepfake images based on an enhanced VGG16-CNN (NA-VGG) by adding a SRM layer and an image-augmentation layer in front of it. The SRM layer pre-processed target RGB images to generate local noise feature maps, which were used as the input for the convolutional neural network (CNN). On the other hand, the image augmentation layer generated new images applying transformations to emphasise the detection of traces.

They compared the average performance of state-of-the-art detection methods on different datasets and decided to use Celeb-DF dataset, which presented the overall lowest accuracy due to the high-quality of its Deepfakes.

Celeb-DF contains 408 original and 795 Deepfake videos from which 15,168 images were extracted (6459 original and 8709 altered), using 12,416 for training, 1,376 for verification and 1,376 for testing.

Their method presented an 85.7% AUC performance in Celeb-DF and was compared with other methods using the same dataset, as shown in Table 4.

Methods	Average AUC performance
Average of Several Methods[22]	48.7%
Two-Stream[10]	55.7%
VGG16	56.4 %
SRM filter +VGG16	73.2 %
NA-VGG	85.7 %

Table 4 Methods comparison (Chang X. et. al. 2020)

The authors claim that their model is much more accurate than other detection methods based on Celeb-DF results. They state that other models which are trained using lower-quality datasets present performance degradation on high-quality datasets. Lastly, they conclude that the use of SRM filter and image-augmentation layers enhance the detection accuracy of VGG16 networks.

In their work, Chang X. et. al. (2020) justified Celeb-DF dataset selection demonstrating its challengingness and, hence, avoiding biased results. They performed a controlled experiment properly comparing the base VGG16 with the proposed variations under the same conditions, proving enhanced accuracy for each layer. Additionally, their experiments present high reproducibility as data extraction and distribution processes are explained precisely. Therefore, the results obtained are objective and provide enough evidence to prove conclusions validity.

Montserrat D.M. et. al. (2020) proposed a method to detect manipulated faces in videos by combining a CNN and a Recurrent Neural Network (RNN). The presented model, shown in Fig.2, firstly extracts the face region using a MTCNN. Afterwards, face features are extracted through the EfficientNet-CNN with additive angular margin loss (ArcFace). Lastly, a final estimation is obtained merging the features from all face regions using an Automatic Face Weighting (AFW) algorithm and a Gated Recurrent unit (GRU).

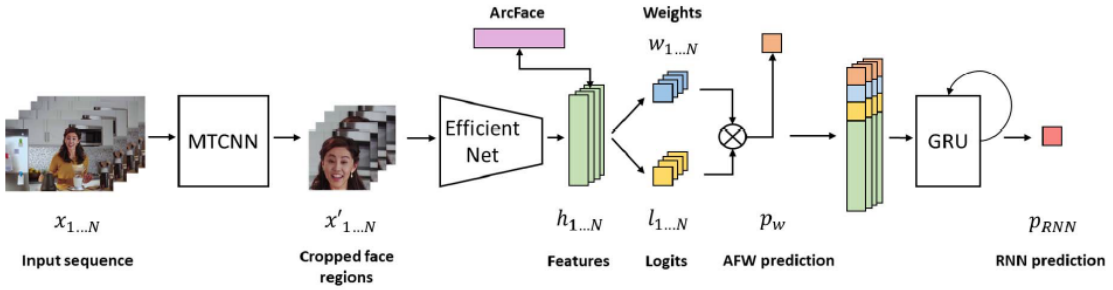


Fig. 2 Montserrat D.M. et. al. (2020) proposed architecture

For their experiments, the authors used a pre-trained MTCNN and trained the EfficientNet-CNN, the GRU and AFW layers. They evaluated their method using the DFDC dataset.

This dataset contains 123,546 videos with manipulated images and audio, from which 119,245 labelled videos were used in the evaluation process. The videos were divided in 50 numbered parts using 30 for training, 10 for validation and 10 for testing.

Their model obtained a 91.88% detection accuracy in DFDC and was compared with other methods, as shown in Table 5. They used the validation dataset to configure the other methods to obtain the best accuracy.

Method	Validation	Test
Conv-LSTM [30]	55.82%	57.63%
Conv-LSTM [30] **+ MTCNN**	66.05%	70.78%
EfficientNet-b5 [42]	79.25%	80.62%
Xception [37]	78.42%	80.14%
Ours	92.61%	91.88%

Table 5 Test Results (Montserrat D.M. et. al. 2020)

The authors claim that the detection accuracy increases when an AFW layer and a GRU are added. Additionally, they present future work implementing audio detection to enhance accuracy results.

Montserrat D.M. et al. (2020) used DFDC, which is a large-scale dataset widely used in Deepfake detection research. They used 119,245 videos for evaluation, obtaining generalizable results that are representative of model's real-world accuracy. These results were compared in a controlled experiment with other methods which were validated to obtain the best results, mitigating potential bias. Additionally, authors are aware that DFDC contains both audio and video Deepfakes and present a continuation of their work to achieve more accurate and precise results including audio detection. Therefore, although current model could achieve higher precision if tested on image-only dataset, the evidence of further work enhances the validity of current results and conclusions.

Rana M.S. and Sung A.H. (2020) presented a detection method called DeepfakeStack, based on the combination of deep learning techniques. DeepfakeStack's architecture consisted of several pre-trained base-learners with a meta-learner on top of them. As meta-learner, they created a CNN-based classifier which was embed in a larger multi-headed neural network. The ensemble technique applied was stacking ensemble so that the meta-learner was trained using the predictions of the base-learners as input vector.

Their experiments were based on FaceForensics++ (FF++) dataset. They used 49 real and 51 altered videos, extracting 101 frames from each to reduce the computational time. These frames were pre-processed to extract the face region and fed into the classifier. As base-learners, they used seven deep learning models that were trained with transfer learning and Greedy Layer-wise Pretraining.

After training the meta-learner with the testing dataset for 300 epochs, the results for the base-learners and ensemble model (DFC) are compared, as shown in Table 6.

The authors claim that, according to AUROC results, their model is the most efficient possible. They conclude that DeepfakeStack provides a

strong basis for future Deepfake detectors development and present further research implementing Blockchain technology on their model.

Model	Precision		Recall		F1-score		Accuracy	AUROC
	0	*1*	*0*	*1*	*0*	*1*		
XCEPN	0.94	1.00	1.00	0.94	0.97	0.97	96.88	0.976
INCV3	0.88	0.95	0.85	0.88	0.86	0.87	86.49	0.866
MOBN	0.84	1.00	1.00	0.81	0.92	0.90	90.74	0.911
RSN101	0.94	0.96	0.96	0.94	0.95	0.95	94.95	0.954
IRNV2	0.82	1.00	1.00	0.79	0.90	0.88	89.26	0.899
DNS121	0.93	1.00	1.00	0.93	0.96	0.96	96.34	0.969
DNS169	0.95	1.00	1.00	0.94	0.97	0.97	97.13	0.971
DFC	**0.99**	**1.00**	**1.00**	**0.99**	**1.00**	**1.00**	**99.65**	**1.000**

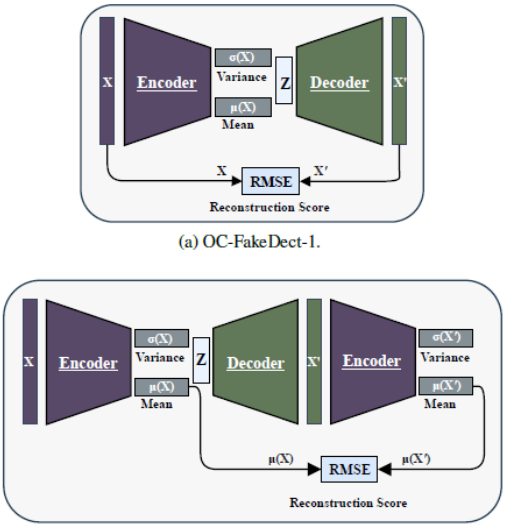
Table 6 Test Results (Rana M.S. and Sung A.H. 2020)

Rana M.S. and Sung A.H. (2020) used FF++, which is a popular and representative large-scale dataset in Deepfake detection research. They generated over 10,000 frames to evaluate the model obtaining generalizable results. The data extraction and training processes are properly explained enhancing experiments reproducibility and reliability. Additionally, they evaluated a wide range of deep learning models and compared the individual results with the ensemble model using the same data in a controlled experiment, demonstrating unbiased results. Therefore, as the experiments were conducted objectively and achieved fair results, there is enough evidence to demonstrate conclusions validity.

2.4 One-class Classifier

Khalid H. and Woo S.S. (2020) proposed a one-class classification detection technique by training a classifier only with real images and considering Deepfakes as anomalies. Their method, called OC-FakeDect, presented two approaches using one-class Variational Autoencoders (OC-VAE), in which one of them included an additional encoder, as shown in Fig.3.

For their experiments, they used five datasets containing different types of Deepfake images from FF++. Additionally, the non-altered FF++ videos were used to extract 30,000 real human face images for training through a MTCNN.
To verify OC-FakeDect detection they used 9,000 real and 9,000 fake images and measured the loss per epoch. For testing, they used 500 real and 500 Deepfake images.



(a) OC-FakeDect-1.

(b) OC-FakeDect-2.

Fig. 3 Khalid H. and Woo S.S. (2020) approaches

The authors compared both proposed approaches with the original one-class Autoencoder (OC-AE) using the same training and testing procedures. Furthermore, they compared their classifiers with MesoNet and XceptionNet binary classifiers on the same five datasets achieving Table 7 results.

Model	NT	DF	FS	F2F	DFD
OC-AE	54.60	49.20	48.20	47.80	66.90
OC-FakeDect1	95.30	86.20	84.80	70.70	98.00
OC-FakeDect2	**97.50**	88.40	86.10	71.20	**98.20**
MesoNet [25]	40.67	87.27	61.17	56.20	N/A
Xcep. Net [25]	80.67	**96.36**	**90.29**	**86.86**	N/A

Table 7 Test Results (Khalid H. and Woo S.S. 2020)

They conclude that OC-FakeDect out-performs previous one-class classifiers as well as some binary classifiers as MesoNet. They claim that one-class-based detection presents a promising approach to combat new Deepfake methods as it does not require fake samples to be trained.

As other researchers, the authors used the popular FF++ dataset, enhancing comparability with other models. The amount and variety of data used from FF++ proved generalization and was representative of real-world situations. Additionally, they provided detailed information regarding data gathering, training, validation and testing processes, demonstrating high reproducibility. Different methods were compared

in a controlled experiment using the same datasets and training processes, avoiding bias. Therefore, the results presented are fair and demonstrate the conclusions reached are valid.

3 Comparison of Methods

The following section presents a comparison of the evaluated Deepfake detection methods based on their methodologies and accuracy results. As these approaches were tested using different datasets, the obtained results are considered an approximation of the potential real-world accuracy of each method.

Comparing the three analysed CNN-based methods it is observed that not only Rana M.S. and Sung A.H. (2020) achieved the highest accuracy (99.65%), but they also applied a solution to the high computational demands problem of CNNs using pre-trained models with transfer learning. Additionally, their architecture provides a major advantage due to its versatility as it could be used to integrate future deep learning technologies to combat new Deepfake techniques.

Regarding non-CNN-based methods, Guarnera L. et. al. (2020) and Yang X. et. al. (2019) models presented the highest accuracy (98.7% and 0.974AUROC respectively). However, while Yang X. et. al. (2019) approach was particularly effective in face-swap situations, Guarnera L. et. al. (2020) technique proved its accuracy regardless of the context. This is a significant advantage as high generalization enables the detection of altered content in more scenarios, including situations not containing humans.

Khalid H. and Woo S.S. (2020), opposed to other methods, approached the detection problem as one-class classification using only real images. Although the achieved accuracy (88.3%) is not as high, their method presents the advantage that it does not require further training to be able to deal with new Deepfake generation techniques. Therefore, their method would be particularly useful to combat future Deepfake models since the moment they are released.

4 Conclusions

This paper analysed and evaluated research methods that provided valid and accurate solutions to the Deepfake detection problem. The evaluated work presented two main strategies: CNN-based and non-CNN-based approaches. After the analysis, it is concluded that the differences in accuracy presented by each current are not significant and, therefore, non-CNN based approaches are preferred for similar accuracy, as they are less time and computational demanding.

However, the CNN-approach by Rana M.S. and Sung A.H. (2020) demonstrated really high accuracy and an interesting architecture based on an ensemble model. Therefore, the most accurate and consistent non-CNN-based method presented by Guarnera L. et. al. (2020) could implement a similar architecture in the feature extraction phase that would enhance the detection of convolutional traces. Further research merging both approaches could provide a fast and accurate method that would not be as computational demanding as CNN approaches and could be applied to any image regardless of containing humans.

It is also concluded that Deepfake detection can be approached as a one-class classification problem and, thus, avoid the need to collect Deepfake samples to train the detection model, as shown by Khalid H. and Woo S.S. (2020). Further work on enhancing the accuracy of one-class detection models could lead to high-generalizable methods that would be able to combat future Deepfake generation models since their emergence without requiring further training.

Lastly, as the Deepfake detection problem has a strong basis on image detection and classification, further research could be conducted to adapt and implement the analysed methods in different areas that require image processing as well. For example, in the case of the automotive industry, methods could be implemented to enhance the accuracy of autonomous vehicles' recognition systems. Similarly, in the healthcare sector, these methods could lead to increased accuracy in pattern recognition systems for disease diagnosis, such as cancer cell detection systems.

References

Chang X., Wu J., Yang T. and Feng G., 2020, 'DeepFake Face Image Detection based on Improved VGG Convolutional Neural Network', *2020 39th Chinese Control Conference (CCC)*, pp.7252-7256

Guarnera L., Giudice O. and Battiato S., 2020, 'Fighting Deepfake by Exposing the Convolutional Traces on Images', *IEEE Access, Vol.8,* pp.165085-165098

Hasan H.R. and Salah K., 2019, 'Combating Deepfake Videos Using Blockchain and Smart Contracts', *IEEE Access, Vol.7,* pp.41596-41606

Jung T., Kim S. and Kim K., 2020, 'DeepVision: Deepfakes Detection Using Human Eye Blinking Pattern', *IEEE Access, Vol.8,* pp.83144-83154

Khalid H. and Woo S.S., 2020, 'OC-FakeDect: Classifying Deepfakes Using One-class Variational Autoencoder', *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),* pp.2794-2803

Montserrat D.M., Hao H., Yarlagadda S.K., Baireddy S., Shao R., Horvath J., Bartusiak E., Yang J., Guera D., Zhu F. and Delp E.J., 2020, 'Deepfakes Detection with Automatic Face Weighting', *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),* pp.2851-2859

Nguyen H.M. and Derakhshani R., 2020, 'Eyebrow Recognition for Identifying Deepfake Videos', *2020 International Conference of the Biometrics Special Interest Group (BIOSIG),* pp.1-5

Patel M., Gupta A., Tanwar S. and Obaidat M.S., 2020, 'Trans-DF: A Transfer Learning-based end-to-end Deepfake Detector', *2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA),* pp.796-801

Rana M.S. and Sung A.H., 2020, 'DeepfakeStack: A Deep Ensemble-based Learning Technique for Deepfake Detection', *2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom),* pp.70-75

Tursman E., George M., Kamara S. and Tompkin J., 2020, 'Towards Untrusted Social Video Verification to Combat Deepfakes via Face Geometry Consistency', *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),* pp.2784-2793

Yang X., Li Y. and Lyu S., 2019, 'Exposing Deep Fakes Using Inconsistent Head Poses', *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp.8261-8265

A Critical Evaluation of Current Brain-Computer Interface Control Methods

Michael Rogan

Abstract

Brain-Computer Interface (BCI) is the most intimate and direct method of interfacing with a computer device, research is being conducted across many application areas. This paper evaluates three different implementation methods for BCI systems; BCI paired with virtual reality for visual feedback, BCI when used to control unmanned vehicles, and novel BCI control methods such as intrusive BCI implants. Analyzing the methods across different application areas will highlight benefits and shortcomings of each method. Recommendations will be made on which methods can be hybridized to create an improved BCI control method, with recommendations being made for further research.
.

1 Introduction

Brain-Computer Interfaces (BCIs) have the potential to be an extremely useful and versatile technology, with the potential to provide responsive and intuitive control to users in a wide variety of applications. However, one of the biggest obstacles in the way of widespread BCI adoption is creating a system which is as accurate and reliable as traditional methods of controlling devices such as touch screens, joysticks, etc. Many users who try BCI systems for the first time are unfamiliar with the technology, which can lead to high error rates and significant difficulty accurately using the system (Kosmyna, 2019).

Chan, A. and Dascalu, S. (2020) propose a method of interacting with digital spaces which received commands through a combination of facial expressions/gestures and reading brain signals. This was applied to allow physically disabled test subjects to use geospatial applications such as google earth with minimal training. This technology reduces the user training required before acceptable controllability is achieved, even for first-time BCI users.

Meng, J. et. al. (2018) identified a promising method of non-intrusive BCI implementation known as electroenceph-alography (EEG) coupled with motor imagery tasks as being a much more efficient method of controlling objects in a 3-dimensional space than other non-intrusive methods of control. Meng, J. et. al. (2018) also note that this method was perceived to be easier to use by test subjects, allowing them to focus on multiple tasks at once. Improving the perceived usability of a BCI system may be vital to improving controllability.

This paper aims to evaluate current research surrounding methods of controlling BCI systems in order to ascertain which methods are most suitable for a variety of scenarios, allowing further research to be built upon this knowledge. In doing so this paper should bring BCI techniques closer to having a responsiveness equal to that of the human body itself in terms of reacting to brain signals.

2 Current Brain-Computer Interface control techniques

In this section we will discuss current methods of pairing BCI systems with virtual reality systems, BCI and its control systems when paired with drones, and alternative methods to improving the controllability of BCI systems, such as an investigation into the impact of compounding variables on the controllability of BCI systems.

2.1 BCI paired with virtual reality for visual feedback

Jaehoon, C. and Sungho, J. (2020) proposed a technique which employed a hybrid BCI and AR display space. The hybrid BCI used a combination of SSVEP and MI based BCI.

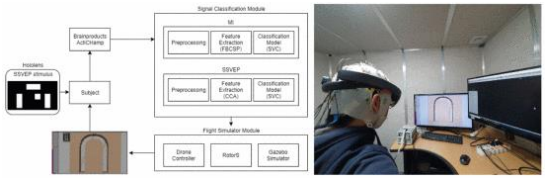


Figure 1. Overview of BCI System (Jaehoon, C. and Sungho, J. 2020)

For their research, Jaehoon, C. and Sungho, J. (2020) split their method into two stages, the training stage and then the experimental control stage.

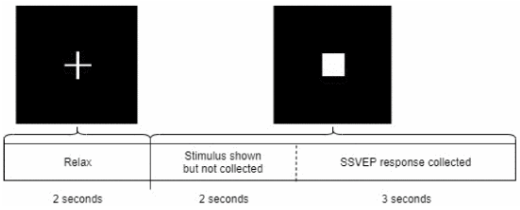


Figure 2 SSVEP training (Jaehoon, C. and Sungho, J. 2020)

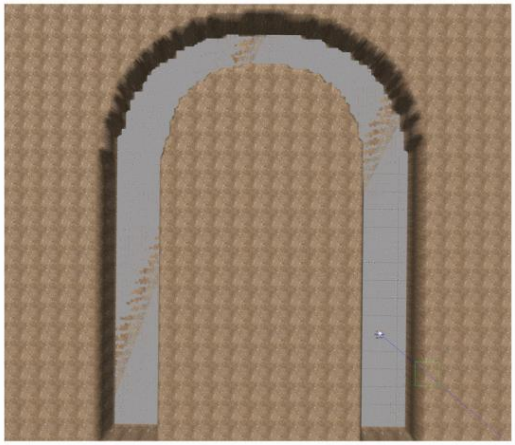


Figure 3 Control environment (Jaehoon, C. and Sungho, J. 2020)

To conclude, Jaehoon, C. and Sungho, J. (2020) claim to have demonstrated that a hybrid MI/SSVEP BCI system coupled with AR can be effectively used to control a quadcopter across a two-dimensional plane.

Jaehoon, C. and Sungho, J. (2020) provide enough details of their experimental setup to allow others to replicate their methods, indicating the results are valid.

Coogan, C. G. and Bin, H. (2018) posit that using virtual reality/augmented reality to create visual feedback for users controlling a BCI system can help make the system more comfortable to use over a long period of time.

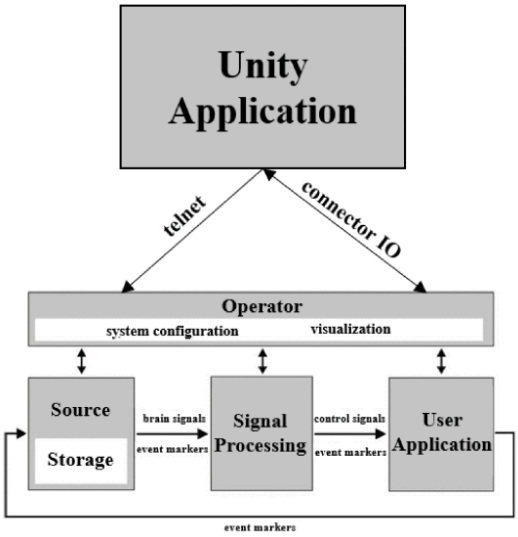


Figure 4 Unity/BCI2000 Integration (Coogan C. G. and Bin H. 2018)

To test their hypothesis, Coogan, C. G. and Bin, H. (2018) had the subject control an in-game object using BCI to try and hit a series of targets.

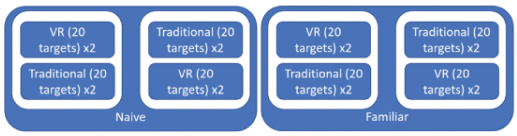


Figure 5 Study pipeline (Coogan, C. G. and Bin, H. 2018)

Coogan, C. G. and Bin, H. (2018) report in their findings that there is no significant performance difference between participants using VR as oppose to those using a standard computer monitor.

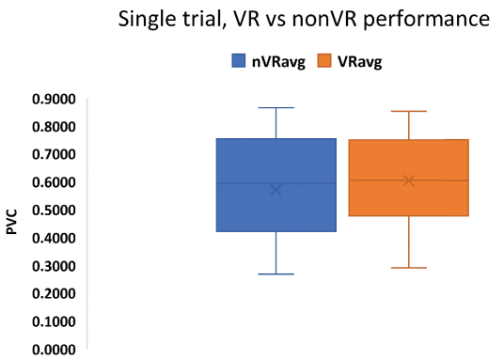


Figure 6 Performance differences between VR and control group (Coogan, C. G., and Bin, H. 2018)

Coogan, C. G. and Bin, H. (2018) provide extensive details on their experimental method, the hardware and software used, and the protocols needed to replicate the system. Their results are clear and precise, and the claims made in their conclusion are substantiated by the evidence provided.

Both Coogan, C. G. and Bin, H. (2018) and Jaehoon, C. and Sungho, J. (2020) present evidence which suggests that using virtual reality to present visual feedback is a viable method of BCI control. Coogan, C. G. and Bin, H. (2018) show that VR is of no detriment to performance when paired with BCI systems compared to computer screens, and Jaehoon, C. and Sungho, J. (2020) present a valid method of creating a hybrid BCI-VR system which attempts to find solutions for issues associated with SSVEP and MI when used individually.

From this research it can be concluded that VR BCI systems are a valid and accessible method of BCI interfacing, with, as stated by Coogan, C. G. and Bin, H. (2018), the potential to allow interfacing with many BCI-enabled devices across the internet of things.

Juliano, J. M. et al. (2020) proposed a novel method of using BCI to rehabilitate people suffering with motor impairments. Their proposed method involves using an immersive VR system (HMD-VR).

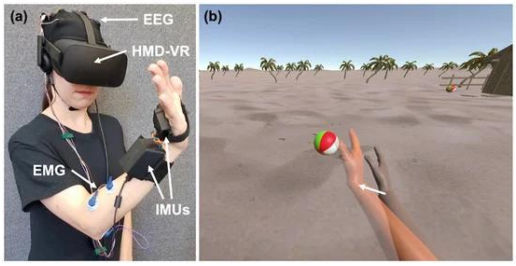


Figure 7 Example of the REINVENT system (Juliano, J. M. et al. 2020).

The trials composed of two rendered objects within the subject's focus. An arm which the participant controlled, and a target arm, which would move to different positions. For each trial, the participant must match the target arm using their thoughts.

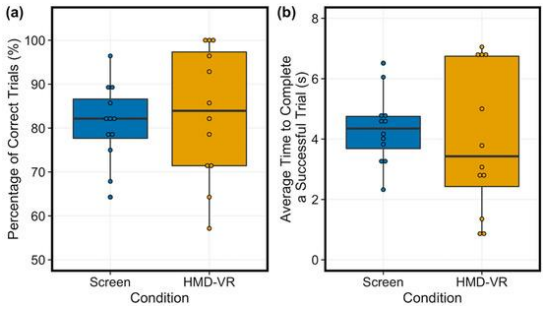


Figure 8 Average trial performance and time to complete successful trials (Juliano, J. M. et al. 2020)

Juliano, J. M. et al. (2020) conclude their research by stating that HMD-VR BCI systems create a higher sense of embodiment which could positively impact neurofeedback when compared to feedback on a computer screen.

Juliano, J. M. et al. (2020) report the results of their experiment in a clear and objective way. Their reporting matches closely to the presented data sets and does not make any unsubstantiated claims.

2.2 BCI Drone Control

Al-Nuaimi, F. A. et al. (2020) proposed a method of controlling a drone using a BCI headset and having the drone intercept a traditionally controlled drone.

Figure 9 Flashing images in the P300-baded drone chasing paradigm (Al-Nuaimi, F. A. et al. 2020)

Al-Nuaimi, F. A. et al. (2020) split the test into two users, User 1 and User 2, User 1 was in control of the BCI enabled device.

User 2, as explained by Al-Nuaimi, F. A. et al. (2020), commands a drone using simple Python commands using a laptop. The success or failure of the task will rely on User 1's ability to actually catch User 2's drone.

Figure 10 Implementation of P300-based drone chasing method (Al-Nuaimi, F. A. et al. 2020)

Al-Nuaimi, F. A. et al. (2020) state that this is a much faster method of controlling a drone compared to selecting the commands by hand using a laptop.

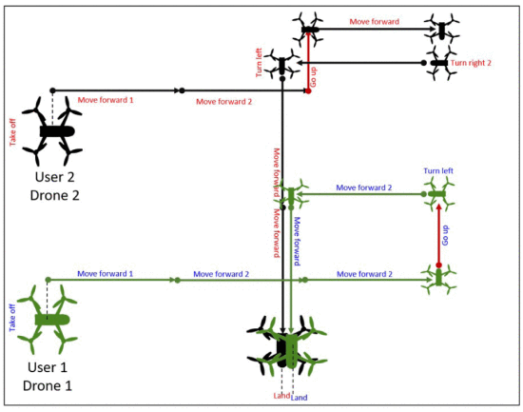


Figure 11 Positioning of User 1 and User 2 (Al-Nuaimi, F. A. et al. 2020)

Ultimately, Al Nuaimi, F. A. et al (2020) claim that their experiment was a success, by merit of User 1 being able to intercept User 2 with faster response times.

Al Nuaimi, F. A. et al (2020) present a detailed account of their experimental method, discussing the equipment used and BCI methods implemented to achieve these results, making the results testable and provable.

While the research presented by Al Nuaimi, F. A. et al. (2020) demonstrates a viable method of BCI control, there are a few specific parameters which are not addressed and would require further research.

Ji-Hoon, J. et al. (2020) proposed a novel method of controlling multiple drones. Their method of controlling a drone swarm (3 or more drones) using an EEG signal reader and a 4-class visual imagery paradigm.

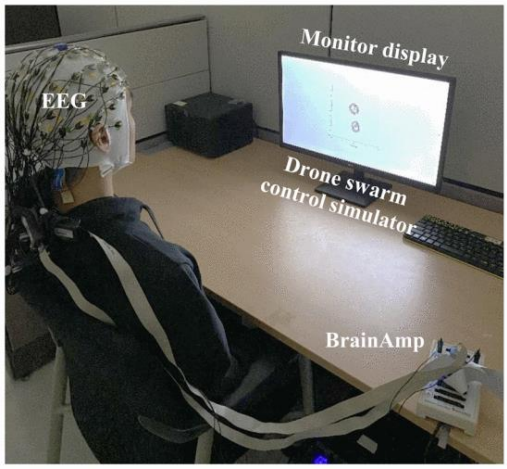


Figure 12 Experimental environment (Ji-Hoon, J. et al. 2020)

Ji-Hoon, J. et al. (2020) reported their findings as following; the average classification of accuracy across all four scenarios is 36.7% across all 7 test subjects. They report that this accuracy is higher than chance level, which was estimated to be around 25%.

To conclude, Ji-Hoon, J. et al. (2020) state that while their research shows that it is possible to control a swarm of drones effectively using EEG based BCI, more work must be done regarding visual imagery training in order to yield better results.

Ji-Hoon, J. et al. (2020) present an experimental method which has been explained clearly, with details of the number of participants, the hardware/software used, and the environment in which the experiment would be carried out.

It is suggested by Ji-Hoon, J. et al. (2020) that accuracy of control may be improved by the inclusion of more stimulating visual feedback, such as the presence of real drones as oppose to small digital renditions. It is possible that this visual feedback could be provided by VR, as suggested by Juliano, J. M. et al. (2020).

Wang, M. et al. (2018) created a method of drone control which incorporated an SSVEP-based BCI with a head-mounted device (HMD) which would administer the visual feedback necessary for control.

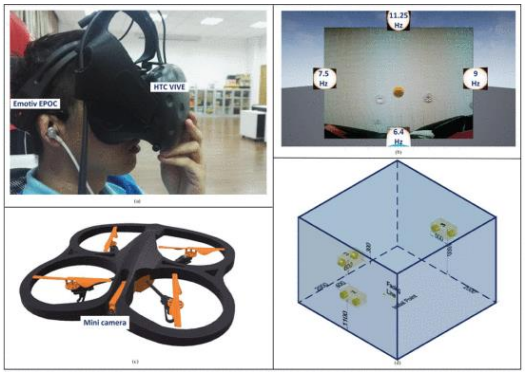


Figure 13 Experimental setup (Wang, M. et al. 2018)

Wang, M. et al. (2018) conducted two experiments, which they refer to as Experiment I and Experiment II. In Experiment II, Wang, M. et al. (2018) recruited the five best performing participants from the first experiment to control the quadcopter in a physical environment. Each subject participated in 12 trials.

Reporting their results for Experiment II, Wang, M. et al. (2018) state that each subject's results were broken into success rate, time efficiency, and information transfer rate. The researcher declares that the evidence shows that on average, participants could perform a "rather smooth and optimal flight toward the target in physical 3D space." (Wang, M. et al. 2018).

Wang, M. et al. (2018) state that coupling a HMD with SSVEP-based BCI system could be an effective method of BCI control, especially in relation to quadcopter control.

Wang, M. et al. (2018) collected evidence using a scientifically sound method. Multiple participants, who were given proper training on how to use the BCI system, completed the trials multiple times.

The results reported by Wang, M. et al. (2018) are some of the most promising in terms of high success rate out of all the BCI methods discussed thus far. This method could potentially be hybridized with the method proposed by Ji-Hoon, J. et al. (2020), and Juliano, J. M. et al. (2020) to improve the accuracy and performance of BCI users when controlling a drone/drone swarm.

2.3 Alternative BCI control methods

Abiri, R. et al. (2020) analyzed user acceptance towards BCI platforms, using a questionnaire design based on common usability questionnaires such as NASA Tax Load Index, Quebec User Evaluation of Satisfaction with assistive Technology.

To conduct their research, Abiri, R. et al. (2020) collected 28 healthy test subjects who were naïve to BCI systems and have no known neurophysiological conditions. The pre-interview questionnaire asks the participant how well they can visualize movements, as well as how they perceive their attention span to be.

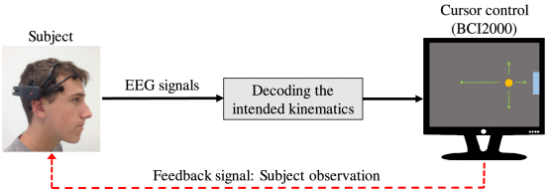


Figure 14 Schematic of the EEG-based BCI platform (Abiri, R. et al. 2020)

After completing the BCI test, the participants were given the post-interview questionnaire. "The questions aimed to evaluate different confounding variables, including the participant's level of energy, level of focus, and self-perception of control over the computer cursor movement." (Abiri, R. et al. 2020). Details of the questions asked can be found in figure 15.

Index	Question
1	How would you describe your energy level during this experiment? Choose one: Far below average, Moderately below average, Slightly below average, Average, Slightly above average, Moderately above average, Far above average.
2	How would you describe your focus level during this experiment? (Response options are the same as in question 1)
3	How would you describe the length of this experiment? Choose one: Far too short, Moderately too short, Slightly too short, Neither too short nor too long, Slightly too long, Moderately too long, Far too long.
4	How would you rate the level of controllability over cursor you had during the experiment (overall rate, individual target control)? Choose one: Extremely bad, Moderately bad, Slightly bad, Neither bad nor good, Slightly good, Moderately good, Extremely good.
5	Did you feel that eye movement had any effect on your ability to control the cursor? Choose one: Definitely yes, Probably yes, Might or might not, Probably not, Definitely not.
6	If you visualized hand movement, how difficult would you say this visualization was to maintain? Choose one: Extremely challenging, Very challenging, Moderately challenging, Slightly challenging, Not challenging at all.
7	How do you feel physically after the experiment? Choose one: Extremely uncomfortable, Moderately unconformable, Slightly uncomfortable, Neither uncomfortable nor comfortable, Slightly comfortable, Moderately comfortable, Extremely comfortable.
8	Describe your level of comfort during the test in relation to the lab environment and headset? (Response options are the same as in question 7)

Figure 15 Questions relating to selected confounding variables in the post-interview questionnaire (Abiri, R. et al. 2020)

In their results, Abiri, R. et al. (2020) report that there is some correlation between answers on the pre-interview and post-interview questionnaires and performance using the BCI system. For instance, there is a noted positive correlation between self-perceived visualization ability and accuracy in controlling the BCI system.

Furthermore, those with higher attention spans were more likely to hit the target, making them more accurate. The data for these results can be seen in figures 16 and 17.

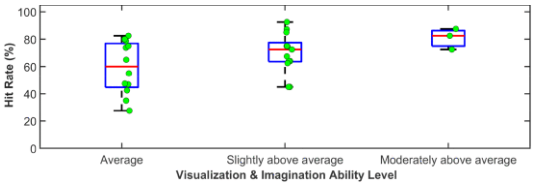


Figure 16 Boxplot of Hit Rate according to perceived visualization (Abiri, R. et al. 2020)

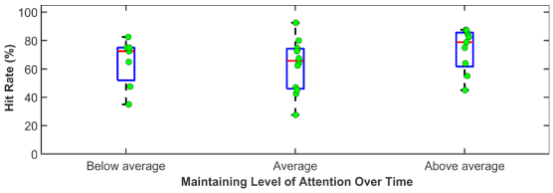


Figure 17 Boxplot of Hit Rate according to perceived attention span (Abiri, R. et al. 2020)

As a conclusion, Abiri, R. et al. (2020) state that participants with higher ability to visualize imaginary movement will typically perform better using BCI as indicated from their experiment.

Abiri, R. et al. (2020) built their questionnaire from the basis of pre-existing and well-respected questionnaires, giving a strong and proven foundation for their data collection method. The experiment they conducted was simple and participants were given adequate training time to reduce the chance of BCI naiveite introducing a bias into the results. The claims made by Abiri, R. et al. (2020) also seem to closely match the results they presented, and their conclusions are supported by this evidence.

From this work it could be suggested that proper visualization training, and attention holding training, may be useful in helping subjects learn to use BCI systems, especially if the user is struggling to control the device.

Leinders, S. et al. (2020) tested the feasibility of using the dorsolateral prefrontal cortex to retrieve signals to be used in a BCI system for people suffering from severe motor issues. This research sought to use an intrusive BCI system which read signals directly from an interface with the user's brain.

For their experiments, Leinders, S. et al. (2020) recruited two patients suffering from locked-in

syndrome who gave their informed consent to be part of the experiment.

"During home visits, participants routinely performed two working memory tasks. Research tasks were developed in-house, based on BCI2000 software." (Leinders, S. et al. 2020). The participants were tested once every 6 to 12 weeks, in which all 6 bipolar electrode combinations in one session.

Reporting on their results, Leinders, S. et al. (2020) determine that electrode pair e8-e10 is the most optimal electrode pairing, as it was recorded most frequently during activities.

As a conclusion to their research, Leinders, S. et al. (2020) recommend that further research should be carried out regarding the activation of dIPFC-based BCI systems without cues to allow patients to control their BCI system without the need for a visual setup.

Leinders, S. et al. (2020) conducted a thorough investigation over an extended period, attempting multiple different pairings of electrode and reporting the suitability of each. The results presented by Leinders, S. et al. (2020) appear to be valid and scientifically accurate.

Xu, M. et al. (2020) propose a hybrid BCI method which would allow for over 100 command codes when in use. The method involves a hybrid P300-SSVEP BCI system which incorporates steady-state visual stimulus (SSVS) to create a BCI system that has 108 instructions.

For their experiment, Xu, M. et al. (2020) used a test group of 10 participants (7 males and 3 females, aged between 21-26) with normal eyesight and no known neurological conditions. The participants were shown a series of stimuli, represented by 108 black characters which were divided into 3 x 3 matrices as seen in figure 18.

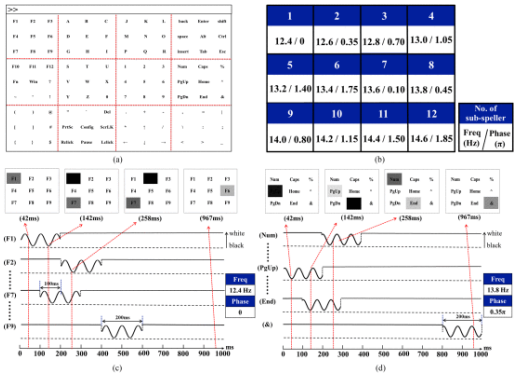


Figure 18 Hybrid BCI speller (Xu, M. et al. 2020)

When reporting their findings, Xu, M. et al. (2020) report that "Specifically, compared to the single SSVEP feature, the hybrid EEG features had an improvement of 7.87%, 4.63%, 2.77%, 2.50% and 1.48% on average at 1 to 5 rounds."

To conclude their research, Xu, M. et al. (2020) state that their research proves the effectiveness of their new hybrid BCI system. They claim that their system has the largest instruction set currently known.

The research conducted by Xu, M. et al. (2020) appears to be valid due to the large amount of detailed evidence presented. The methodology is easy to replicate, and steps were taken to remove bias, such as giving participants proper training into the use of the BCI system before beginning. Xu, M. et al. (2020) compare their research to existing datasets to justify their results.

The method proposed by Xu, M. et al. (2020) could be used in tandem with all other methods discussed in this paper, adding greater functionality to any device controlled by the BCI system.

3 Conclusions

Some specific recommendations for future research would be the implementation of the method proposed by Xu, M. et al. (2020) into any of the other discussed papers. For instance, coupling this method with a BCI enabled prosthetic may allow for greater range of movement and an overall more articulate prosthetic.

Implementing this method may not be difficult, as Xu, M. et al. (2020) use a method like those used by others mentioned above (an EEG headset used with SSVEP methodology). This similarity should allow for the method to be implemented without drastically overhauling the other methods.

By combining the methods of Wang, M. et al. (2020), Ji-Hoon, J. et al. (2020), and Juliano, J. M. et al. (2020) it could be possible to allow users to control multiple objects with a high degree of accuracy. While these researchers focused on drone control, it is possible that the same method could be applied to many objects across the internet of things. For example, issuing commands on a computer to allow those suffering from motor issues to properly interface with web browsers and other applications.

Furthermore, through the research of Leinders, S. et al. (2020), we understand the best practices for installing intrusive BCI systems. These systems can provide more accurate and direct signal readings than external BCI systems such as EEG. This should be the next step forward in further improving the accuracy of the other methods of control.

References

Abiri, R., Borhani, S., Kilmarx, J., Esterwood, C., Jiang, Y., & Zhao, X. 2020. 'A Usability Study of Low-Cost Wireless Brain-Computer Interface for Cursor Control Using Online Linear Model.' *IEEE Transactions on Human-Machine Systems*, 287-297.

Al-Nuaimi, F. A., Al-Nuami, R. J., Al-Dhaheri, S. S., Ouhbi, S., & Belkacem, A. N. 2020. 'Mind Drone Chasing Using EEG-based Brain Computer Interface.' *2020 16th International Conference on Intelligent Environments* (IE) (pp. 74-79). Madrid: IEEE.

Chan, A., & Dascalu, S. 2017. 'Using Brain Computer Interface Technology in Connection with Google Street View.' *2017 21st International Conference on Control Systems and Computer Science (CSCS)* (pp. 571-576). Bucharest: IEEE.

Coogan, C. G., & He, B. 2018. 'Brain-Computer Interface Control in a Virtual Reality Environment and Applications for the Internet of Things.' *IEEE Access Vol. 6*, 10840-10849.

Jaehoon, C., & Sungho, J. 2020. 'Application of Hybrid Brain-Computer Interface with Augmented Reality on Quadcopter Control.' *2020 8th International Winter Conference on Brain-Computer Interface (BCI)* (pp. 1-5). Gangwon: IEEE.

Ji-Hoon, J., Dae-Hyeok, L., Hyung-Ju, A., & Seong-Whan, L. 2020. 'Towards Brain-Computer Interfaces for Drone Swarm Control.' *2020 8th International Winter Conference on Brain-Computer Interface (BCI)* (pp. 1-4). Gangwon: IEEE.

Juliano, J., Spicer, R., Vourvopoulos, A., Lefebvre, S., Jann, K., Ard, T., . . . Liew, S.-L. 2020. 'Embodiment Is Related to Better Performance on a Brain–Computer Interface in Immersive Virtual Reality: A Pilot Study.' *Sensors 2020 vol. 20 Issue 4*, 1204.

Kosmyna, N. 2019. 'Brain-Computer Interfaces in the Wild: Lessons Learned from a Large-Scale Deployment.' *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC) Systems* (pp. 4161-4168). Bari: IEEE.

Leinders, S., Vansteensel, M. J., Branco, M. P., Freudenburg, Z. V., Pels, E. G., Van der Vijgh, B., Aarnoutse, E. J. 2020. 'Dorsolateral prefrontal cortex-based control with an implanted brain–computer interface.' *Scientific Reports Vol. 10 Iss. 1*, 1-10.

Meng, J., Streitz, T., Gulachek, N., Suma, D., & He, B. 2018. 'Three-Dimensional Brain–Computer Interface Control Through Simultaneous Overt Spatial Attentional and Motor Imagery Tasks.' *IEEE Transactions on Biomedical Engineering IEEE Trans. Biomed. Eng. Biomedical Engineering*, 2417-2427.

Wang, M., Li, R., Zhang, R., Guangye, L., & Zhang, D. 2018. 'A Wearable SSVEP-Based BCI System for Quadcopter Control Using Head-Mounted Device.' *IEEE Access*, 26789-26798.

Xu, M., Han, J., Wang, Y., Jung, T., & Ming, D. 2020. 'Implementing Over 100 Command Codes for a High-Speed Hybrid Brain-Computer Interface Using Concurrent P300 and SSVEP Features.' *IEEE Transactions on Biomedical Engineering*, 3073-3082.

Critical Evaluation of Current Load-balancing Schemes in Software-defined Networks for Improving Network Performance

Michal Skvarek

Abstract

With the increasing growth of modern networks, software-defined networking has become to be a promising new paradigm in today's networks. However, there are present issues with network performance and load redistribution that need addressing, and many projects were done trying to solve this issue. This research paper analyses part of these papers about improving load-balancing network performance in terms of link-utilization, control-plane or both and compares the effectiveness of proposed methods in different environments. At the end of this paper, conclusions are presented showing limitations of our current knowledge and which solution has the biggest potential solving the problem in most environments in comparison to others.

1 Introduction

The new promising paradigms for the future of the internet are considered to be used of Software-defined networking thanks to its capabilities. However, with the growth and usage of modern networks, one of the major challenges in this promising approach is an uneven distribution of traffic load in controllers which leads to degrading system performance and denial of service to users for their required services (Priyadarsinia M. et.al., 2019).

Different research was conducted trying to solve these issues, for example, Sun P. et.al. (2020) proposed their dynamic controller workload balancing scheme MARVEL working on multi-agent reinforcement learning rather than focus on optimizing.

Other methods were also presented as showed in the following mentions. Ejaz S. et.al. (2019) proposed a traffic-load balancing mechanism utilizing SDN controller as VNF in SDN-enabled networks by adding a secondary vSDN controller which is a duplicate of the original one. Their proposed system led to increased system performance.

Cui J. et.al. (2018) focused their work on SMCLBRT, a load-balancing strategy of multiple SDN controllers based on response time. Their claim made in this project is selecting an appropriate time threshold that can solve the load-balancing problem in SDN.

The aim of this survey paper will focus on critical evaluation and analysis of current load-balancing schemes used in software-defined networks to improve network performance. With a focus on research papers solving the problems in areas of link-utilization, control-plane load-balancing. Although, the related research mentioned presents good solutions. They do not fit into the category the main body if this paper is focused on. This paper will discuss methods used, experiments conducted, and their results while reaching conclusions and comparison regarding their effectiveness of improving network performance.

2 Evaluation of Current Load-balancing Technics in SDN

This section will be focused on Load-balancing schemes for achieving improved network performance in SDN.

2.1 Link Utilization Load-balancing

Attarha S. et.al (2017) researched developing an algorithm where network state is monitored by calculating link utilization periodically. The proposed algorithm was designed to avoid congestions in SDN to reduce network overhead

by rerouting a minimum number of flows. The controller anticipates congestion on the link and calculates the amount of load that needs to be transferred whenever link utilization surpasses the rate threshold. The flows are shifted to a path not congested by adding these flows.

The reported experimental setup by Attarha S. et.al (2017) was built on utilizing Mininet to create the testing topology. The authors have chosen ONOS as the controller platform for running the proposed algorithm. The experiment compared packet loss and throughput parameters as performance metrics with and without a congestion-aware algorithm.

The results gained from experimenting are shown in Fig. 1 and Fig. 2. Reported results showed increased performance while running the Congestion aware algorithm.

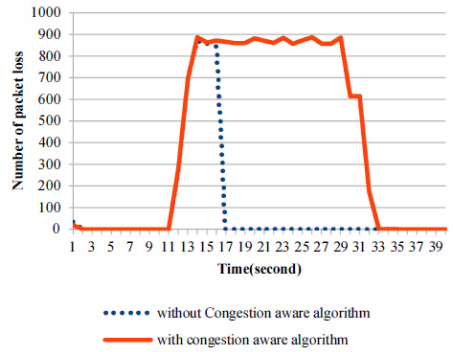


Fig. 1 Packet loss rate (Attarha S. et.al 2017)

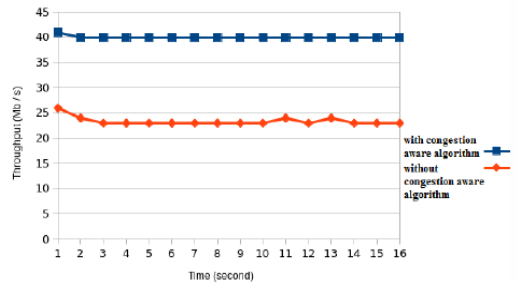


Fig. 2 Throughput with newly arrived flows (Attarha S. et.al 2017)

Based on the results Attarha S. et.al (2017) concluded that the prosed algorithm chooses a minimum number of flows as possible to resolve congestions and has a short running time and it is efficient. Furthermore, experiment results show improvement in packet loss and throughput.

The methodology mechanics were well designed with a clear objective on how to achieve improved performance by using load-balancing in SDN. However, the experiment proposed is not taking into consideration other factors as delay and response time. The method could have been compared with other load-balancing schemes to well justify results. In conclusion, more data should be present to ensure the method is not underperforming in other areas.

Dewanto R. et.al (2018) presented in their research an SDN-based ECMP program designed to avoid network congestions. By measuring the bandwidth available of every available path in advance. The program would send traffic through a non-overlapping path when possible. Dijkstra's widest path algorithm was used instead of max-min remainder capacity (MMRCS). After the path with the highest available bandwidth is found switches involved with the flow would update their flow table.

The experimental setup built by Dewanto R. et.al (2018) used Mininet tool to create simulated network topology based on fat-tree design and used Ryu controller platform in the simulated topology.

The result reported in the work of Dewanto R. et.al (2018) showed 8.7% lower packet, 14.21% higher throughput loss, and 92.27% lower delay was achieved against standard ECMP during congestion time. With a trade of 3.46% higher delay and 0.19% lower throughput in the same comparison during the non-congestion time. The proposed scheme showed improvement in comparison to round-robin by having 80.56 higher throughput, 75.2% lower LSD value, and 24.54% less packet loss.

Based on results obtained Dewanto R. et.al (2018) concluded that their proposed scheme improves the balancing capability of the system. The authors will focus their work on minimizing the trade-off mentioned during the non-congestion time.

The methodology of the scheme presented was tested in a good structured experimental setup and different scenarios were taken into consideration. A sound evaluation was performed, and the authors reported a slight limitation of their proposed method. Therefore,

making their claims well justified. However, further work is advised to remove the limitation mentioned.

Zhang S. et.al (2018) proposed in their work an online controller load balancing (OCLB) to address the issue with distributing loads among controllers. “An OCLB algorithm is designed based on derived optimality and termination conditions of switch migration.” (Zhang S. et.al 2018). The mechanism is based on real-time request distribution to minimize average controller response time is depending on the load of the controllers. The OCLB algorithm is made of two phases and thus detection and execution.

Zhang S. et.al (2018) conducted simulations implemented in python based on data center network scenarios to get dynamic flow fluctuations. The topology used is a Fat-tree topology as it is widely used in data centers. Controllers were deployed on hosts connected to edge switches. Three simulations were conducted for testing Load Balancing, Online Execution, and Convergence Rate. The scheme is compared with Optimal and Greedy scheme. Optimal uses Gurobi optimizer while Greedy scheme where controllers tend to migrate the heaviest switch to the lightest controller.

The results showed that the proposed scheme is 5.79% faster than Greedy scheme and only 0.04% slower on average than optimal response time and this result is still within its goal of reducing the response time. Moreover, the results showed as in Fig. 3 online scheme can react to changes of request distribution in time. Results in Fig. 4 showed switch migrations grow linearly with a growing number of switches demonstrating scalability functions.

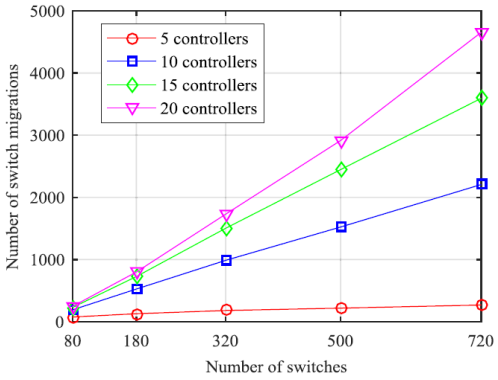


Fig. 3 Number of switch migrations needed to achieve load balancing, under different system settings. (Zhang S. et.al 2018)

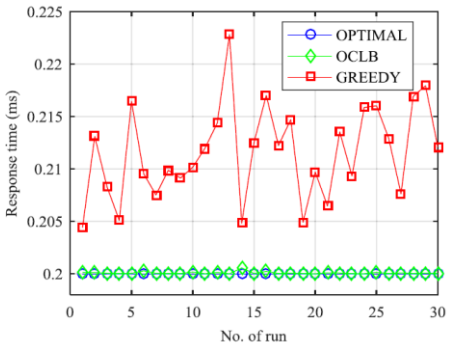


Fig. 4 Average response time of control plane. (Zhang S. et.al 2018)

Based on the results gained Zhang S. et.al (2018) Concluded that their proposed algorithm OCLB has met its goal as mentioned previously.

The research done by Zhang S. et.al (2018) proposed methodology is rigorous as it was tested in multiple scenarios using fat-three topology for realistic results. The results were compared against other methods tested in the same environment and undergoing the same scenarios. Therefore, the results are valid and proving the proposed scheme can improve network performance.

Askar S. (2016) proposed an adaptive load-balancing scheme for data center networks utilizing SDN utilizing Fat-tree network topology. The purposed scheme objective was to adaptively balance the load of DCN based on bandwidth and packet loss triggering parameter thresholds. The load-balancing was done by a centralized controller which had an entire overview of

network resources. The algorithm finds alternative paths for the traffic with reduces throughput or high packet loss rate.

Experimental setup Askar S. (2016) decided to utilize a virtual environment using Mininet tool and make use of MiniEdit for GUI utilization. Furthermore, OpenFlow switches were used for the experiment, and the controller was deployed by making use of POX Controller as a platform for the proposed load-balancing algorithm. The experiment is structured upon two scenarios testing load-balancing efficiency when there are already established connections between two hosts and with new host joining.

The results of the proposed load-balancing scheme were compared against the traditional scheme conducted in the same virtualized environment. The proposed adaptive load-balancing scheme had increased performance by 81% in the first scenario and 15% to 31% in the second for packet loss rate. The algorithm had improved throughput in the first scenario and maintained overall throughput in the second scenario.

Askar S. (2016) concluded based on results that the proposed scheme has considerable superiority over a traditional load-balancing algorithm and the algorithm is ready for implementation in such networks is simple to implement.

The methodology is sound with enough detail about the operation provided. The experiment is using available technologies and the steps undertaken are well document ensuring high repeatability. The results showed that the proposed method is more efficient than the traditional scheme. In conclusion, with all evidence, this research is well justified, and no bias was introduced.

2.2 Control Plane load-balancing

Cimorelli F. et.al (2016) proposed a distributed load balancing algorithm based on a game theory that converges to Wardrop equilibrium. Designed to dynamically balance control traffic across SND cluster controllers. The presented algorithm has minimized latency and increasing cluster throughput. The authors stated that. Using this algorithm, switches decide which controller to use without the communication among themselves.

Cimorelli F. et.al (2016) tested the proposed algorithm using simulation utilizing the tool MATLAB ®. The algorithm was executed on every switch. Strategies were computed based on the latencies of controllers depended on their response times. Ten simulations have been conducted with every simulation repeated 10 times. In each simulation, the positions of switches were randomly assigned. The proposed algorithm has been compared with the static switch-controller association strategy.

The results from testing showed that the proposed algorithm outperforms the nearest controller strategy. Fig. 5 shows the results achieved.

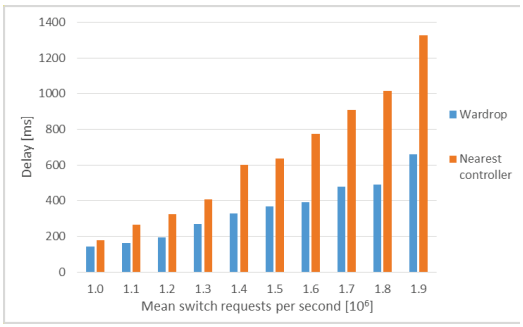


Fig. 5 Delays with Wardrop strategy and nearest controller strategy. (Cimorelli F. et.al 2016)

Based on these results the Cimorelli F. et.al (2016) concluded that the Wardrop strategy can counteract the non-homogeneous spatial distribution of switches and controllers with no explicit communication among them and their objectives have been met.

The proposed method was well designed and rigorous with results proving the claims. The method was tested using mathematical simulation conducted by a reliable tool making results viable. In conclusion, the research is valid, but a lab experiment would be advisable as a next step to fully justify the method.

Gasmelseed H. and Ramar R. (2018) proposed a traffic pattern-based load balancing algorithm. This algorithm was designed to handle TCP and UDP protocols and it is hosted on distributed SDN controllers. The algorithm's process was to

check and detect incoming traffic flows header to differentiate the TCP and UDP protocols. Decision-making and detection are showed in Fig. 6. After detection, traffic is sent to a dedicated controlled configured to handle the specific protocol distributing traffic to a server farm.

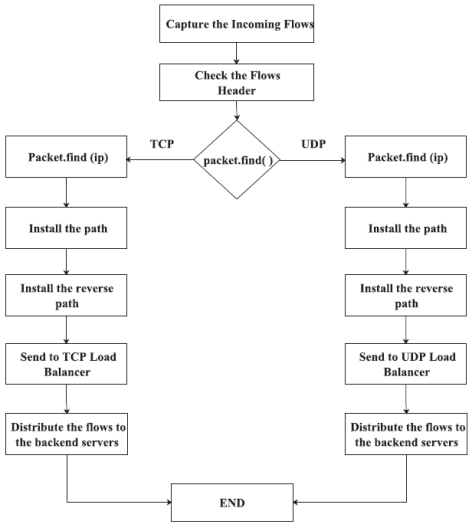


Fig. 6 Traffic pattern-based load-balancing algorithm flow chart (Gasmelseed H. and Ramar R. 2018)

In the experimental setup, Gasmelseed H. and Ramar R. (2018) deployed emulated SDN network based on distributed controller's architecture using Mininet and MiniEdit for the creation of topology. The experiment used 10 servers as backend servers and 4 controllers using POX controller as a platform to host the load balances application. The algorithm was tested against random, round-robin, and weighted round-robin algorithms.

The testing resulted that in comparison with other mentioned algorithms availability increased by 11%, throughput increased by 206%, packet loss was reduced by 86%, throughput increased by 206%, throughput increased by 206%, concurrency was reduced by 63%, and increased transaction rate by 258%.

From the results gained, Gasmelseed H. and Ramar R. (2018) concluded that their proposed algorithm proves its efficiency using distributed controller architecture in terms of the above-mentioned metrics against other mentioned algorithms.

The method proposed by Gasmelseed H. and Ramar R. (2018) provides a simpler solution for load-balancing in SDN. However, this research does not provide any information about the step taken for the experiment conducted only the experimental setup. Despite this, the authors provided results that were compared against other algorithms using the same environment and showing improved efficiency in the metrics defined. To conclude this research, the experiment is not repeatable without this information therefore the results are not justified.

2.3 Other Methods of Load-balancing

Zhong H. et.al (2017) proposed a load balancing scheme operating based on server response time (LBBSRT). The advantage of the SDN flexibility is also utilized. The proposed scheme is achieving evenly balanced server load and load-balancing by using the real-time response of every server measured by the controller. "Server response time directly reflects the server load capability." (Zhong H. et.al 2017).

As reported in the documentation by Zhong H. et.al (2017). The experimental setup was created by Open vSwitch. Furthermore, the floodlight controller was picked as an SDN controller. Three virtual machines with identical configurations were used as web servers to provide web service running WordPress to build a blog. Two different access frequencies were set up to send an HTTP request to servers. One is continuous and another sends a request every 2 seconds. Topology is utilizing 30 clients to access the server for real-life scenario mitigation.

The results gained from running the experiment showed the average server's response time of the three schemes and thus Round Robin, Random, and LBBSRT are 1.236s, 1.366s, and 1.119s, respectively.

Based on evaluation of these results the Zhong H. et.al (2017) concluded that their proposed scheme achieves better load balancing results against standardly used Round Robin and Random schemes.

The newly proposed method presented by Zhong H. et.al (2017) was well justified as the authors

took into consideration traffic level fluctuations to a certain level. A comparison of the results against other load-balancing algorithms proves their validity. In conclusion, the research is well justified, and the proposed algorithm provides improved SDN performance.

Wang H. et.al (2017) mentioned problems in SDN with load-balancing both with links and controllers and proposed a rounding-based algorithm (RDMAR) to address the mentioned problems. The purpose was to decrease link utilization and controller response time both simultaneously. The algorithm is utilizing area-load bounds and controller-load bounds which are used for routing decisions.

The testing environment builds by Wang H. et.al (2017) used Mininet to create a VL2 and Fat-tree topology. The RDMAR algorithm was evaluated by comparing 5 other benchmarks. In addition, emergency accidents affecting traffic were added. Testing was conducted to get results on Link Load performance, Controller response time, Impact of area-load bound, and running time performance.

The results showed RDMAR can reduce maximum link load by 37% and 17% respectfully in the two topologies. Controller response time was reduced ranging between 30% to 70% in comparison with other benchmarks. RDMAR reduced link-load by 12% in comparison with RDMAR-F. Finally, RDMAR requires by average 0.19s while NOX-MT about 0.6s to make a routing decision.

Wang H. et.al (2017) concluded that the results showed prove the efficiency of the proposed algorithm. The simulation showed that the algorithm largely reduced controller response time while achieving a similar performance of link load balancing in comparison with existing solutions.

The proposed method is significant in the way it covers both problems with link-utilization and control plane to achieve improved load-balancing for SDN. A good testing plan was conducted comparing the proposed approach with other used technologies. Test results are backed up by running the experiment in two different topologies environments. The authors conclusion is well justified.

3 Comparison of Load-balancing Techniques to Achieve Improved SDN Performance

In comparison the Link-utilization method proposed by Zhang S. et.al (2018) would be suitable in a large network environment as it is scalable and works with distributed controllers sharing workload which gives it this ability. In comparison to the method designed by Askar S. (2016) works with one controller having an entire network overview requiring a lot of computing resources making it better suited to medium and small networks.

Cimorelli F. (2016) method can theoretically achieve the best performance in large networks as it is utilizing many controllers and clusters thanks to its ability to balance control traffic between them. The larger the network the better its ability is utilized. In comparison with other algorithms mentioned in this paper it is scalable by core.

Zhong H. et.al (2017) proposed algorithm is best suitable for data centers as the algorithm operates on server response time. In other networks without servers this algorithm therefore could note use its full potential. On the other hand, Wang H. et.al (2017) method has a general application and deals with both link and control plane problems making it suitable to general networks rather than datacentres.

4 Conclusions

Possible approaches to improve network performance with load-balancing for SND have been analyzed and compared. the review of the current literature showed many researchers came up with various methods to solve the problem.

Zhang S et.al (2016) together Cimorelli F. (2016) proposed methods offered good solution to large networks with possibility to scale and keep the performance.

Askar S. (2016) solution offered great performance improvement. However, the solution might underperform in larger networks and the scaling solution was not present.

Although, Zhong H. et.al (2017) solution was designed to work in server hosting networks. The

combination with another method could bring a solution working with other devices as well.

The comparison between Attarha S. et.al (2017) and Dewanto R. et.al (2018) works showed that although the methodologies are sound and valid, their methods would be valid for small to medium-sized SDN networks. The load on a centralized controller might be a challenge in a large data center network.

Wang H. et.al (2017) method is best suitable from all the others as it is scalable and offers solutions both with link-utilization and control-plane issues to increase network performance. As the most prosing solution, real word application testing should be conducted to prove its capabilities.

References

Askar S, 2017, 'SDN-Based Load Balancing Scheme for Fat-Tree Data Center Networks', *Al-Nahrain Journal for Engineering Sciences (NJES)*, Vol. 20 No.5, pages 1047-1056

Attarha S., Hosseiny K. H., Mirjalily G., Mizanian K., 2017, 'A Load Balanced Congestion Aware Routing Mechanism for Software Defined Networks', *25th Iranian Conference on Electrical Engineering, pages* 2206-2210

Cimorelli F., Priscoli F. D., Pietrabissa A., Celsi L. R., Suraci V., Zuccaro L., 2016, 'A Distributed Load Balancing Algorithm for the Control Plane in Software Defined Networking', *24th Mediterranean Conference on Control and Automation (MED)*, pages 1033-1040

Cui J., Lu Q., Zhong H.,Tian M., and Liu L., 2018, 'A Load-Balancing Mechanism for Distributed SDN Control Plane Using Response Time,' in *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1197-1206

Dewanto R., Munadi R., Negara R. M., 2018, 'Improved Load Balancing on Software Defined Network based Equal Cost Multipath Routing in Data Center Network', *Journal Infotel*, Vol 10, Iss 3, pages 157-162

Ejaz S., Iqbal Z., Shah P. A., Bukhari B. H., Ali A., and Aadil F., 2019, 'Traffic Load Balancing Using Software Defined Networking (SDN) Controller as Virtualized Network Function', in *IEEE Access*, vol. 7, pages 46646-46658

Gasmelseed H., Ramar R., 2018, 'Traffic pattern–based load-balancing algorithm in software- defined network using distributed controllers', *International Journal of Communication Systems*, Volume 32, Issue 17, e3841

Zhong H., Fang Y., Cui J., 2017, 'LBBSRT: An efficient SDN load balancing scheme based on server response time', *Future Generation Computer Systems*, Vol. 68, pages 183-190

Sun P., Guo Z., Wang G., Lan J., Hu Y., 2020, 'MARVEL: Enabling controller load balancing in software-defined networks with multi-agent reinforcement learning', Computer Networks Journal, Vol. 177, 107230,

Wang H., Xu H., Huang L, Wang J, Yang X, 2018, 'Load-balancing routing in software defined networks with multiple controllers', *Computer Networks Journal*, Vol. 141, pages 82-91

Zhang S., Lan J., Sun P., and Jiang Y., 2018, 'Online Load Balancing for Distributed Control Plane in Software-Defined Data Center Network', *IEEE Access*, Vol. 6, pages 18184 – 18191

Evaluating Machine Learning Approaches to Car Detection using Unmanned Aerial Vehicle Imagery

Thabang Fenge Isaka

Abstract

Due to the rampant evolution in car manufacturing worldwide, the need for real time surveillance and traffic monitoring has become a great necessity. Therefore, Unmanned Ariel Vehicles (UAV) have become a cost effective and felicitous strategy to capture quality aerial images. This paper critically evaluates current approaches established by different researchers to detect cars in live coverage using computer vision techniques. The discussion will also entail a conclusion of the best method found which was the Deep Vehicle Counting Framework. Moreover, this research will confer significant strategies derived from current research in order to improve on methodologies towards developing more robust vehicle monitoring systems under harsh climatic conditions.

1 Introduction

The ability to precisely detect, track, count and analyze cars from high and low altitudes has become a pain point in the sector of road safety, traffic monitoring and highway infrastructure management (Outay et al, 2020).

According to the World Health Organization (2020) approximately 1.35 billion people die due to car accidents each year and these traffic accidents cost most countries 3% percent of their Gross Domestic Product (GDP).

As such the immense growth in computer vision models that are built using machine learning algorithms have made themselves known to be competent in addressing the challenges found in the transportation sector with an ultimate goal of saving lives. Thus, this paper is of critical importance, as it will relay Deep learning solutions based on current research in order to contribute to the knowledge base of technologies that aim to reduce mortality rates associated with road accidents.

Nguyen (2019) explored this ability by building an enhanced Faster Region Based Convolutional Neural Networks (Faster R-CNN) framework with an improved base network and classifier. The model encountered challenges in accurate car detection in instances such as traffic congestion, occluded cars, and vehicles moving under light variations.

On the other hand, Bazi et al. (2018) proposed a novel convolutional support vector machine network. Which capitalized on SVM classifiers mainly for feature map generation. The model detection accuracy is still being fine-tuned through exploring sophisticated architectures based on inception and residual layers used in recent Deep learning algorithms to make the model applicable to real-time traffic analysis.

Therefore, the economic and social problems faced by the transportation sector globally proves the need for more research in this area of car detection. Hence, this paper will contribute to this by analyzing deep learning algorithms in current research, mainly to detect, classify and track cars using UAV imagery. Each approach will be critically evaluated and compared, to find the most appropriate method based on accuracy and processing time, coupled with recommendations to enhance the detection accuracy in these deep neural network (DNN) architectures.

2 Current Car Detection Methods Using Deep Learning algorithms.

This section discusses current methods for car detection using different classification approaches. We will discuss deep learning methods; You Only Look Once (YOLO) version 2 and 3, Faster R-CNN, Convolutional Neural Networks, and a Deep Vehicle Counting Framework.

2.1 You Only Look Once (YOLO)

YOLO is an object detection system that analyses images only once using an image grid. YOLO is comprised of different variants.

Tang et al. (2017) took advantage of this object detection system by using YOLOv2 to create a model that detects cars from an image dataset captured using UAV's. To create this model Tang et al. (2017) proposed to use a Circulant Structure of Tracking-by Detection with Kernel (CSK) to help elucidate images of vehicles taken from basic scenes, due to few available manual annotations of cars in the actual UAV images.

The data collection processes involved images of cars captured by the UAV on platforms where a car is stationary and where vehicles are on the move. The model architecture consisted of 19 convolutional layers and 5 max pooling layers as shown in Figure 1.

recall rate under different Intersection Over Union (IoU) thresholds.

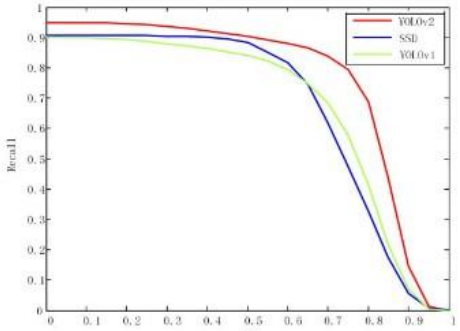


Figure 2: IoU thresholds against the recall rates. (Tang et al. 2017).

Results showed that with the cumulative IoU thresholds the recall curve of YOLOv2 increased more than other algorithms as seen in Figure 2 above.

Table 1 also shows results of YOLOv2 achieving the best performance in terms of Average Precision (AP). It was able to detect an image of 640 x 480 pixels in 0.048 seconds.

Method	mAP	Average running time per image (second)
SSD	72.95%	0.055
YOLO	67.99%	0.056
YOLOv2	77.12%	0.048

Table 1: Results AP & Running Time (Tang et al. 2017).

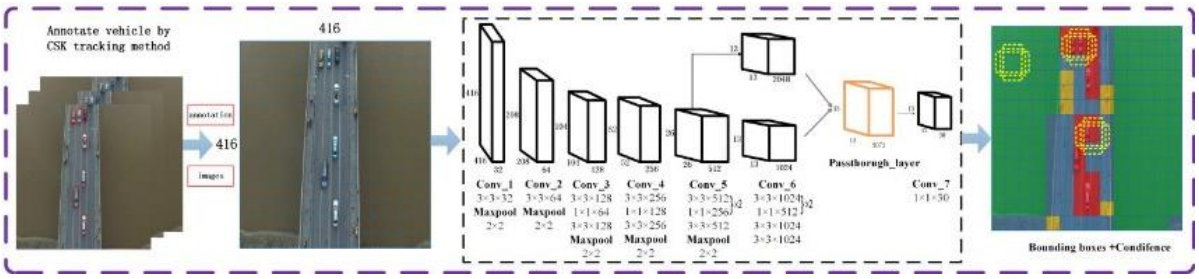


Figure 1: YOLO v2 architecture and model training process (Tang et al. 2017).

To perform the experiment the data used was divided into 20604 images for training, and 1374 images for testing. The experiment was carried following the deep learning framework café and executed on an intel i7 Pc NVIDIA GTX-1060 GPU (6 GB GPU memory), running an Ubuntu 14.04 operating system. The model was firstly evaluated with other algorithms to perceive its

Tang et al. (2017) concluded their claims by mentioning the effectiveness of their model being able to attain an accuracy of 77.12% in vehicle detection and its ability to spot vehicles in both simple and complex scenes. They mentioned also that their experiment followed the deep learning framework café which is widely recommended.

However, their experiment does not adhere to standards of a reproducible experiment – statements of hardware used were abstract since they did not mention the hardware specifications of the UAV used. Also, the altitudes of the images captured by the UAV are not stated at all, thus making it impossible for other researchers to perform a similar experiment like they did.

Following another instance of YOLO, Benjdira et al. (2019) used a more complex variant YOLOv3 and it is characterized by precise accuracy and supplants the softmax function with logistic regression and threshold. The model was made to take input images from a UAV and create grids that scale and generate a possibility map for successful car detection.

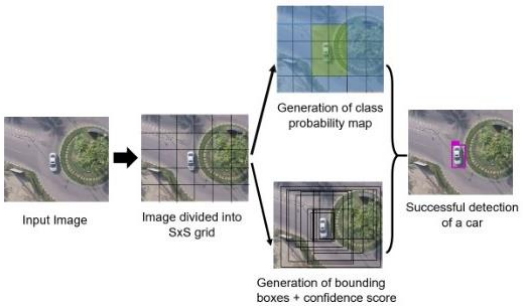


Figure 3: YOLO v3 architecture (Benjdira et al. 2019).

The experiment's dataset was built using UAV imagery separated into 218 training image sets with 3,365 occurrences of cars and 52 testing image sets with 737 instances of cars with a UAV hovering at an altitude of 55m and alternatively 80m above Prince Sultan University Campus. Benjdira et al. (2019) experiment was compared against another state-of-the-art deep learning algorithm - Faster R-CNN which attained a learning rate of 0.00019 during training whilst YOLOv3 training process was optimized using stochastic gradient descent with a momentum of 0.9 with a learning rate set to 0.001 and weight decay of 0.005. The computer used to run the models was a Linux (Ubuntu 16.04) OS using Intel Core i9-8950HK, Nvidia GTX 1080, 8GB GDDR5 and 32 GB RAM.

The evaluation metrics used to compare the two models was based on precision, Recall, F1 score, quality, and processing speed. According to the results obtained by Benjdira et al. (2019) both the algorithms attained high precision rates with YOLOv3 obtaining 99.73% as compared to 99.66% by Faster R-CNN. What distinguished the two algorithms was that YOLOv3 had a recall rate of 99.07% ahead of the 79.40% by the Faster R-CNN model. This can be seen with the illustrations on Figure 4 and Table 2 below.

Figure 4: YOLOv3 and Faster R-CNN car detection (Benjdira et al. 2019).

Measure	Faster R-CNN (test dataset)	YOLOv3 (test dataset)
TP (True positives)	578	751
FP (False positives)	2	2
FN (False negatives)	150	7
Precision (TPR)	99.66%	99.73%
Sensitivity (recall)	79.40%	99.07%
F1 Score	88.38%	99.94%
Quality	79.17%	98.81%
Processing time (Av. in ms)	1.39 s	0.057 ms

Table 2: YOLOv3 and Faster R-evaluation metrics (Benjdira et al. 2019).

The experiment conducted by Benjdira et al. (2019) follows good scientific practices as it portrayed no essence of bias. Both models were trained with the same data and processed with the same hardware. Moreover, their experiment is easily reproducible due to the clear information given about the equipment and data used in his experiments.

This is seen by the GitHub link provided to access the image dataset. Another thing to note concerning the experiment are the performance

metrics utilized to compare these methodologies. The metrics were purposefully used in the most relevant way for the setup. The turnaround time of their model, which is 0.057ms, is worth noting, making this model very applicable in vehicle tracking applications.

2.2 Faster R-CNN

Xu et al. (2017) used another deep learning algorithm, Faster R-CNN in a different context - using low altitude images. The Faster R-CNN architecture was such that, from the UAV images the model distinctively fetches multiple regions of interest (RoIs) as input. Thereafter a fixed-length feature vector is extricated by the RoI pooling layer issued by the convolutional layers. Feature vectors are then passed into a sequence of fully connected layers and finally pass through a region of softmax layer and bounding box regressor layer, hence outputs will include estimates of image backgrounds and object classes that contain a car.

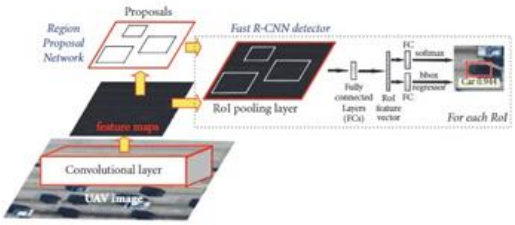


Figure 5: Faster R-CNN architecture (Xu et al 2017).

Experiment data was captured using a DJI Phantom 2 drone combined with a 3-axis stabilized gimbal. Images of cars were captured from different orientations and at flight heights of 100 - 150 meters. Thus, a total image dataset of 400 images containing 12,240 samples for training and 100 images comprising 3,115 samples for testing. The model was particularly trained with passenger cars. The computer used to run the model was Intel Core i7, Nvidia TITAN X, 12 GB GDDR5 and Ubuntu 14.04.

Model evaluation process was based on four indicators. These parameters were calculated using the evaluation formula. Where TP represents the number of true car objects detected and FB represents number of false non-car objects detected.

$$\text{Correctness} = \frac{TP}{TP + FP},$$

$$\text{Completeness} = \frac{TP}{TP + FN},$$

$$\text{Quality} = \frac{TP}{TP + FP + FN},$$

Formula 1: Method to calculate evaluation indicators (Xu et al 2017).

The model was compared against other algorithms namely ViBe, Frame difference, the AdaBoost method using Haar-like features and Linear SVM classifier with HOG features.

Results on Table 3 showed that Faster R-CNN achieved 94.94% in quality compared with the low percentages by other methods. Moreover, the ability portrayed by Faster R-CNN in learning image orientations, aspect ratios and scales during training lead to 98.43% correctness and 96.40% completeness attained. Unlike other methods which were unable to pass the 90% mark, due to sensitivity to cars only on plane rotation and specific scale variations.

Metrics	ViBe	Frame difference	V-J	HOG + SVM	Faster R-CNN
Correctness (%)	76.64%	78.17%	84.74%	84.33%	98.43%
Completeness (%)	38.65%	39.78%	41.89%	43.18%	96.40%
Quality (%)	34.58%	35.80%	38.96%	39.97%	94.94%
Detection speed (f/s)					
CPU mode	7.42	11.83	3.38	1.45	0.018
GPU mode	N/A	N/A	20.61	6.82	2.10

Table 3: Experiment results for Faster R-CNN and other algorithms (Xu et al 2017).

Xu et al (2017) further tested the robustness of his model in detecting cars under environments with dynamic illuminations. Under light changing environments Faster R-CNN achieved a completeness of 94.16% in Table 4 slightly lower than the 96.40 in Table 3.

Metrics	ViBe	Frame difference	V-J	HOG + SVM	Faster R-CNN
Correctness (%)	81.91%	80.15%	87.27%	88.45%	98.26%
Completeness (%)	67.90%	64.69%	81.36%	82.38%	94.16%
Quality (%)	59.05%	55.76%	72.73%	74.38%	92.61%

Table 4: Results of detection under illumination changing environments (Xu et al 2017).

The experiment carried out is well justified and follows good science. The experiment setup was well stated, as hardware for the UAV and computer used were asserted properly. Moreover, the test experiments revealed no form of biasness all algorithms received the same image dataset. Xu et al (2017) continued by doing another experiment to test the algorithms in terms of illuminating changing environments and dynamic image orientations to prove the accuracy of his proposed model. The results and procedures carried out proved the veracity of his model.

Although Faster R-CNN attained a significant accuracy level. This model will not be suitable in real life applications like traffic monitoring considering the 2.1 frames per second under GPU mode achieved by it. This shows that the model is slow in detection speed even when executed on powerful hardware.

2.3 Deep Vehicle Counting Framework

Zhu et al (2018) proposed an enhanced Deep Vehicle Counting Framework (DVCF). Model consists of two parts a deep learning vehicle detection with type identification and an Enhanced Single Shot Multibox Detector (Enhanced-SSD) to detect a variety of vehicles - trucks, buses, and motorcycles. The default VGG model was then replaced with a more powerful ResNet model tied with a restructuring of feature layers meant to improve detection rates as illustrated in Figure 6.

A DJI In-spire 1 Pro drone was used to capture the images during peak hours on a busy metropolis. The dataset was made of 17 168 image patches annotated correctly. A random 85% was used for training and 15% for validation. Model was tested against other deep learning algorithms YOLO, Faster R-CNN, and conventional SSD. The training process was done using the Caffe toolkit processed on a GTX 1080Ti GPU with 11 GB video memory.

Model	SSD	Faster-RCNN	YOLO	Enhanced-SSD
Batch size	32	32	32	6
Optimizer	SGD	SGD	SGD	SGD
Learning rate	0.001	0.001	0.001	0.001
Momentum	0.9	0.9	0.9	0.9
Epoch	12,000	12,000	60,000	12,000

Table 5: Model training parameters (Zhu et al 2018).

Experiment was based on two testing datasets - video and imagery. Evaluation metrics were centered around accurate counting and detection capabilities on reduced resolutions.

Method	TP (true positive) ↑	FP (false positive) ↓	FN (false negative) ↓	Correctness ↑	Completeness ↑	Quality ↑
Faster-RCNN	184	58	138	0.760	0.571	0.484
YOLO	158	1	174	**0.994**	0.476	0.474
SSD	287	8	45	0.973	0.864	0.844
Enhanced-SSD (ours)	**293**	7	**39**	0.977	**0.883**	**0.864**

Table 6: Test 1 Images - Counting results (Zhu et al 2018)

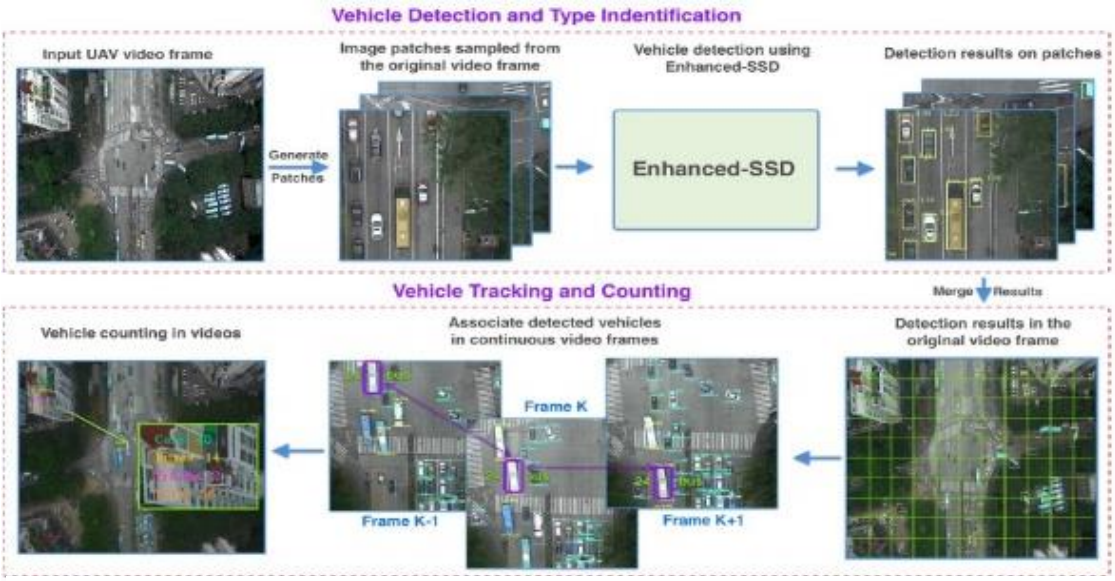


Figure 6: Deep Vehicle Counting Framework model architecture and flow (Zhu et al 2018).

Type/Method	Faster-RCNN	YOLO	SSD	Enhanced-SSD
Car	0.486	0.521	0.864	**0.866**
Bus	0.674	0.229	0.829	**0.892**
Truck	0.25	0.143	0.571	**0.80**

Table 7: Test 1 Images - Counting results based on vehicles (Zhu et al 2018)

Table 7 above showed that the Enhanced-SSD was able to accurately count car objects in both image-based datasets and peculiarly identify each vehicle type correctly. Hence, attaining high scores for quality and accuracy. The models were further analyzed with another testing phase that contained images with cars, trucks, buses, and any other possible vehicle. Where these objects are, they are annotated in light green, orange, and light blue bounding boxes as seen below in Figure 7.

This second experiment was performed on test images of high resolution to check how the Enhanced SSD method could work with other deep neural networks. Results portrayed the best counting accuracy on the testing set. Whereas others like YOLO and Faster R-CNN methods missed a lot of vehicles as seen below in Figure7.

Figure 7: Test 2 Videos results - True positives marked with green boxes. False positives are other colors (Zhu et al 2018).

Zhu et al (2018) concluded that his method portrays a level of reliability in real-life applications. Considering its efficiency and application to different forms of vehicles. Methods carried out in this research followed good science practices. The experiment dataset used follows the principles of a reliable dataset, such that the dataset composes of image and video formats made up of stationary and moving cars under varying weather conditions.

As such, this model is capable of being applied in vehicle tracking on more sophisticated technologies like satellites based on the depth of the neural network and accurate detection of varied vehicle type objects.

3 Methodology Comparisons

The methods proposed by the different authors in this paper all showed a significant approach to detecting cars using UAV images, but each have a unique attribute that made it better than the other making them comparable.

The YOLOv3 Benjdira et al. (2019) proposed had a noteworthy advantage over the YOLOv2 suggested by Tang et al. (2017). Due to the bounding box prediction used which enabled it to calculate objectness score for each bounding catapulted by a logistic regression, resulting in great accuracies. Moreover, In the context of accuracy and reliability YOLOv3 attained results ranging from 98 – 99%. Whereas YOLOv2 showed no competency in efficiency and reliability considering the 0.048 seconds turnaround time and 77.12% accuracy. Consequently, making it a slow and less efficient methodology that cannot be applied in real-time

applications like traffic analysis, whilst YOLOv3 can be used in real-time traffic analytics due to its magnificent 0.057 milliseconds turnaround time.

Xu et al. (2017) proposed a more sophisticated enhancement on the Faster R-CNN model. Training it to achieve an ability to detect cars under light changing environments with a completeness of 94.16%. This is a significant achievement and is very promising considering that in real life cars move under tunnels and trees making illuminations around them dynamic, unlike the YOLO variants that although can detect cars in complex scenes still struggled with images that had varied exposure and lighting.

Zhu et al (2018) Deep Vehicle Counting Framework showed competency than all other mentioned algorithms, since it is the only algorithm that was able to detect not only passenger cars but trucks, buses, and motorcycles. Which was something the YOLO variants and Faster R-CNN were unable to achieve. The framework did not only detect cars in image formats but was also able to spot vehicles in video formats making it the best method for live traffic coverage.

4 Recommendations

Car detection does not depend solely on an image-based point of view. There are other aspects that if converged with the image-based models suggested by the authors discussed, can make them much more accurate and sophisticated. Sangeetha et al (2019) introduced a novel model that utilized thermal cameras to capture cars with thermal emissions in rainy traffic scenarios and a background subtraction algorithm based on the Scilab (version 6.0.2) to detect the vehicles. Cao et al. (2019) on the other hand used an enhanced one-stage detector - ThermalDet to detect vehicles on various situations such as day and night the model was able to achieve 72.96% mean Average Precision. Likewise, Chang et al (2019) used a TOLO model with 3x3 Convolutional Neural Network layers and an additional Thermal Feature Enhancement (TFE) and attained a 97% precision in car detection.

These recommended models can be an enhancement to the current discussed techniques and other DNN algorithms making them robust to track traffic seamlessly in harsh weather conditions thus relevant authorities can take necessary precautions to safeguard lives.

5 Conclusions

The literature discussed in this paper contains state-of-the-art deep neural networks specifically for car detection. Which have proved that enhanced models are the most suitable for practical applications than plain basic models. This is analyzed looking at the YOLO variants proposed by the two authors Benjdira et al. (2019) and Tang et al. (2017) both models to become capable in accurately detecting cars they needed extra optimization and performance enhancements. But with all these enhancements detecting other forms of vehicles proved unsuccessful as the models could detect only passenger cars. We discussed Faster R-CNN model by Xu et al (2017) a very promising model that was able to detect cars in illuminating changing environments accurately but still needs fine tuning.

Over and above the best model was the Deep Vehicle Counting Framework by Zhu et al (2018) this model was the only one able to detect vehicles of all sorts - cars, buses, and motorcycles accurately regardless of image saturations and shadows on the UAV dataset then other discussed algorithms. It also had a greater turnaround time than other models making it the most suitable for real-time traffic analytics and car tracking.

References

Bazi, Y & Melgani, F 2018, ‘Convolutional SVM Networks for Object Detection in UAV Imagery’, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 6, pp. 3107–3118

Benjdira, B., Khursheed, T., Koubaa, A., Ammar, A. and Ouni, K. 2019, ‘Car Detection Using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3’, in *2019*

1st International Conference on Unmanned Vehicle Systems-Oman (UVS). IEEE, pp.1–6.

Cao, Y., Zhou, T., Zhu, X. and Su, Y. 2020, 'Every Feature Counts: An Improved One-Stage Detector in Thermal Imagery', in *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*. IEEE.

Chang, C.-W., Srinivasan, K., Chen, Y.-Y. and Cheng, W.-H. 2019, 'Vehicle Detection in Thermal Images Using Deep Neural Network', in *2018 IEEE Visual Communications and Image Processing (VCIP)*. IEEE.

Kadar A, Sangeetha N, Sathyanarayan K, 2019, 'Vehicle Detection using Thermal Sensors', *International journal of engineering research & technology (IJERT)*, vol. 08, Issue 06 (June 2019),

Nguyen, H., 2019, 'Improving Faster R-CNN Framework for Fast Vehicle Detection'. *Mathematical Problems in Engineering*, 2019, pp.1–11.

Outay, F, Mengash, HA & Adnan, M 2020, 'Applications of unmanned aerial vehicle (UAV) in road safety, traffic and highway infrastructure management: Recent advances and challenges', *Transportation Research Part A: Policy and Practice*, vol. 141, pp. 116–129.

Tang, T, Deng, Z, Zhou, S, Lei, L & Zou, H 2017, 'Fast Vehicle Detection in UAV Images', in 2017 *International Workshop on Remote Sensing with Intelligent Processing (RSIP)*, IEEE, pp. 1–5.

World Health Organization, 2020, *Road traffic injuries*. [online] Who.int. Available at: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries [Accessed 18 Oct. 2020].

Xu, Y., Yu, G., Wang, Y., Wu, X. and Ma, Y. , 2017, 'Car Detection from Low-Altitude UAV Imagery with the Faster R-CNN'. *Journal of Advanced Transportation*, 2017, pp.1–10.

Zhu, J., Sun, K., Jia, S., Li, Q., Hou, X., Lin, W., Liu, B. and Qiu, G., 2018, 'Urban Traffic Density Estimation Based on Ultrahigh-Resolution UAV Video and Deep Neural Network'. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(12), pp.4968–4981.

A Critical Analysis of Cloud Security Frameworks Aimed at Improving Data Security

Nsikakabasi-inam Akpan James Udia

Abstract

Cloud security has inarguably become an important aspect of the cloud that has gotten so much attention. With the cloud having substantial merits such as scalability, cost saving and availability to both small and large businesses, it also has major downsides with respect to data confidentiality, privacy, and integrity. This paper is focused on analyzing in-depth cloud security frameworks which were designed to improve data security in the cloud, these frameworks include: Ethereum Blockchain Technology, Genetic Algorithm and RSA and AES Encryption Algorithm. Finally, comparison of frameworks and experiments carried out are discussed throughout the paper, evaluation and conclusions are reached. Also, recommendations for future work will be provided.

1 Introduction

In recent years cloud insecurity has posed major challenges to businesses. These insecurities have resulted in data threats and breaches. In a recent study by Sophos, 96% of companies across 26 countries including United Kingdom have showed concern about the state of their cloud security. 70% of these companies experienced crypto jacking, data theft, malware, ransomware in their public cloud, 44% reported data loss and leakage (Sanhotran, R. 2020). This continuous cloud security challenge is on the rise due to the involvement of third-party vendors and components (Chawkia, E. B. et. al. 2018).

Li, J. et. al. (2019) proposed a scheme with constant resistance to both master key and secret key leakages, with continuous recovery capability based on the hierarchical attribute-based encryption (HABE) security model. Similarly, Deng, H. et. al. (2014) performed a research on cipher-text hierarchical (CP-HABE) model, the scheme utilizes linear confidential sharing method and delegation mechanisms with short cipher-texts. The scheme establishes the security of the scheme using three constant notions.

Arki, O. and Zitouni, A. (2018) proposed a cloud security scheme based on agents. The security model utilizes trust model, encryption method and integrity technique to ensure confidentiality, integrity, and privacy of cloud data. Research performed by Garg, P. and Sharma, V. (2014) demonstrates the use of Third-Party Auditor (TPA) with a combination of other cryptographic mechanisms such as RSA, DES and the hash function to enhance data integrity and security, although the use of Third-Party Auditors cannot be trusted fully as it can use data for personal gain. (Mall, S. and Saroj, S. K. 2018)

This survey paper will focus on analyzing several security strategies that has been proposed by researchers. It will analyze the frameworks and experiments from research done in improving cloud security. This paper will further compare these cloud security frameworks to find the most effective framework strategy in improving data security in the cloud.

2 Analysis of Cloud Security Framework Strategies

This section provides an in-depth analysis of proposed framework methods for preventing data insecurity. The analysis will be based on the following cloud security framework methodologies using (a). Blockchain Technology (b). Genetic Algorithm (c). RSA and AES Encryption Algorithms.

2.1 Blockchain Technology

Wang, S. et. al. (2019) proposed a decentralized security cloud storage framework which uses a combination of Ethereum blockchain technology and ciphertext-policy attribute-based encryption algorithm (CP-ABE). The use of Ethereum smart contract helps to achieve a decentralized scheme without any trusted center authority.

Wang, S. et. al. (2019) explained that Ethereum smart contract technology is used to store key data which is in the form of ciphertext in the Ethereum blockchain network. At the same time Ethereum smart contract takes the responsibility of supervising and tracking the attitude and performance of data that is accessed. Whenever an attribute set is allocated to the data user, the data owner attaches an access period which is stored in the Ethereum blockchain. The data user is allowed to perform decryption process and access data only when the access control policy is met within the allocated access period.

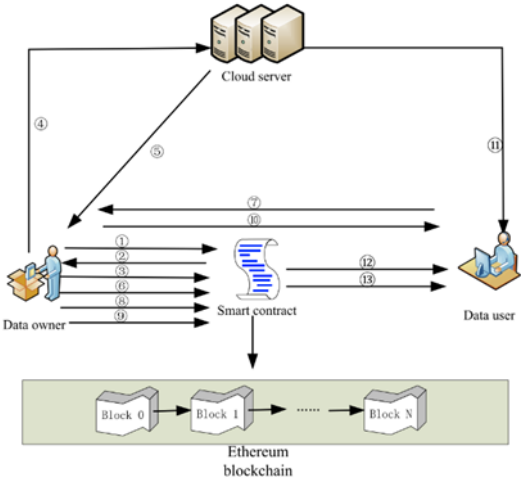


Figure 1: Representation of proposed System Model (Wang, S. et. al. (2019)

The experiment was carried out in a controlled environment utilizing two operating systems. Ubuntu Linux system is used as the environment established in a virtual machine, while the encryption algorithm is implemented using windows 10 system.

Wang, S. et. al. (2019) added that there are extra drains in carrying out the formation and implementation of smart contract operations, which is the use of Ethereum. This is to help in evaluating the cost of experimental data. Ethereum is set to 1 *ether* ≈ 200 *USD*, and 1 *gas*Price ≈ 1 *Gwei*, $1Gwei = 10^{9}wei = 10^{-9}ether$.

function	GasUsed	Actual Cost(ether)	USD
Contract create	1272934	0.002545868	0.5091736
setSecretKey	911964	0.001823928	0.3647856
setHashFileId	69501	0.000139002	0.0278004
setCipherText	110852	0.000221704	0.0443408
getSecretKey	41088	0.000082176	0.0164352
checkHashFileId	24133	0.000048266	0.0096532
getCipherText	25798	0.000051596	0.0103192
setInterval	87589	0.000175178	0.0350356
getInterval	28871	0.000057742	0.0115484

Table 1: List of gas cost and cost of some operations on smart contracts (Wang, S. et. al. (2019)

In Table 1 above Wang, S. et. al. (2019) explains that based on prototypes deployed on the blockchain; each file uploaded by the data owner is less than $1 in order to implement a smart contract. In the same way, a data user spends $0.048 every time data is accessed.

Figure 2 below represents the broken line of the orange square, which shows the change in execution time of the algorithm as the attribute grows.

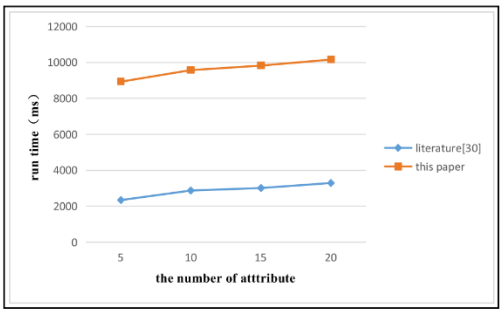


Figure 2: Representation of run time of algorithm under different number of Attributes (Wang, S. et. al. (2019)

Wang, S. et. al. (2019) experiment result demonstrates higher efficiency based on blockchain technology and compared their research to Zhang, P. et. al. (2016) whose run time algorithm increases as the attribute increases although at a lower efficiency as seen in Figure 2.

In evaluating the research by Wang, S. et. al. (2019) based on Ethereum blockchain technology, the research demonstrates an efficient approach in securing data with a higher efficiency output in testing. Wang, S. et. al. (2019) also proved their choice of removing the

center authority which was to avoid attack and replaced it with a decentralized scheme which is implemented through interaction between the data owner and the data user. Furthermore, Wang, S. et. al. (2019) stated that future research is required, as cloud storage platforms are semi-honest so leading to a lack of research integrity to ensure that data uploaded was not tampered with. Hence in the future, cloud storage platforms could be substituted with devolved storage platforms.

2.2 Genetic Algorithm

Mall, S. and Saroj, S. K (2018) proposed a security framework which will deliver data protection and privacy based on genetic algorithm with a combination of a capability list. The researchers claim that the proposed security framework will provide more security and confidentiality to data in the cloud.

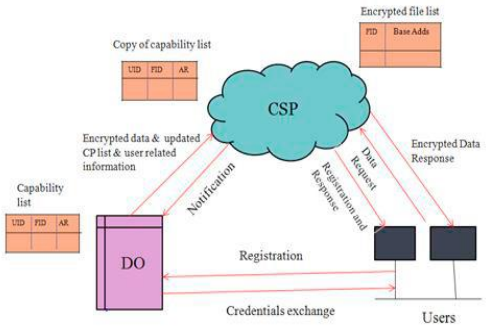


Figure 3: Representation of proposed communication method (Mall, S. and Saroj, S. K, 2018)

Mall, S. and Saroj, S. K (2018) carried out a simulation experiment to prove their claim using visual studio simulation platform. The researchers apply genetic operations such as crossover, mutation, and pseudorandom number processes to the data to be encrypted. The pseudorandom number is generated to decide which genetic algorithm process will be applied to the data which was converted to binary bits and then further divided into blocks of 8 bits. The result is a ciphertext made up of two blocks of bits stored in the different locations in the cloud.

Figure 4 below represents the security framework of the proposed scheme which converts data first into ASCII values and then into ASCII bits, these bits are further split into blocks before generating a pseudorandom number which enables the data to be encrypted. Figure 5 is a representation of the working splitting, encryption, and decryption process on the system.

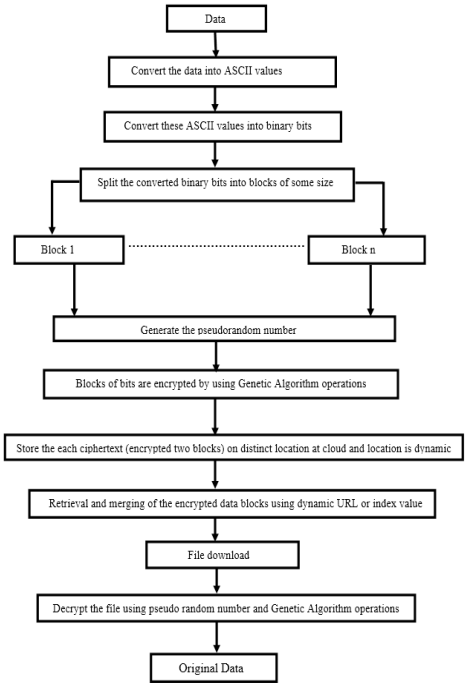


Figure 4: Security Framework of Proposed Scheme (Mall, S. and Saroj, S. K, 2018)

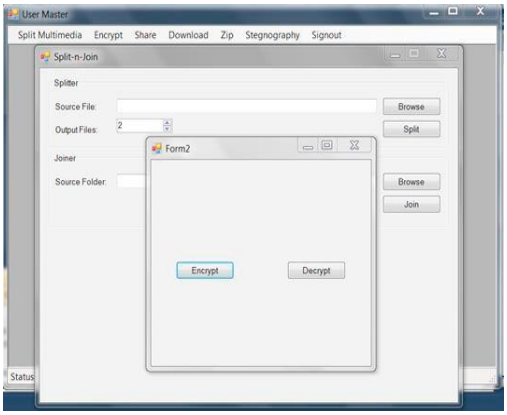


Figure 5: File splitting, encryption, and decryption process (Mall, S. and Saroj, S. K, 2018)

Tahir, M. et. al. (2020) proposed an enhanced security framework utilizing cryptosystem based

genetic algorithm to improve cloud data security which will provide data privacy and integrity.

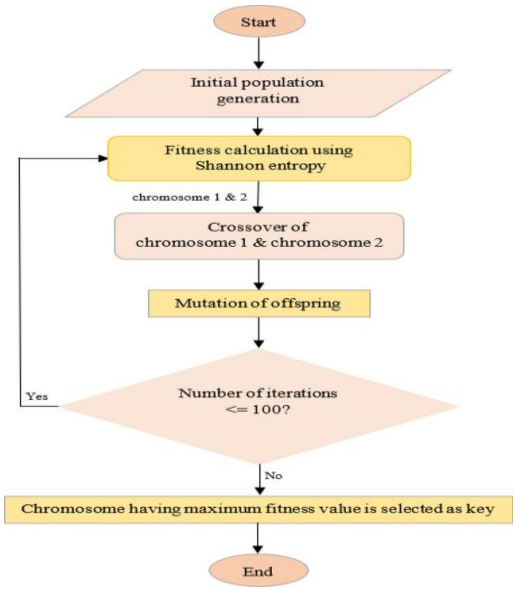


Figure 6: Flow chart of Cryptosystem using Genetic Algorithm (Tahir, M. et. al. 2020)

Tahir, M. et. al. (2020) framework utilizes Shannon entropy which is used to calculate the data to determine the capacity needed to unfailingly transmit the data in encoded binary numbers. The Genetic Algorithm generates three random numbers used for encryption and decryption of data and are combined with a cryptographic algorithm to secure data which is transmitted and stored in the cloud.

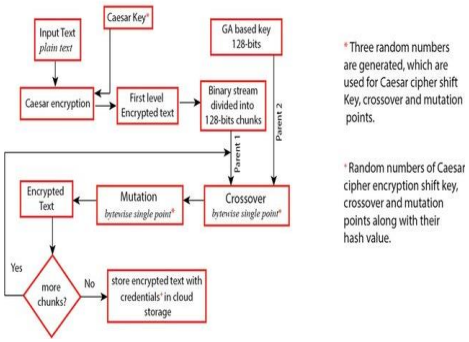


Figure 7: Encryption process of Cryptosystem using Genetic Algorithm (Tahir, M. et. al. 2020)

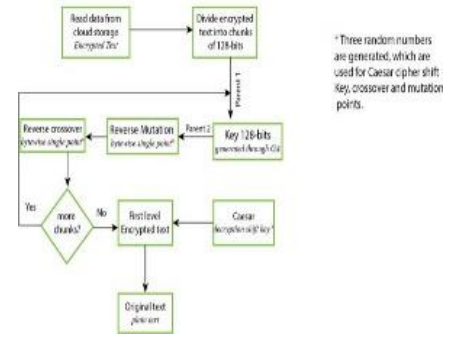


Figure 8: Decryption process of cryptosystem using Genetic Algorithm (Tahir, M. et. al. 2020)

In figure 7 the initial phase encrypts plain text using Caesar key is applied. Three random numbers are generated, and these are used for the Caesar cipher shift key, crossover, and mutation points. Figure 8 is a reverse of the encryption process. The encrypted data is read from cloud storage and divided into chunks of 128 bits, Caesar decryption shift key is applied to encrypted text to get the original text.

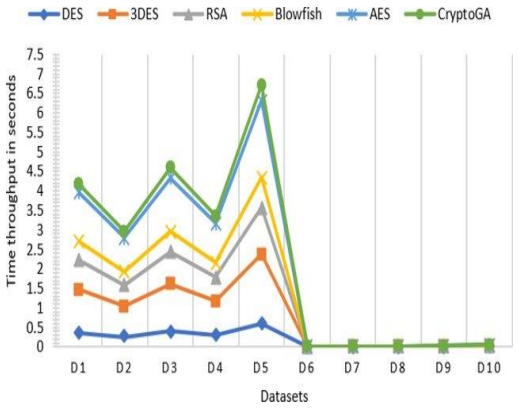


Figure 9: Encryption throughput efficiency behavior comparison from largest to smallest size datasets (Tahir, M. et. al. 2020)

Based on Figure 9, ten dataset results are used for experimenting the test and for validation, the experiment results show that the framework model guarantees data integrity, protects privacy of user's data from being accessed illegitimately. The researchers comparing their result to advanced cryptographic algorithms such as DES, 3DES, RSA, Blowfish, and AES using selected parameters concluded that Crypto-Genetic Algorithm offers improved performance.

Evaluating the research carried out by Wang, S. et. al. (2019**)** the result validated the claim made

and presents no bias. The framework does not make use of keys during encryption reducing the extra time added to computation. Also, the experiment was clear allowing for repeated experimentation of the process by other researchers. However, research done by Tahir, M. et. al. (2020) represented an improved operation using genetic algorithm and Caesar encryption.

2.3 RSA and AES Encryption Algorithms

Khanezaei, N. and Hanapi, Z. M. (2014) conducted research into developing more complex algorithm which would reduce transmission time of data between the data user and cloud. The security scheme utilizes the combination of AES and RSA encryption algorithms for each file uploaded to the cloud.

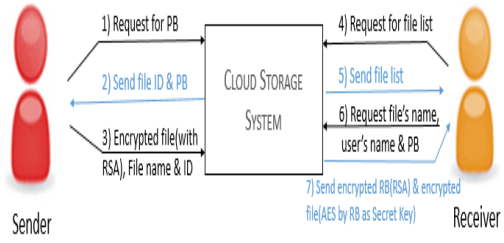


Figure 10: Proposed Cloud Storage Framework (Khanezaei, N. and Hanapi, Z. M. 2014)

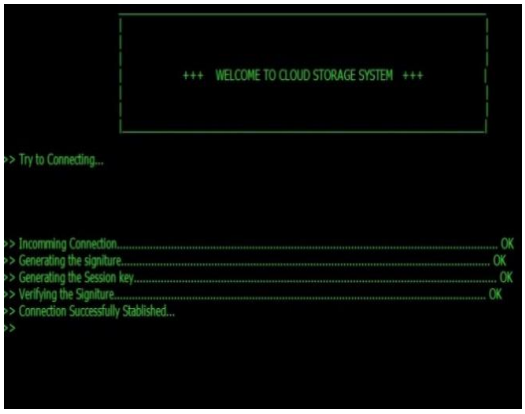


Figure 11: Captured Interface of Connectivity (Khanezaei, N. and Hanapi, Z. M. 2014)

For the experiment, the researchers used two different simulated .net platform to represent the user and server. Parameter for the size of file used for the experiment is 256 bytes. Figure 10 shows the communication between the sender, receiver, and cloud storage. Figure 11 is a display of the interface which checks to verify the system connections.

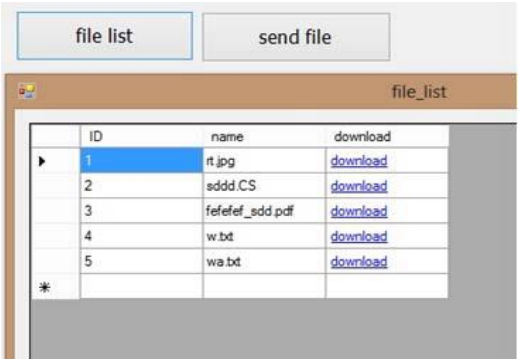


Figure 12: Interface of User (Khanezaei, N. and Hanapi, Z. M. 2014)

Figure 12 displays the interface of the system which contains list of files to be downloaded and sent.

Khanezaei, N. and Hanapi, Z. M. (2014) stresses that the security scheme uses symmetric algorithm to retrieve data from the cloud to eliminate the challenge of key distribution which results in optimal outcomes.

Amalarethinam, D. I. G. and Leena, H. M. (2017) proposed an improved framework based on asymmetric algorithm (ERSA) which uses two keys for the encryption and encryption operation. The key size is varied to strengthen the process of encryption, which results in increase in the encryption and decryption time. To increase the complexity of the algorithm, two additional prime numbers are added to the enhanced RSA algorithm. Amalarethinam, D. I. G. and Leena, H. M. (2017) claims the enhanced algorithm boosts the speed during encryption and decryption and this is achieved by separating and encrypting each file by blocks.

In experimenting the scheme Amalarethinam, D. I. G. and Leena, H. M. (2017) carried out the test using Java environment.

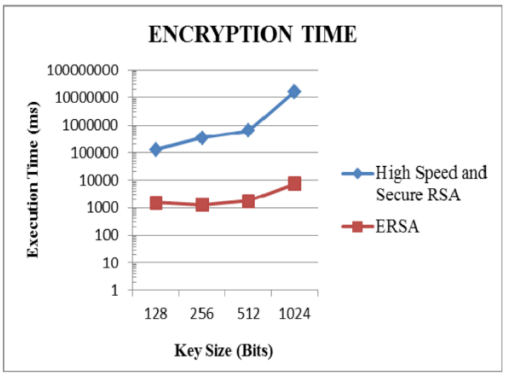


Figure 13: Comparison of encryption Time with different Key Sizes

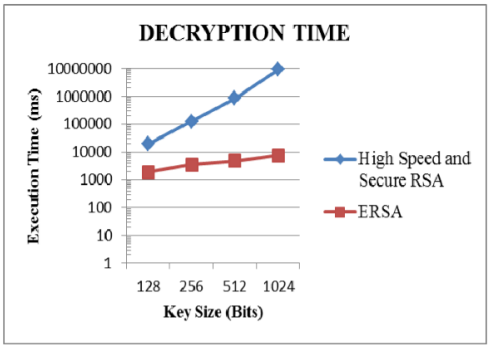


Figure 14: Comparison of decryption Time with different Key Sizes

FILE SIZE (KB)	KEY SIZE (bits)	BLOCK SIZE (bits)	Encryption Time(ms)		Decryption Time(ms)	
			High Speed and Secure RSA	ERSA	High Speed and Secure RSA	ERSA
32	128	255	125206	1489	19641	1873
	256	511	334979	1283	122211	3454
	512	1023	623236	1730	846567	4773
	1024	2047	15652812	7581	9262206	7519

Figure 15: Comparison of encryption and decryption time of two algorithms with different key sizes.

Based on Figures 14, 15, and 16, the researchers compared their scheme with several keys during encryption and decryption and concluded that the algorithms used shows an improved timing and more stable performances compared to the high speed and secure RSA.

3 Comparison of Cloud Security Frameworks

The research evaluated throughout this paper all aimed at improving data security in the cloud. The various frameworks evaluated showed more merits in the strategies used and few demerits, however the results from the different algorithms permitted comparisons and recommendations.

Research by Wang, S. et. al. (2019) showed that using smart contract along with cipher-text, enabled it to be decentralized and increased the efficiency of the algorithm outcome compared to Zhang, P. et. al. (2016). However, from the experiment the further the data users access shared files the lesser cost for the data owner. Although there could also be accumulated losses for the data owner if large file uploads are done with less data users accessing the files. Similarly, there could be an accumulation of cost for the data user if files to be accessed cannot be downloaded during the access period validity, meaning each time a data user returns to access a specific file, continued payment will precede its access.

Mall, S. and Saroj, S. K (2018) and Tahir, M. et. al. (2020) experiment result using simulation showed an efficient security framework, although Mall, S. and Saroj, S. K (2018) experiment applies either the crossover chromosome or mutation operations on the split binary blocks after the pseudorandom number is generated. While Tahir, M. et. al. (2020) algorithm applies both crossover chromosomes and mutation offspring for higher level encryption resulting in a more efficient outcome than Mall, S. and Saroj, S. K (2018) as seen in Figure 9.

The research experiment by Khanezaei, N. and Hanapi, Z. M. (2014) and Amalarethinam, D. I. G. and Leena, H. M. (2017) based on AES and RSA resulted in successful outcomes. However, the experiment by Khanezaei, N. and Hanapi, Z. M. (2014) had some limitations, the encryption and decryption operations are carried out twice for each file which means keys are generated twice and this could lead to system overhead and storage overload. This could inevitably slow the speed of the system and increase waiting time. In comparison to Amalarethinam, D. I. G. and Leena, H. M. (2017) algorithms, it showed an enhanced method data encryption and decryption using several key sizes as seen in Figures 14, 15 and 16.

The findings show that the performances of the various frameworks differ significantly so, experiment by Wang, S. et. al. (2019) on Blockchain Technology has the best performance as seen in Figures 1, 2, and 3.

4 Conclusions

As cloud adoption increases, more intelligent threats and attacks will be launched to its storage systems. Hence the need for continuous research into more efficient ways of providing a and maintaining cloud data privacy and integrity. This paper has critically analyzed the various security frameworks utilizing different frameworks to improve security of data in the cloud and the experiments are well presented in terms of security and performance.

The research carried out by Wang, S. et. al. (2019) and Zhang, P. et. al. (2016) shows good prospects as more research is done in this area, however, more research on decentralized storage using smart contract is recommended.
Study by Mall, S. and Saroj, S. K (2018) and Tahir, M. et. al. (2020) represents efficiency as the use of crossover and mutation eliminates the idea of utilizing keys which usually leads to reduction in system performance and poor data confidentiality.

Based on the research by Khanezaei, N. and Hanapi, Z. M. (2014) comparing results to other algorithms is recommended and experiments showed improved timing and stable performance.

References

Amalarethinam, D. I. G. and Leena, H. M. 2017. 'Enhanced RSA Algorithm with varying Key Sizes for Data Security in Cloud.' *2017 World Congress on Computing and Communication Technologies (WCCCT)*. p. 172-175.

Arki, O., Zitouni, A. 2018. 'A Security Framework for Cloud Data Storage (CDS) based on Agent.' *Advances in Intelligent Systems and Computing*. 662 (1), p1-12.

Chawkia, E. B., Ahmeda, A., Zakariae, T. 2018. 'IaaS Cloud Model Security Issues on Behalf Cloud Provider and User Security Behaviors.' *Procedia Computer Science Journal*. 134 (2), p328-333.

Deng, H., Wu, Q., Qin, B., Domingo-Ferrer, J., Zhang, L., Liu, J., Shi, W. 2014. 'Ciphertext-policy hierarchical attribute-based encryption with short ciphertexts.' *Elsevier Information Sciences*. 275 (5), P370-384

Garg, P., Sharma, V. 2014. 'An Efficient and Secure Data Storage in Mobile Cloud Computing through RSA and Hash Function.' *IEEE International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*. p334-339.

Khanezaei, N. and Hanapi, Z. M. 2014. 'A Framework Based on RSA and AES Encryption Algorithms for Cloud Computing Services.' *IEEE Conference on Systems, Process and Control (ICSPC).* p 58-62.

Li, J., Yu, Q. and Zhang, Y. 2019. 'Hierarchical attribute-based encryption with continuous leakage-resilience.' *Elsevier Information Sciences.* 484 (5), p113-134.

Mall, S. and Saroj, S. K. 2018. 'A New Security Framework for Cloud Data.' *Procedia Computer Science - 8th International Conference on Advances in Computing and Communication (ICACC)*. p765-775.

Sanhotran, R. 2020. 'Report: The State of Cloud Security' 2020. Available: https://news.sophos.com/en-us/2020/08/07/the-state-of-cloud-security-2020/. Last accessed Dec. 2020.

Tahir, M., Sardaraz, M., Mehmood, Z. and Muhammad, S. 2020. 'CryptoGA: A Cryptosystem based on Genetic Algorithm for Cloud Data Security.' *Cluster Computing Journal*. 11 (1), p516–521.

Wang, S., Wang, X. and Zhang Y. 2019. 'A Secure Cloud Storage Framework with Access Control Based on Blockchain.' *IEEE Access*. 7 (7), p112713-112725.

Zhang, P., Chen, Z., Liang, K., Wang, S. and Wang, T. 2016. 'A Cloud-Based Access Control Scheme with User Revocation and Attribute

Update.' *Australasian Conference on Information Security and Privacy*. p2713-2725.

Evaluation and Analysis of the Click-through Rate Prediction Method of the Current Advertising Push System

Xiaoyao Li

Abstract

The benefit of advertising is directly related to the click-through rate of advertising. By predicting the click-through rate of advertising, optimization to the mode of delivery can ensure maximum value from any given advert. This paper introduced the different single-feature modelling and multi-feature modelling methods for click-through rate prediction. These methods were compared and evaluated from the aspects of feature extraction, data acquisition, algorithm model and prediction performance. According to the real experimental data and practical application requirements, an idea of combining different feature methods was put forward to predict the click-through rate of ads more accurately in the future research. Ultimately, it will result in the achievement of a greater user experience and higher business value from advertisements.

1 Introduction

The accuracy of the prediction of the click-through rate of advertisements is an important business issue, which directly affects the profits of an enterprise and the experience of users. Dan et. al. (2021) mentioned that mobile video advertising generated $19.93 billion in 2019, and Google became the world's first online digital advertising provider with revenue of more than $100 billion in 2019. Han and Jifan (2020) mentioned that no one is willing to click on the ad that they are not interested in. Hailong et. al. (2020) pointed out that the click-through rate has a direct impact on the revenue of advertisers. Qianqian et. al. (2020) mentioned that both a purchase and a search process require user preferences to be judged by the click-through rate. Dongfang et. al. (2021) proposed that the click-through rate can greatly increase the revenue and traffic of advertisers. Therefore, the importance of advertising click-through rate prediction is self-evident. Optimizing advertising through the prediction of click-through rate is a necessary link in the development of advertisers. In academic circles, how to make advertising play a more effective role has increasingly become a major research focus in the field of computer science (Edward and Hairong, 2017).

In order to alleviate this problem, Ying X et. al. (2019) used a robust integrated local kernel embedded (RILKE) model, which can solve the problem of data sparsity if used. Unsupervised transfer learning is added to obtain an improved model, called robust transfer integrated local kernel embedded (RTILKE), which can effectively overcome the problem of data imbalance and improve the prediction of click-through rate.

In addition, Dongfang et. al. (2021) found that most of the mainstream methods only excavate data based on the characteristics of users' historical behaviour. They said that it was not enough to capture the diversity of users' interests, so they proposed a new algorithm to improve the capture accuracy, thus improving the click-through and conversion rate of ads.

What is more innovative, Dan et. al. (2021) proposed that the multi-view feature transfer method used the transfer relationship among the features of each data set to find more feature details to improve the click-through rate.

Through critical evaluation, this study will analyse the predictive methods and experiments of click-through rate in the current advertising system, in order to find a more effective prediction method of click-through rate from the point of view of prediction accuracy. In addition, the

comparison and combination of these methods will be discussed.

2 Current techniques for predicting the click-through rate of advertisements using different methods

In this part, we will analyse several current papers, using different methods divided into single-feature and multi-feature models of click-through rate prediction, and evaluate their experiments and results.

2.1 Single feature extraction modelling

Qianqian et. al. (2020) found that there were few methods to predict click-through rate based on user interest, so they proposed a model method based on attention-depth interest.

In the experiment, Qianqian et. al. (2020) used book data-sets, electronic data-sets and Frappe data-sets from two Amazon data-sets as well as two common data-sets Frappe and MovieLens. The interest sequence was captured by the interest extractor, then the dependence relationship between behaviours was modelled by a two-way long-term and short-term memory network and applied in real data. Finally, As shown in figure 1, using AUC as indicators, the performance of this model was better than the other three models by 1.8%. By comparing the results of log loss and RMSE, the model also has better accuracy. Qianqian et. al. (2020) concluded from the experimental results that it is very effective to predict click-through rates from the perspective of users' interests.

Qianqian et. al. (2020) used the user interest characteristics as the research object, then carried on the precise experimental design by using the common index as the judgment index, which is convenient to compare the performance with other models. Although they only used one feature as the research object, they studied from the perspective of the user of the click on the advertisement, which is more conducive to the accurate prediction of click-through rate. However, in order to more accurately understand the user's interests, more work needs to be done in the acquisition and analysis of user information, such as the filtering of invalid information.

In order to overcome the limitations of the advertising click-through rate prediction model, Ashish and Jari (2018) proposed a method to predict click-through rate, via the positioning of advertising links, from the point of view of psychology and visual attention.

In the experiment, Ashish and Jari (2018) used data from Liana Technologies email service providers. They divided the screen into four different regions, then collected and analysed the data

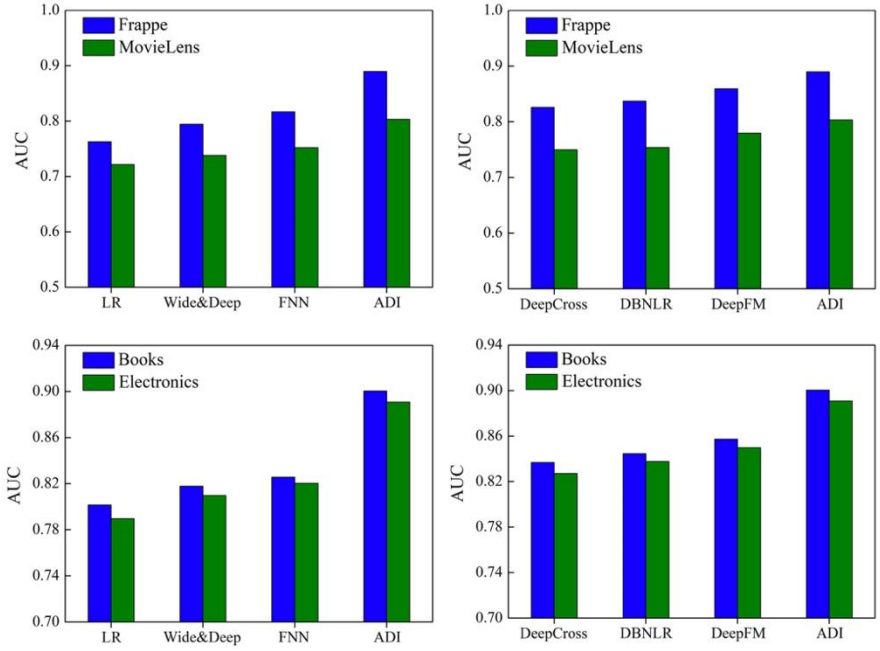


Figure 13 AUC performance comparison in different datasets (Qianqian et. al. 2020)

of users' responses to advertising links in different locations. This experiment lasted for 6 months. 80% of the customers from 12 companies in 4 different countries were randomly selected. A total of 110 e-mail newsletters were received from 10 companies. It was found that the click-through rate in the North-east region of the screen was the highest with 67% form table 1. To confirm causality, they wanted to conduct a more detailed experiment. However it was terminated without the approval of the contractor.

Table 1 summary statistics of data (Ashish and Jari 2018)

Variable	Definition	Percentage (%)
NEClickR	Email newsletters with clicks in northeast region.	67.27
NWClickR	Email newsletters with clicks in northwest region.	58.18
SWClickR	Email newsletters with clicks in southwest region.	61.82
SEClickR	Email newsletters with clicks in southeast region.	59.09

Although the idea of the new method proposed by Ashish and Jari (2018) is novel and some conclusions have been obtained through preliminary experiments, the fact that there is not any further experimentation means that we cannot assume the preliminary conclusions are valid. That being said, it provides us with a good idea for further experimentation on whether the position of adverts can influence click-through rate vs. impressions for an advert.

Carsten (2018) realized that ranking was also a key factor influencing click rates from the research of the impact of advertising positioning on the click-through rate of ads in search engine advertisements, so he tried to analyse the feature of ranking by using single feature modelling.

In this experiment Carsten (2018) recorded the search engine advertising campaign of a service company for three years. Approximately 20 million advertising impressions, 400000 clicks, 13000 conversions and 800 contracts were generated. Table 2 shows the daily average age data and rankings. For the service company, the No. 3 ad had the highest average click-through rate. Carsten (2018) claimed that the highest click-through rate was found with the third/fourth highest ranking.

Although Carsten (2018) proved that there is not a positive-relationship between the ranking of advertisements and click-through rate, he did not specify how ranking affects the click-through rate in search engines or generate more specific ways to improve click-through rate based on this feature. However, it can help us to build a clearer and more effective model in future work on the topic.

In order to explore the relationship between advertising itself and click through rate. Lohtia et. al. (2003) presented a new idea about the influence of content and designed elements on the click-through rate of banner ads.

In the experiment, Lohtia et. al. (2003) used the data of 8725 real banner ads, classified and recorded the samples according to the three criteria of interaction, colour and animation, after which they carried out a controlled experiment to collect the click-through rate record. The results showed that the content and design of adverts had

Table 2 Descriptive statistics of daily search engine advertisements (Carsten 2018)

Rank	Days	Impr.	Clicks	CTR	Costs	Conv.	CR	Contracts	Signing rate	Value after advertising
1	–	–	–	–	–	–	–	–	–	–
2	34	9917	215	.0217	546.57	7	.0313	.47	.0698	394.61
3	600	19,330	406	.0210	926.74	11	.0274	.83	.0745	729.92
4	303	19,893	385	.0194	868.10	15	.0399	.73	.0472	584.05
5	126	17,527	281	.0161	620.96	14	.0482	.63	.0462	633.00
6	28	7084	120	.0170	243.65	7	.0607	.43	.0586	613.49
7	2	1399	11	.0075	3.61	–	–	–	–	−3.61
8	1	12,929	210	.0163	439.45	5	.0224	–	–	−439.45
9	–	–	–	–	–	–	–	–	–	–
10	2	12,171	205	.0169	438.24	5	.0230	.50	.1061	561.76

a significant impact on the click-through rate. They placed emphasis on four advertising features: cognitive content; emotional content; cognitive design and emotional design.

The advertising features Lohtia et. al. (2003) proposed have significance for the design of advertising content in the future, they found that these factors only affected click-through rate and failed to generate a standardized model for judging the advantages and disadvantages of an advertisement according to the four characteristics of the advertisement. Therefore, if their model is to be used in the actual advertising design, additional experiments and analysis are needed to verify its validity.

2.2 Multi-feature modelling

To solve the problem of sparse feature learning, Qianqian et. al. (2019) developed a method of learning sparse features based on deep learning, and studied the direct relationship between data reduction and dimension features.

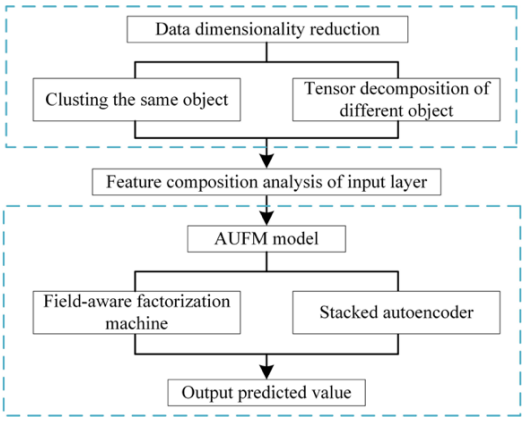


Figure 2 DLSFL method structure (Qianqian et. al. 2019)

In this part of the experiment, Qianqian et. al. (2019) had a total of 149639105 records (9.5GB) for training data and 20275594 records (1.28GB) for test data. These data were extracted from the advertisement click log data SIGKDD Cup2012 track2 of Tencent's website and filtered, then 3.5 million samples were randomly selected for the experiment. As illustrated in Figure 2, DLSFL was used to train the model in seven different scale data sets, and the prediction performance of different methods was verified on the same test set. Finally, AUC and log-loss were used as evaluation criteria. The experimental results (table 3 and 4) shows that the DLSFL method performs is better than the AUC and log-loss of other methods, so the prediction is more effective.

The method proposed by Qianqian et. al. (2019) discovered more subtle features related to click-through rate through data dimensionality reduction, deep learning and could be verified by a large number of experiments based on data. This method could greatly improve the feature richness under the determined existing data, thus improving the prediction of click-through rate.

Table 3 AUC evaluation results (Qianqian et.al. 2019)

(a)

Data Size	AUC				
	LR	HPCM	Human_ LR	FM	DLSFL
150,000	0.6851	0.6934	0.6931	0.7113	0.7205
200,000	0.6925	0.7003	0.7024	0.7235	0.7328
300,000	0.7034	0.7145	0.7115	0.7386	0.7476
500,000	0.7121	0.7226	0.7197	0.7422	0.7539
600,000	0.7183	0.7393	0.7285	0.7498	0.7663
750,000	0.7267	0.7485	0.7372	0.7537	0.7781
1000,000	0.7302	0.7502	0.7428	0.7649	0.7922

(b)

Data Size	AUC				
	FNN	Wide& Deep	DeepCross	DBNLR	DLSFL
150,000	0.7141	0.7196	0.7155	0.7164	0.7205
200,000	0.7266	0.7288	0.7255	0.7285	0.7328
300,000	0.7382	0.7402	0.7392	0.7403	0.7476
500,000	0.7435	0.7475	0.7447	0.7470	0.7539
600,000	0.7498	0.7563	0.7548	0.7563	0.7663
750,000	0.7560	0.7625	0.7582	0.7607	0.7781
1000,000	0.7617	0.7759	0.7703	0.7769	0.7922

Table 4 Log-loss evaluation results (Qianqian et.al. 2019)

(a)

Data Size	Log-Loss				
	LR	HPCM	Human_ LR	FM	DLSFL
150,000	0.02955	0.02926	0.02932	0.02813	0.02773
200,000	0.02893	0.02884	0.02887	0.02782	0.02764
300,000	0.02886	0.02880	0.02882	0.02779	0.02659
500,000	0.02872	0.02869	0.02871	0.02768	0.02647
600,000	0.02764	0.02710	0.02762	0.02660	0.02633
750,000	0.02751	0.02748	0.02749	0.02647	0.02576
1000,000	0.02648	0.02643	0.02646	0.02637	0.02508

(b)

Data Size	Log-Loss				
	FNN	Wide& Deep	DeepCross	DBNLR	DLSFL
150,000	0.02792	0.02701	0.02784	0.02732	0.02773
200,000	0.02764	0.02678	0.02757	0.02708	0.02764
300,000	0.02734	0.02663	0.02689	0.02671	0.02659
500,000	0.02703	0.02656	0.02677	0.02664	0.02647
600,000	0.02655	0.02649	0.02693	0.02668	0.02633
750,000	0.02640	0.02638	0.02676	0.02643	0.02576
1000,000	0.02617	0.02603	0.02638	0.02615	0.02508

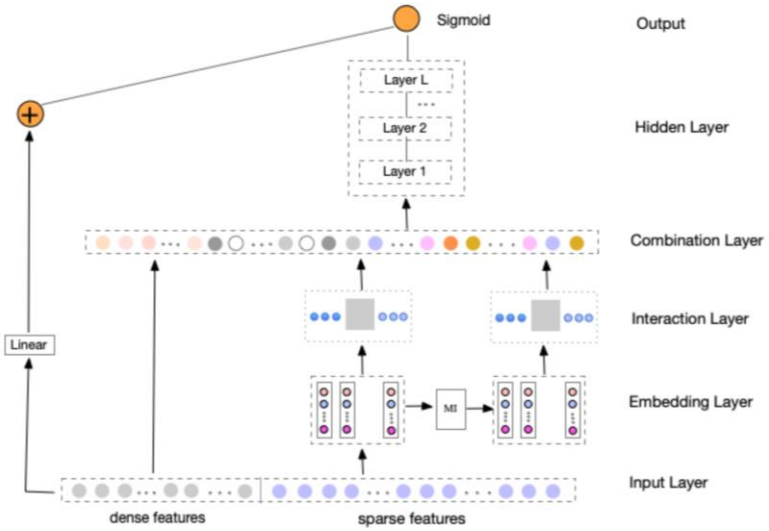


Figure 3 Architecture of MiFiNN (Xiaowei et. al. 2020)

Xiaowei et. al. (2020) took advantage of the sparsity and high-dimensional characteristics of data features and proposed a new CTR model (MiFiNN) based on interactive information and features. Figure 3 shows the structure of this method.

In the experiment, Xiaowei et. al. (2020) used Criteo, Avazu, Movielens and ICME (four sets of open CTR benchmark datasets) to make a comprehensive comparison of MiFiNN, for both a shallow model and deep learning model respectively: 1: overall performance; 2: analysis of different operations; 3: effectiveness of MiFiNN deformation; 4: hyperparametric study. They used AUC, Log-loss, RMSE as the evaluation index to analyse the results (table 5), which shows that the performance of the three evaluation indicators is better than other models.

Xiaowei et. al. (2020) established a brand-new MiFiNN model through a large amount of reliable data analysis and controlled comparative experiments. This model could not only use interactive features to improve data richness, but also interacted with the inner and outer for a second time, especially through a large number of rigorous comparative experiments to get an excellent performance in predicting the click-through rate of advertisements. At the same time, it also shows the scientific effectiveness of the controlled experimental environment.

Table 5 Performance comparison (Xiaowei et. al. 2020)

Model	Criteo			Avazu			Movielens			ICME		
	AUC	Log loss	RMSE	AUC	Log loss	RMSE	AUC	Log loss	RMSE	AUC	Log loss	RMSE
FM	0.7498	0.492	0.4015	0.7543	0.3969	0.3522	0.8282	0.4246	0.3657	0.8052	0.0567	0.1048
MLR	0.7532	0.4889	0.4	0.7598	0.3937	0.3517	0.8424	0.3925	0.3487	0.7789	0.0557	0.1037
WDL	0.7812	0.4695	0.3898	0.7618	0.3914	0.3512	0.8685	0.3485	0.3274	0.9098	0.0379	0.0912
NFM	0.7803	0.4675	0.3896	0.7693	0.3873	0.3492	0.8695	0.3487	0.328	0.8911	0.0397	0.0921
DCN	0.7832	0.4652	0.3886	0.7603	0.3923	0.3513	0.8682	0.3497	0.3284	0.9148	0.0372	0.0916
DeepFM	0.7854	0.4634	0.3884	0.7642	0.3908	0.3509	0.8686	0.3484	0.3278	0.9161	0.037	0.0912
xDeepFM	0.7869	0.4626	0.388	0.7788	0.382	0.3468	0.8696	0.3471	0.3271	0.9173	0.037	0.0914
AutoInt	0.7854	0.4615	0.3868	0.7778	0.3835	0.3476	0.8676	0.3513	0.3292	0.9155	0.0375	0.0917
FiBiNET	0.789	0.4611	0.3875	0.7797	0.3831	0.3474	0.872	0.3462	0.327	0.9161	0.0368	0.091
MiFiNN	0.7905	0.458	0.385	0.7814	0.3808	0.3465	0.8772	0.3382	0.3232	0.9178	0.0366	0.0901

Han and Jifan (2020) proposed the XGBDeepFM model and its framework is shown in figure 4. It is a method that combined contextual feature interaction to capture more details, and then to improve the prediction of advertising click-through rate.

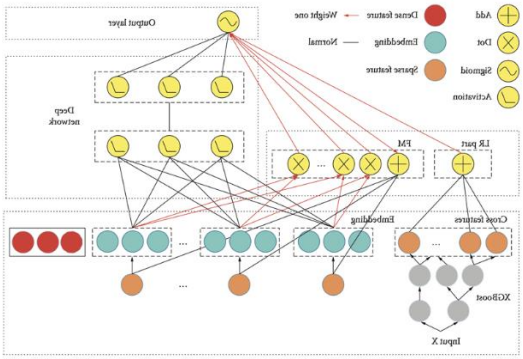


Figure 4 XGBDeepFM model (Han and Jifan 2020)

In the experiment, the dataset used by Han and Jifan (2020) contained the O2O mobile advertising data of the mobile Internet platform, which covered many offline scenarios. 4369918 accurate historical weather data from 122 cities was also obtained from the weather platform WunderGround. As seen in Figure 5, they conducted the same predictive experiment on the same data using different models (Widd, FNN, PNN, XDeepFM and XDeepFM). The experimental results showed that the prediction performance of XGBDeepFM was better than that of other models.

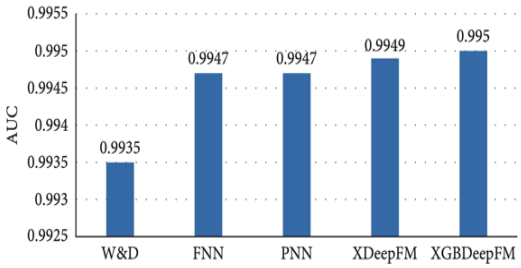


Figure 5 Prediction performance comparison (Han and Jifan 2020)

It is always difficult to obtain effective information by relying solely on the characteristics of the data itself, so the new model proposed by Han and Jifan (2020) is very effective. Innovatively connecting the context on the traditional interaction model, while strictly following the scientific experiment process and using a large amount of authentic randomized data for a large number of controlled experiments resulted in the effectiveness of the experiment being greatly improved. This method also unexpectedly improved the prediction performance of the click-through rate of the advertisement.

Similar to feature interaction, Dan et. al. (2021) came up with a method of transferring (MFT) based on multi-view features creatively. This could find subtle connections among seemingly unrelated features, thus improving the prediction of click-through rate. Figure 6 shows this framework.

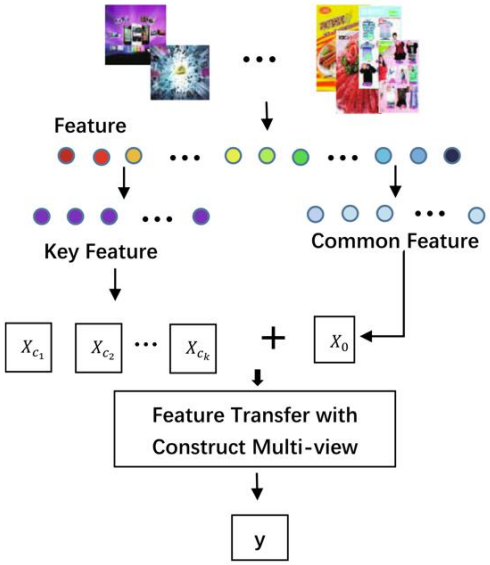


Figure 6 MFT framework (Dan et. al. 2021)

In the experiment, Dan et. al. (2021) adopted five common data sets provided by the Kaggle platform and five data sets (Avazu, Avito, Kad, Talking) with different data and characteristics. They conducted seven classic experiments: SVM, DNN, LR, FM, DSL, DAN and GAN. They also compared ACC, AUC and purity as standards with the classical machine learning algorithms SVM, DNN, LR and FM, as well as the latest prediction methods DSL, DAN and GAN. They concluded that the ACC value of the MFT method was closer to 0.9 and the AUC was more stable. Besides, the MFT method had stronger anti-imbalance performance, anti-stability and predictability because of the better purity clustering effect.

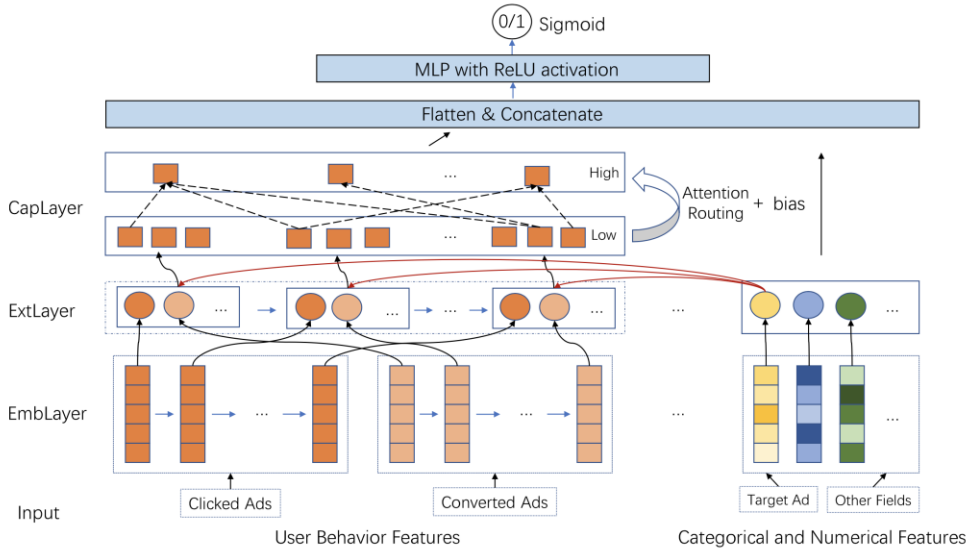


Figure 7 ACN overall architecture (Dongfang et. al. 2020)

Instead of adopting the traditional interactive characteristics, Dan et. al. (2021) combined the multi-view method and the feature transfer method into a newer, more efficient method and strictly followed the scientific experimental process without exception, taking data sources; experimental processes; controlling models and evaluation criteria seriously and finally developing a new advertising click-through rate prediction model. This model can link seemingly unrelated features together for the analysis of data, thus greatly improving the performance in the prediction of the advertising click-through rate model.

Dongfang et. al. (2020) raised an attention capsule network (ACN), which designed a dynamic algorithm based on user interest to improve the prediction of click through rate.-Figure 7 shows the overall framework of this approach.

In this experiment, the data sets used by Dongfang et. al. (2020) were from the public collection of Alimama Inc display and click logs, the product review and metadata collection of Amazon, and the click ad log collection of OCPA advertising platform. The statistics of all data sets are shown in Table 6. Finally, some classical models and new models were compared with the same data according to the two indexes of AUC and Log-loss, and the results were analysed in figure 8. The results showed that: compared with other models, ACN-T performs better in a reasonable time without affecting the user experience.

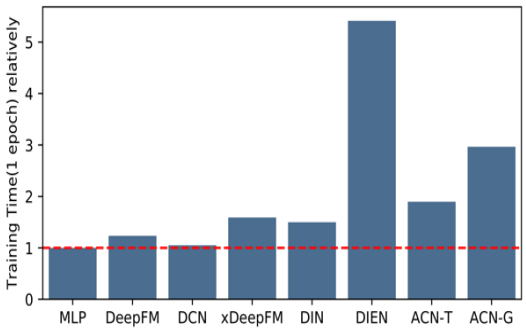


Figure 8 Comparison of the efficiency of each model (Dongfang et. al. 2020)

Table 6 Statistics of datasets (Dongfang et. al. 2020)

Data	#Samples	#Users	#Items
Advertising	26,557,962	1,140,000	846,811
Amazon	603,668	603,668	367,982
TC-CVR	290,853,252	18,232,343	2,283

The research of Dongfang et. al. (2020) combined a variety of sources to obtain the same information, and also combined the dynamic algorithm of attention to predict the click-through rate of advertisements. They obtain users' interests from the historical behaviour records, which enriches user characteristics and is more consistent with the thoughts of users, so that more accurate results can be obtained. Therefore, an excellent experiment was designed to use a variety of experimental data from reliable sources for the experiment. Good prediction performance makes the model have great potential in practical environment.

3 Comparison of Methods

The purpose of this paper is to analyse some useful techniques to improve the prediction of click-through rate in advertising systems. However, the scope of application of different methods is not necessarily the same. By comparing these conclusions and methods, we can better understand the advantages and disadvantages of these methods, and then flexibly apply these methods according to different environments, so as to improve the prediction accuracy of advertising click through rate and improve the user experience

From the view of the research topic, the perspectives of Dan et. al. (2021), Ashish and Jari (2018), Qianqian et. al. (2020) and Lohtia et. al. (2003) were all very novel. They chose interest, advertising location and content design as their research directions innovatively, which provided new ideas for future research.

In terms of experimental integrity, most of the methods were complete and rigorous, such as Qianqian et. al. (2020), Qianqian et. al. (2019), Dan et. al. (2021) and Dongfang et. al. (2020), they all had reliable data sources and established effective models; However, Ashish and Jari's (2018) experiment was not fully supported by users, Lohtia et. al. (2003) and Carsten (2018) did not come up with specific methods to improve predictability in the study, so their model required to be improved further.

Finally, regarding the experimental results, the MFT model of Dan et. al. (2021) performed best under AUC and log-loss which were commonly used in the industry as evaluation criteria. But it was a pity that the models of Ashish and Jari (2018); Lohtia et. al. (2003); Carsten (2018) and Han and Jifan (2020) did not use AUC or log-loss, so they could not be compared with other methods. Moreover, Dan et. al. (2021) did not consider some characteristics such as advertising position and user interest, so it could not be applied to all practical environments.

4 Conclusions

This paper evaluated the current methods of single-feature and multi-feature models for predicting click-through rates. In these methods, the researchers used a variety of research features, such as attention-depth interest features, learning sparse features, and creatively selecting the location of advertising links and search engine rankings to predict the click-through rate, which provided a lot of new directions for future work.

By comparing these methods from the point of research perspectives, experimental design and results, it can be seen that the MFT model of Dan et. al. (2021) complete and innovatively combined the multi-view method, feature transfer method and showed excellent performance in predicting click-through rate. However, other factors must be considered comprehensively if the model is to truly be applied in an authentic environment, such as the locations mentioned by Shish and Jari (2018), the advertising content explored by Qianqian et. al. (2020), and the interest of users studied by Lohtia et. al. (2003). If these four methods can be integrated, the feature richness and stability of the model could be improved to a greater extent to better combat the high sparsity of data in advertising prediction. Compared with a single model, this integrated approach should be used in a wider range of applications for a network platform that can interact with users, rather than a one-way communication medium similar to TV.

References

Ashish K, Jari S, 2018, 'Effects of link placements in email newsletters on their click-through rate', *Journal of Marketing Communications*; 24(5), pages 535–548.

Carsten D, 2020, 'The impact of ad positioning in search engine advertising: a multifaceted decision problem', *Electronic Commerce Research*; 20(4), pages 945–968.

Dan J, Rongbin X, Xin X, Ying , 2021, 'Multi-view feature transfer for click-through rate prediction', *Information Sciences*; 546, pages 961–976.

Dongfang L, Baotian H, Qingcai C, Xiao W, Quanchang Q, Liubin W, Haishan L, 2021, 'Attentive capsule network for click-through rate and conversion rate prediction in online advertising', *Knowledge-Based Systems*; 211(2021), 106522.

Edward C, Hairong L, 2017, 'Opportunities for and pitfalls of using big data in advertising research', *Journal of Advertising*; 46(2), pages 227–235.

Hailong Z, Jinyao Y, AND Yuan Z, 2020, 'CTR prediction models considering the dynamics of user interest', *IEEE Access*; 8, pages. 72847–72858.

Han A, Jifan R, 2020, 'XGBDeepFM for CTR predictions in mobile advertising benefits from ad context', *Mathematical Problems in Engineering*; 2020, 1747315.

Qianqian W, Fang'ai L, Pu H, Shuning X, Xiaohui Z, 2020, 'A hierarchical attention model for ctr prediction based on user interest', *IEEE Systems Journal*; 14(3), pages 4015-4024.

Qianqian W, Fang'ai L, Shuning X, Xiaohui Z, Tianlai L , 2019, 'Research on CTR prediction based on deep learning', *IEEE Access*; 7, pages 12779-12789.

Ritu L, Naveen D, Edmund K, 2003, 'The impact of content and design elements on banner advertising click-through rates', *Journal Of Advertising Research*; 43(4), pages 410-418.

Xiaowei W, Hongbin D, Shuang，2020, 'Click-through rate prediction combining mutual information feature weighting and feature interaction', *IEEE Access*; 8, pages 207216-207225.

Ying X, Dan J, Xinmei W, Rongbin X, 2019, 'Robust transfer integrated locally kernel embedding for click-through rate prediction', *Information Sciences*; 491, pages 190–203.

An Investigation of Current Methods for Improvement of Spoofing Attack Detection in Face Recognition Systems

Joel Modongo Tshwene

Abstract

Face spoofing attacks have become almost undetectable on face recognition systems which use only traditional or handcrafted features. This paper analyses some current CNN based methods under the use of deep neural networks, support vector machine and multi-channel CNN for improving the detection of spoofing attack on face recognition system. The paper goes on to compare the best methods picked from the above-mentioned areas of analysis. A recommendation of a possible fusion of some methods which can result in more robust method is made but that would seek further research.

1 Introduction

Biometric systems like the face recognition system are vulnerable to attacks such as print attacks, replay attacks and 3D masks attacks which fool such systems. Over the years researcher have been coming up with methods to detect spoofing attacks which are also known as presentation attacks (PA) and their methods performed very well. Attackers on the other hand keep coming up with complex ways of fooling face recognition systems.

Sepas-Moghaddam et al. (2018) worked on a Face spoofing detection using the IST Lenslet Light Field imaging framework. The method was made to benefit from the contrast color spaces on different directions in a captured light field.

A combination of features like Discrete Wavelet Transform, Local Binary Pattern and Discrete Cosine Transform was made by Zhang & Xiang (2020) to evaluate if a video is valid. Their goal was to develop a method which can be used on devices that have lower computation abilities.

Sun et al. (2020) proposed a Domain Adaptation and Lossless Size Adaptation method. Researchers wanted to achieve a method which works well across different domains. Their method showed some competitive results.

Arashloo et al. (2015) proposed the use of discerning representation dependent on dynamic multiscale binarized statistical image features and a kernel discriminant analysis approach. Their aim was to increase the ability of detection without the use of costly eigen-analysis. They conclude that the combination of their method with other techniques improved the performance of the system.

This paper investigates research on state-of-the-art methods which improve the accuracy of detection of spoofing attacks using various machine learning classification. A comparison of methods with higher accuracy will be made and a possible combination of two methods to come up with a robust spoofing attack detector.

2 Current Face Spoofing Detection Methods to Improve Detection Accuracy on Face Recognition Systems

This section will analyse the current methods for face spoofing detection that use various machine learning classification from several research papers. The section will mainly focus on the use of deep neural network, support vector machine (SVM), multi-channel CNN on detecting face spoofing attacks. Methodologies, results, and conclusions on those topics will be analysed.

2.1 Deep Neural Network

Das et al. (2019) worked on a detection method which fuses the handcrafted features and deep neural network features which are based on transfer learning. The method aims at creating a deep neural network which can solve robust features for face anti-spoofing detection.

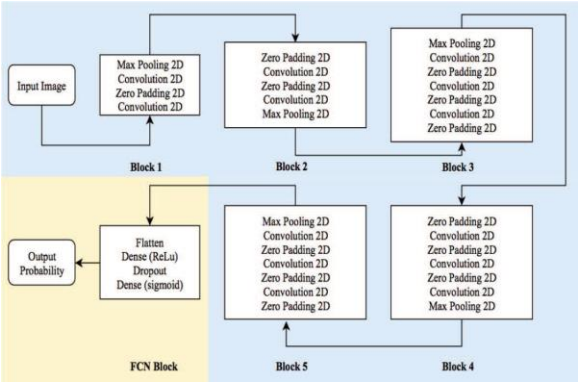


Fig. 1: Shows the upper layers of the VGG-16 represented in blue and the layers modified on (Das et al. 2019) research represented in yellow.

On their experiment Das et al. (2019) used four datasets namely, SSIJRI face spoofing dataset, replay-attack dataset, replay-mobile dataset and 3DMAD dataset. Two folders were formed for the training part and testing part, the folders were then divided into "real" or "attack". Keras with Tensorflow were used as backend to increase the performance of their approach. From the four datasets the proposed method only got three highest accuracies after training from those datasets. Das et al. (2019) stressed that both the methods they used, performs better on a different image quality so if the method is fused perfectly, it improves the accuracy of face spoofing detection as compared to being used as a single method.

Das et al. (2019) made experiments which trained and were tested on the four datasets with many different data features using deep neural networks. They continued with the process adding different data from a pre-processed database.

Researchers repeated experiments to be sure of the performance of their method. They tested their method on many different scenarios. The flow of the methodology on their paper was good and the claim made that their method works best is backed up by fully justified experiments.
Alotaibi & Mahmood (2016) were trying to detect presentation attacks by using a nonlinear diffusion which depend on the additive operator splitting. They presented a specially designed deep convolutional neural network which obtain convoluted and high features of the input diffused frame.

On their experiment Alotaibi & Mahmood (2016) used the SoftMax activation function as a classifier. Data used was from the Replay Attack Database. The researchers came to some conclusions that their proposed method detected complex features on diffused images as they had expected. Researchers made a conclusion that their method works better than most methods.

Research used a replay attack database which is used by other researchers in testing their spoofing detection methods. Their research shows good science practice with one exception; more data from other different datasets would have been ideal for testing their method to increase knowledge on the reliability of their proposed method.

2.2 Support Vector Machine (SVM)

Yao et al. (2020) used Riemannian reweighted KNN fused with an SVM that is sensitive to attacks to create a relativity representation on Riemannian manifold for detecting spoof attacks. It increases abstraction potential at the same time checking differentiability. Researchers also consider a fusion of the proposed method with some classifiers which were not used on their research to increase the accuracy and performance in the detection of face spoofing attacks.

For their experiments Yao et al. (2020) used four datasets to test their method. They compared their proposed method to other methods that are related to the proposed one, they did that on the two datasets that were used in the testing of such methods. Their method performed very well giving out excellent results. It showed results that were more accurate than the ones from all traditional methods, however three methods that are based on deep learning showed more excellent results than the proposed methods. Researchers stated clearly where their methods

would perform poorly and how that could be addresses. The adoption of other method would help to improve the accuracy on areas where their method lacks like on the detection of unseen attacks.

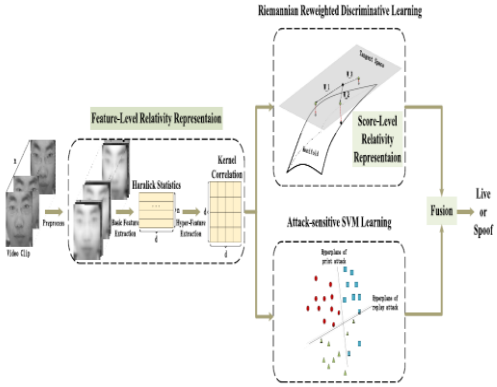


Fig. 2: Indicates how Yao et al. (2020) achieved their final classification after they combined Riemannian reweighted discriminative learning and attack sensitive SVM learning.

Yao et al. (2020) made experiments which were all fair. They presented their result in a fair manner. They did not hide negative results as a way of promoting their proposed method. This proves that presentation of results is not biased. Their research illustrated a valid science practice.

Raghavendra et al. (2018) presented a method which is targeted at detecting face spoofing attacks which are a result of high-quality printed attacks as such attacks are not easier to detect. For a powerful detection, their method is based on the multi resolution local phase quantization.

For their experiment Raghavendra et al. (2018) firstly train SVM classifier with samples containing positive and negatives. The linear SVM will distinguish between positive class (genuine presentation) and negative class (attack presentation). The accuracy of their method was compared with six current methods in use which are more accurate, and the proposed method gave out the best performance.

During the performance testing process there were no bias indicated when they were comparing their method with other methods as the results were presented following the ISO/IEC metrics. The claim that their method performs best was fully justified.

Fatemifar et al. (2019) proposed an anomaly detection approach to detect presentation attacks. They state that their method can fully detect unseen attack which were not detected by previous methods. Their one-class classifier is developed with depictions from a deep pre-trained CNN model together with the SVM and one class SRC.

On the experiments they included two more classification approaches namely the Mahalanobis Distance and the Gaussian Mixture Model. Three datasets were used on the experiments. On each query sample the classifiers were used to compute some scores which differentiated spoofing and genuine faces. Mismatched fusions of CNNs, classifiers and datasets were used to conduct ample experiments. They used HTER to measure the performance and ResNet50 showed to be the best approach.

The Replay-Attack Dataset (HTER%)									
	SVM		SRC		MD			GMM	
	Spec	Indp	Spec	Indp	Spec	*Cs-Gb*	Indp	Spec	Indp
GoogleNet	16.17	36.35	16.45	18.15	4.04	5.92	17.19	15.89	16.56
ResNet50	11.99	41.66	21.06	19.54	**2.82**	5.23	15.59	14.44	15.12
VGG-VD-16	10.24	35.65	16.12	14.65	3.26	*4.8*	13.26	15.03	17.01
VGG-Face	17.33	46.5	19.11	17.45	5.68	7.27	12.75	9.96	12.79

Table 1. Presents the HTER (%) marked in bold of how the method proposed by Fatemifar et al. (2019) performed on the Replay-Attack dataset.

Fatemifar et al. (2019) conducted experiments on three large datasets, this showed enough data which allows for a clear justification of how the method performs. They went on to suggest how the performance of their detector can be enhanced with the use of client-specific detection thresholds. Their experiments were valid.

2.3 Multi-channel CNN

George et al. (2020) worked to develop a framework for the detection of presentation

attacks which aimed at detecting different types of 2D and 3D attacks which were in obfuscation or impersonation settings. On their research they proposed a new Multi-Channel CNN architecture which is good at merging information from multi-channel for prosperous presentation attack detection.

George et al. (2020) used a pre-trained Light CNN model which was trained on a massive number of face images. Capturing of both presentation attacks and genuine faces was done under the same conditions. On the experiment, they obtained four channels of data. All experiments were executed on the wide multi-channel presentation attack dataset. They made several experiments to check how their method performs on both "seen" and "unseen" attack's structure.

Researchers found out that combing many channels boosts the accuracy. They continue to stress that the fusion of channels makes the structure more vigorous in general and that the achievement of algorithms is lower when only using the colour channel. The performance of their algorithm exceeded the performance of another feature-based criterion. The researcher goes on to mention that a method with multiple channels on its own would not solve the problem at hand which is presentation attack detection.

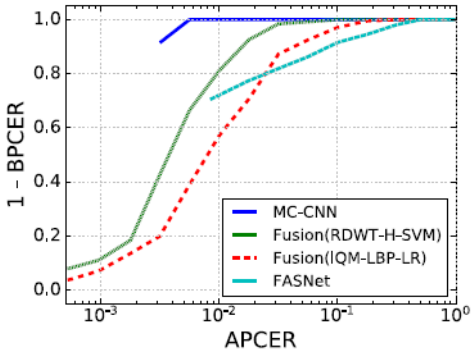


Fig. 3: Shows a comparison of the method proposed by George et al. compared to other baseline methods.

Data used it this research was carefully captured under the same conditions for both the genuine and fake faces. The dataset set used had a wide range of data which is necessary for testing how the proposed algorithm will perform. Researchers followed protocols like APCER and BPCER which other researchers are also using to test the detection accuracy of their face presentation attack detection systems. This proves that George et al. (2020) followed good science in testing their method and did not introduce any bias when presenting their results as they did not exclude negative results from their research.

Li et al. (2020) research was on studying the face liveness detection methods found on infrared and depth images. They proposed a framework which combined local and global features together to acquire multi-channel features.

They used the ACER metric, equal error rate metric and the area under the curve on the experiment they carried out. They tested their proposed method with the CASIA-SURF dataset. Li et al. (2020) concluded that multi-channel network would boost the achievement of the method they presented.

Name	ACER	EER	AUC	TPR@FPR=10E-2
MobileLiteNet	0.032	0.028	0.9962	0.941
FeatherNetA	0.026	0.025	0.9969	0.952
Our method	**0.024**	**0.024**	**0.9970**	**0.958**
GT	0	0	1	1

Table 2. Shows how the method proposed by Li et al. (2020) perfomed with different metrics.

Researchers mentioned all the steps they took to create the framework that they proposed even though the steps briefly stated it is still clear to other researchers who may be interested in doing that research. Results for this research are reproducible. They even went on to mention how the performance of that framework can be improved. Other researchers can add on what Li et al. (2020) had done. Therefore, the researchers followed valid science practice.

3 Comparison of methods

The research analysed throughout this section focused on improving the detection of spoofing attacks in face recognition systems with some applicable methods. Each of the methods had its strengths in certain areas and some weaknesses as well.

The approach to face spoofing detection that was presented by Das et al. (2019) performed very well on several different datasets. Their method was able to detect both low and high-quality image attack. However, if the model experiences overfitting during training it results in lower performance. This is because the model learns some deep features of the images and during that process the method also learns some of the noises that such an image has and that negatively affects the performance of their model.

For the detection of spoofing attacks Yao et al. (2020) presented a method which had most of its components being highly effective. Their method used texture features of a photo or video but due to the different patterns that photos and videos have, this leads to very lower performance of their model. A method with such limitation will not perform well at detecting unseen attacks.

George et al. (2020) presented a method which used multiple channels to learn deep network models without out being affected by the challenge of overfitting. Their model does not tolerate the absence of other channels as this negatively affects the execution of the method.

Das et al. (2019) method produced the best results and achieved great accuracy detection score as compared with the method proposed on Yao et al. (2020) research. On the other hand, George et al. (2020) proposed a method which outperformed the other methods compared above in terms of the ability to avoid any noises during learning of image features, but the method produced a lot of false negatives making it less reliable.

The research shows that each method has its own upper hand on the detection of spoofing attacks and some limitations as well.

4 Recommendation

Face recognition systems may encounter attacks which use images that have noise interference to fool such systems. Images with noise interference reduces the accuracy of the face spoofing detection method. Other conditions like variations in brightness and the color information may lead to the camera capturing noises.

Therefore, a combination of a high accuracy method consisting of the solution presented on Das et al. (2019) research and a multi-channel method which is not affected by noise interferences as proposed by George et al. (2020) is advised. The method proposed by George et al. (2020) also proved the ability to function well without the challenge of overfitting.

Having a method which is both accurate and not affected by noise interference will be useful in the real-world situations. The method will enable the face spoofing detection systems to detect spoofing attacks more accurately during any time of the day and in any condition. The system may also detect unseen attacks.

Further research experiments will be required to test if the above-mentioned methods work well together without encountering any problems.

5 Conclusions

This research paper analysed current research on the detection of spoofing attacks, most of the methods analysed were mostly based on convolutional neural networks. It is evident that methods which incorporate CNN stand a better chance of detecting face spoofing attacks on face recognition system.

Attackers keep coming up with ways that fool the detection methods on face recognition systems. Most of the methods had the limitation of being unable to avoid any noises that are captured by cameras. The method must be able to detect the type of an attack being performed but if there are any noises the system will not be able to pick if there is an attack.

Another limitation was that the methods on their own were unable to detect “unseen” attacks. Therefore, that affects the implementation of such methods on real-life face recognition application when used alone.

Some methods analysed in this paper allowed for fusion with traditional methods, other techniques and adding of multiple channels to improve the detection accuracy. Their experiments followed valid science practices and the results are reproducible. Other researchers could perfectly fuse a method proposed in Das et al. (2019) and George et al (2020) to have a more robust

detection method which has minor limitations and improved accuracies. This will require further research.

References

Alotaibi A, and Mahmood A, 2016, 'Enhancing computer vision to detect face spoofing attack utilizing a single frame from a replay video attack using deep learning', *2016 International Conference on Optoelectronics and Image Processing (ICOIP)*, Warsaw, pp. 1-5

Arashloo S. R, Kittler J and Christmas W, 2015, 'Face Spoofing Detection Based on Multiple Descriptor Fusion Using Multiscale Dynamic Binarized Statistical Image Features', *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 11, pp. 2396-2407

Das P. K, Hu B, Liu C, Cui K, Ranjan P and Xiong G, 2019, 'A New Approach for Face Anti-Spoofing Using Handcrafted and Deep Network Features', *2019 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, Zhengzhou, China, pp. 33-38

Fatemifar S, Arashloo S. R, Awais M and Kittler J, 2019, 'Spoofing Attack Detection by Anomaly Detection', *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, United Kingdom, pp. 8464-8468

George A, Mostaani Z, Geissenbuhler D, Nikisins O, Anjos A, and Marcel S, 2020, 'Biometric Face Presentation Attack Detection with Multi-Channel Convolutional Neural Network', *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 42-55

Li X, Wu W, Li T, Su Y, & Yang L, 2020, 'Face Liveness Detection Based on Parallel CNN', *Journal of physics: Conference Series*, United Kingdom, pp. 1-6

Raghavendra R, Venkatesh S, Raja K. B, Wasnik P, Stokkenes M and Busch C, 2018, 'Fusion of Multi-Scale Local Phase Quantization Features for Face Presentation Attack Detection', *2018 21st International Conference on Information Fusion (FUSION)*, Cambridge, pp. 2107-2112

Sepas-Moghaddam A, Malhadas L, Correia P. L and Pereira F, 2018, 'Face spoofing detection using a light field imaging framework', *IET Biometrics*, vol. 7, no. 1, pp. 39-48

Sun W, Song Y, Zhao H and Jin Z, 2020, 'A Face Spoofing Detection Method Based on Domain Adaptation and Lossless Size Adaptation', *IEEE Access*, vol. 8, pp. 66553-66563

Yao C, Jia Y, Di H and Wu Y, 2020, 'Face Spoofing Detection Using Relativity Representation on Riemannian Manifold', *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3683-3693

Zhang Wanling, Xiang Shijun, 2020, 'Face anti-spoofing detection based on DWT-LBP-DCT features', *Signal Processing: Image Communication*, Volume 89, pp. 1-9