INNOV 2023

The Twelfth International Conference on Communications, Computation, Networks and Technologies

ISBN: 978-1-68558-104-6

November 13th – 17th, 2023

Valencia, Spain

INNOV 2023 Editors

Anders Fongen, Norwegian Defence University College, Norge

Dragana Krstic, University of Niš, Faculty of Electronic Engineering, Serbia

INNOV 2023

Forward

The Twelfth International Conference on Communications, Computation, Networks and Technologies (INNOV 2023), held on November 13 - 17, 2023 in Valencia, Spain, aimed at addressing recent research results and forecasting challenges on selected topics related to communications, computation, networks and technologies.

Considering the importance of innovative topics in today's technology-driven society, there is a paradigm shift in classical-by-now approaches, such as networking, communications, resource sharing, collaboration and telecommunications. Recent achievements demand rethinking available technologies and considering the emerging ones.

The conference had the following tracks:

☐ Communications
☐ Networking
☐Computing
☐Web Semantic and Data Processing
☐Security, Trust, and Privacy

We take here the opportunity to warmly thank all the members of the INNOV 2023 technical program committee, as well as the numerous reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors that dedicated much of their time and effort to contribute to INNOV 2023. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations and sponsors. We also gratefully thank the members of the INNOV 2023 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope that INNOV 2023 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the areas of communication, computation, networks and technologies. We also hope that Valencia provided a pleasant environment during the conference and everyone saved some time for exploring this beautiful city

INNOV 2023 Steering Committee

Sean Sturley, University of the West of Scotland, UK
Yeim-Kuan Chang, National Cheng Kung University, Taiwan

INNOV 2023 Publicity Chair

Lorena Parra Boronat, Universitat Politecnica de Valencia, Spain
Sandra Viciano Tudela, Universitat Politecnica de Valencia, Spain
Jose Miguel Jimenez, Universitat Politecnica de Valencia, Spain

INNOV 2023

Committee

INNOV 2023 Steering Committee

Sean Sturley, University of the West of Scotland, UK
Yeim-Kuan Chang, National Cheng Kung University, Taiwan

INNOV 2023 Publicity Chair

Lorena Parra Boronat, Universitat Politecnica de Valencia, Spain
Sandra Viciano Tudela, Universitat Politecnica de Valencia, Spain
Jose Miguel Jimenez, Universitat Politecnica de Valencia, Spain

INNOV 2023 Technical Program Committee

Lavanya Addepalli, Universitat Politecnica de Valencia, Spain
Amjad Ali,University of Swat, Pakistan
Eugen Borcoci, University "Politehnica"of Bucharest (UPB), Romania
Constantin Caruntu, Gheorghe Asachi Technical University of Iasi, Romania
YK Chang, National Cheng Kung University, Taiwan
DeJiu Chen, KTH Royal Institute of Technology, Sweden
Yung-Yao Chen, National Taiwan University of Science and Technology (NTUST), Taiwan
Albert M. K. Cheng, University of Houston, USA
Karl Cox, University of Brighton, UK
Daniela D'Auria, Free University of Bozen-Bolzano, Italy
Panagiotis Fouliras, University of Macedonia, Thessaloniki, Greece
Marco Furini, University of Modena and Reggio Emilia, Italy
Laura García, Universitàt Politècnica de València, Spain
Nikolaos Gorgolis, University of Patras, Greece
Victor Govindaswamy, Concordia University Chicago, USA
Jens Grambau, HdM Stuttgart, Germany
Qiang He, Swinburne University of Technology, Australia
Mehdi Hosseinzadeh, Washington University in St. Louis, USA
Shih-Chang Huang, National Formosa University, Taiwan
Wen-Jyi Hwang, National Taiwan Normal University, Taipei, Taiwan
Sergio Ilarri, University of Zaragoza, Spain
AKM Kamrul Islam, North Carolina A&T State University, USA
Brigitte Jaumard, Concordia University, Canada
Thomas Jell, Siemens Mobility GmbH, Germany
Alexey Kashevnik, SPIIRAS, Russia
Khaled Khankan, Taibah University, Saudi Arabia
Vasileios Komianos, Ionian University, Greece
Igor Kotenko, SPIIRAS, Russia

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Two-Stage Object Detectors: A Comparative Evaluation

Jihad Qaddour
School of Information Technology
Illinois State University
Normal, IL, USA
jqaddou@ilstu.edu

***Abstract*— Object detection is a fundamental task in computer vision with many applications, such as self-driving cars, security, and medical imaging. Recent advances in deep learning have led to significant improvements in the performance of object detectors. This paper presents a comparative performance analysis of generic object detectors, focusing on two-stage detectors. Two-stage detectors are a type of object detector that first generates region proposals and then classifies and refines those proposals. The paper first provides an overview of the taxonomy of two-stage object detection algorithms. It then presents a detailed performance comparison of two-stage detectors on two datasets, Microsoft COCO and PASCAL VOC 2012. The results show that DetectoRS is a state-of-the-art two-stage object detector, outperforming all other two-stage models. However, it is also more complex. The more practical of the two-stage object detectors that performed well in the comparison are Neural Architecture Search-Feature Pyramid Network (NAS-FPN), cascade R-CNN, and Mask R-CNN.**

Keywords-deep learning; object detection; computer vision; two-stage detectors; and performance analysis.

I. INTRODUCTION

Object detection is a fundamental computer vision task that identifies and localizes objects in images or videos. It has recently received a great deal of attention due to its wide range of applications. Deep convolutional neural networks (CNNs) have enabled significant advances in object detection. CNNs can learn powerful features from images, which can be used to identify and localize objects accurately. This paper provides a comparative performance analysis of two-stage object detectors, such as Fast R-CNN [5], Spatial Pyramid Pooling (SPP) [11], Region-Convolution Neural Network (R-CNN) [7], Faster R-CNN 2], Mask R-CNN [21], and DetectoRS [24]. We also discuss the trade-offs in object detection. We believe that this paper will be valuable to researchers and practitioners who are interested in object detection.

The rest of this paper is organized as follows. Section II provides background on object detection. Section III discusses two-stage detectors. Section IV presents a comparative performance analysis of two-stage detectors. Section V concludes the paper with the future direction.

II. BACKGROUND

Object detection is a fundamental task in computer vision that has been revolutionized by deep learning in recent years. Deep learning models can learn powerful features from images, which can be used to identify and localize objects accurately. Object detection can be divided into four tasks:

- Object classification: Assign a class label to an object, such as "car" or "person."
- Object localization: Predict the bounding box of an object.
- Semantic segmentation: Assign each pixel in an image to a class label.
- Object instance segmentation: Predict the bounding box and class label of each object instance in an image.

Object detection is used in a wide range of applications, such as self-driving cars, video surveillance, and medical imaging.

A. Structure of Target Detection

Object detection can be divided into two approaches: region proposal-based detectors and single-stage detectors. Region proposal-based detectors generate region proposals (bounding boxes) and then classify each proposal into an object category [8]. Single-stage detectors directly predict the bounding boxes and class labels of objects in an image.

B. Historical Roadmap Taxonomy of Object Detectors

The development of object detection can be divided into two historical periods:

- Before 2012: This period is often called the traditional object detection period. During this time, object detection algorithms were primarily based on handcrafted features and shallow machine learning models.
- 2012 and after: This period is often called the deep learning-based detection period. During this time, object detection algorithms have been revolutionized by the introduction of deep learning models, such as R-CNN [7].

C. Challenges in object detection

One of the biggest challenges in object detection is dealing with complex scenes. Complex scenes may contain many objects, some of which may be partially occluded or

 ISBN: 978-1-68558-104-6

overlapping. Additionally, the objects in a scene may vary in size and scale. Another challenge in object detection is real-time performance. For many applications, such as self-driving cars, it is important to be able to detect objects in real time. However, deep learning models can be computationally expensive, which can make it difficult to achieve real-time performance.

D. Future of object detection

The future of object detection is bright. As deep learning models continue to improve, object detection algorithms will become more accurate and efficient. This will enable object detection to be used in a wider range of applications.

Additionally, researchers are exploring new ways to improve the performance of object detection algorithms. For example, some researchers are developing new deep learning architectures that are specifically designed for object detection. Others are developing new training techniques to help deep learning models learn to detect objects more effectively.

Overall, object detection is a rapidly evolving field with a bright future. As deep learning models continue to improve, object detection algorithms will become more accurate, efficient, and versatile.

III. TWO-STAGE OBJECT DETECTORS

Region-based object detection is inspired by the human visual system, which scans images and focuses on regions of interest. R-CNN [7] was the first region-based detector to show that CNNs are better than handcrafted features, such as HOG, for object detection. In this paper, we review many two-stage detectors that have been proposed since R-CNN.

A. R-CNN object detection

The R-CNN object detection model [7] is a region-based approach that was the first to demonstrate the superiority of convolutional neural networks (CNNs) over handcrafted features, such as HOG.

R-CNN works as follows:

1. Region proposal generation: R-CNN uses selective search [15] to generate 2000 region proposals from an image.
2. Feature extraction: R-CNN extracts 4096-dimensional features from each region proposal using a pre-trained CNN.
3. Classification and localization: R-CNN uses a linear SVM to classify each region proposal and predict its bounding box.

R-CNN has several limitations, including:

- Slow testing speed: R-CNN has to recalculate the CNN for each region proposal, which adds to the testing time.
- Time-consuming training: R-CNN has to fine-tune the CNN on a dataset of region proposals.
- High memory usage: R-CNN has to store the features extracted from each region proposal.
- Prone to overfitting: R-CNN is prone to overfitting, as the region proposals generated by selective search are not always accurate.
- Object localization errors: R-CNN uses bounding boxes to localize objects, which can lead to errors, as the boxes may not be perfectly aligned with the objects.

Researchers have proposed several solutions to these limitations, such as:

- The MCG system [7]: This system uses a variety of techniques to generate region proposals, which helps to reduce the risk of overfitting.
- The GOP system [27]: This system uses a geodesic-based segmentation technique to split the voters, which helps to improve object localization.
- Edge box techniques: These techniques return objects with fewer outlines crossing their bounds, which helps to reduce object localization errors [28].
- Pre-extracted reranking: This method removes duplicate region proposals from the recommendation lists, which helps to improve the accuracy of object detection.
- Semantic segmentation [29]: This technique can be used to improve object localization by providing more accurate information about the objects in an image.

The R-CNN object detection model is a landmark paper in the field of computer vision. It was the first to show that CNNs could be used to achieve state-of-the-art results in object detection. However, R-CNN has several limitations, such as slow testing speed and high memory usage. Researchers have proposed several solutions to these limitations, which have led to the development of more efficient and accurate object detection models.

B. SPP-Net object detection

He et al. [11] proposed Spatial Pyramid Pooling (SPP)-Net to address the limitations of R-CNN, such as the loss of object content and geometric deformation caused by cropping and wrapping.

SPP-Net uses spatial pyramid pooling to create a new CNN design that allows the SPP layer to be reused for different region proposals, regardless of their size. This makes SPP-Net more efficient and scalable than R-CNN.

SPP-Net has been shown to achieve better results than R-CNN, especially when the corresponding scale of different region proposals is precisely determined. However, SPP-Net can be slower than R-CNN at test time due to the pooling computation expenses.

Overall, SPP-Net is a significant improvement over R-CNN, and it has laid the foundation for many modern object detection algorithms.

 ISBN: 978-1-68558-104-6

C. Fast R-CNN object detection

Fast R-CNN [5] addresses the limitations of R-CNN and SPP-Net, such as the need to train different systems individually and the high storage capacity requirements. Fast R-CNN works as follows:

1. Feature extraction: Fast R-CNN extracts a single feature map from the entire image using a CNN.
2. Region proposal generation: Fast R-CNN uses a region proposal network (RPN) to generate region proposals from the feature map.

Classification and localization: Fast R-CNN uses a single linear Support Vector Machine (SVM) to classify each region proposal and predict its bounding box.

Fast R-CNN is more efficient and accurate than R-CNN and SPP-Net, and it has become the basis for many modern object detection algorithms.

D. Faster R-CNN object detection

Faster R-CNN [4] addresses the limitations of previous object detection algorithms, such as the need for external region proposal generation methods and the slow speed of Fast R-CNN. Faster R-CNN introduces a region proposal network (RPN) that is fully integrated into the CNN architecture. The RPN generates region proposals directly from the CNN feature maps, which eliminates the need for external region proposal generation methods. Faster R-CNN also uses a single-stage training procedure, which further improves speed. In a single pass, the RPN generates region proposals, and the CNN classifies and localizes the objects in the region proposals.

Faster R-CNN has achieved state-of-the-art results on many object detection benchmarks, and it has become the basis for many modern object detection algorithms.

E. Feature Pyramid Network (FPN)

FPN is a deep convolutional network that can generate high-level semantic features of varying sizes. It is a flexible and powerful tool for computer vision tasks. It can be used in various applications, including object detection, instance segmentation, key point detection, image classification, and semantic segmentation. FPN uses a top-down approach to combine features from higher and lower levels of the network. This allows FPN to preserve high-level semantic information while also providing fine-grained details. FPN can be used with any CNN architecture and has been shown to improve performance on various computer vision tasks.

One of the main advantages of FPN is that it can achieve state-of-the-art performance on object detection tasks. This is because FPN can generate features at multiple scales, which allows it to detect objects of different sizes. FPN is also not tied to a specific CNN architecture, which makes it more flexible and adaptable. This means that FPN can be used with various CNN architectures. FPN has been shown to be effective in various computer vision applications. For example, FPN has been used to improve the performance of object detectors on the Microsoft COCO and Pascal VOC datasets. FPN has also been used to improve the performance of instance segmentation algorithms on the Cityscapes dataset.

Overall, FPN is a powerful and versatile tool for computer vision tasks. It is easy to implement and can be used with any CNN architecture.

F. R-FCN object detector

Region-based Fully Convolutional Network (R-FCN) [26] uses fully connected layers that share almost all processing across the entire image instead of convolutional layers for object detection. This makes R-FCN faster and more efficient than previous region-based detectors, such as Faster R-CNN.

R-FCN addresses the translation invariance problem by using position-sensitive score maps. These score maps are generated for each object category, and they indicate the likelihood of an object of that category being present at a particular location in the image. The position-sensitive score maps are combined with region proposals to generate bounding box predictions.

R-FCN can be easily adapted to fully convolutional image classifier backbones, such as Residual Networks (ResNets). This makes it easy to train and deploy R-FCN models. R-FCN has been shown to achieve competitive performance on the PASCAL VOC datasets. For example, R-FCN with ResNet-101 achieves 83.6% mAP on the 2007 set.

Finally, R-FCN consists of four convolutional networks:

1. The input image is passed through a CNN to obtain feature maps.
2. The feature maps are then passed to a region proposal network (RPN) to identify potential object locations.
3. The potential object locations are then passed to the R-FCN network, which generates position-sensitive score maps.
4. The position-sensitive score maps are then used to classify and regress the bounding boxes of the objects.

G. Mask R-CNN object detector

Mask R-CNN [14] is a deep learning algorithm that can perform both object detection and instance segmentation. It is an extension of Faster R-CNN and adds a branch for each region of interest (ROI) to predict segmentation masks. The mask branch is a small, fully convolutional network (FCN) that is added to each RoI and predicts a pixel-by-pixel segmentation mask. The FCN is trained to predict a binary mask for each pixel, indicating whether the pixel belongs to the object or not. Mask R-CNN uses a two-stage approach to instance segmentation:

1. The first stage uses the Region Proposal Network (RPN) to generate a set of candidate RoIs.
2. The second stage uses the mask branch to predict segmentation masks for each RoI.

 ISBN: 978-1-68558-104-6

Mask R-CNN also introduces a new RoI pooling layer called RoIAlign, which is designed to improve the alignment of RoIs with the original image regions. RoIAlign uses bilinear interpolation to sample the feature map at the RoI's center and four corners, which results in a more accurate alignment than the traditional RoI pooling layer because it preserves the spatial information of the RoI.
Mask R-CNN has been shown to be very effective, for instance, segmentation, achieving state-of-the-art results on several benchmarks. It is a simple and efficient algorithm that can be easily extended to other object detection and instance segmentation tasks, such as pedestrian detection, car detection, and instance segmentation of medical images.

Advantages of Mask R-CNN:

- Accurate and efficient instance segmentation.
- Can be easily extended to other object detection and instance segmentation tasks.
- Simple to implement.

Applications of Mask R-CNN:

- Object detection.
- Instance segmentation.
- Pedestrian detection.
- Car detection.
- Instance segmentation of medical images.

Finally, Mask R-CNN is a powerful and versatile instance segmentation algorithm that can be used for various tasks. It is easy to implement and can be extended to other object detection and instance segmentation tasks.

H. Cascade R-CNN

Cascade R-CNN [10] is an object detection model that uses a cascade of detectors to gradually increase the quality of hypotheses while ensuring that all detectors have access to a positive training set of similar size. This technique eliminates the quality mismatch between hypotheses and detectors during inference, which can lead to overfitting and reduced inference speed.

Cascade R-CNN consists of a series of detectors that are trained with increasing intersection of union (IoU) thresholds. This means that the first detector is trained to only detect objects with high IoU scores, while the second detector is trained to detect objects with lower IoU scores, and so on. This allows the Cascade R-CNN to gradually increase the quality of hypotheses while ensuring that all detectors have access to a positive training set of similar size.

Cascade R-CNN has been shown to be effective in reducing and eliminating overfitting. Overfitting occurs when a model learns the training data too well and is unable to generalize to new data. The Cascade R-CNN addresses this problem by training the detectors in a sequential manner. This means that the first detector is only trained on the most difficult examples, while the later detectors are trained on easier examples. This helps to prevent the model from overfitting to the training data. In addition to reducing overfitting, the Cascade R-CNN also improves the speed of inference. This is because the later detectors only need to process the examples that were not detected by the earlier detectors. This can significantly reduce the amount of time it takes to process an image.

Overall, Cascade R-CNN is a promising approach to object detection. It has been shown to be effective in reducing overfitting and improving the speed of inference. As a result, it is a promising technique for achieving high-quality object detection. Some of the applications of Cascade R-CNN are object detection, self-driving cars, robotics, and surveillance.

Finally, Cascade R-CNN is a powerful and versatile object detection model that can be used for various tasks. It is easy to implement and can be extended to other object detection tasks.

I. DetectoRS

Detection and Retrieval System (DetectoRS) [24] is a new object detection and retrieval system that combines the Recursive Feature Pyramid (RFP) and Switchable Atrous Convolution (SAC) techniques. It achieves state-of-the-art accuracy for object detection and instance segmentation on the COCO test-dev platform, with 55.7% box accuracy, 48.5% mask accuracy, and 50.0% PQ for panoptic segmentation. Key Features of DetectoRS include:

- Recursive Feature Pyramid (RFP): RFP is a new feature pyramid network that provides additional feedback connections from the Feature Pyramid Networks (FPN) into the bottom-up backbone layers. This allows for more efficient processing and allows the network to learn to use different atrous rates for different objects.
- Switchable Atrous Convolution (SAC): SAC is a new type of convolution that allows the network to learn to use different atrous rates for different parts of an object. This is useful for detecting objects of different sizes and shapes.

Advantages of DetectoRS include:

- State-of-the-art accuracy: DetectoRS achieves state-of-the-art accuracy on the MSCOCO test-dev platform for object detection, instance segmentation, and panoptic segmentation.
- Efficiency: DetectoRS is an efficient object detection system due to the use of RFP and SAC.
- Flexibility: DetectoRS is built on top of the Faster R-CNN framework and uses the ResNet-50 backbone network. This makes it flexible and adaptable to different needs.

 ISBN: 978-1-68558-104-6

Applications of DetectoRS include:

- instance segmentation: DetectoRS can be used, for instance, for segmentation tasks, such as medical image segmentation and scene parsing.
- Panoptic segmentation: DetectoRS can be used for panoptic segmentation tasks, which involve simultaneously detecting and segmenting all objects in an image.

Finally, DetectoRS is a powerful and versatile object detection system that can be used for a variety of tasks. It achieves state-of-the-art accuracy and efficiency and is built on top of a flexible and adaptable framework.

J. NAS-FPN

Neural Architecture Search-Feature Pyramid Network (NAS-FPN) [22] is a modified version of neural architecture search (NAS) that allows for feature fusion at different scales through top-down and bottom-up connections. It achieves state-of-the-art accuracy on object detection tasks while using less computation time than other methods.

Key Features of NAS-FPN

- Feature fusion at different scales: NAS-FPN uses a combination of top-down and bottom-up connections to fuse features from different scales. This allows the network to learn a more comprehensive representation of the input image, which leads to improved accuracy.
- Neural architecture search: NAS-FPN uses NAS to automatically search for the optimal network architecture. This allows the network to be tailored to the specific task at hand, which can lead to further improvements in accuracy and efficiency.

Advantages of NAS-FPN

- State-of-the-art accuracy: NAS-FPN achieves state-of-the-art accuracy on object detection tasks, outperforming other methods such as SSD and Mask R-CNN.
- Efficiency: NAS-FPN is more efficient than other methods, such as SSD and Mask R-CNN, while still achieving state-of-the-art accuracy.
- Flexibility: NAS-FPN can be used with various backbone networks, such as ResNet-50 and AmoebaNet. This makes it flexible and adaptable to different needs.

Applications of NAS-FPN

- Object detection: NAS-FPN can be used for a variety of object detection tasks, such as self-driving cars, robotics, and surveillance.
- Instance segmentation: NAS-FPN can be used for instance segmentation tasks, such as medical image segmentation and scene parsing.

Finally, NAS-FPN is a powerful and versatile object detection system that achieves state-of-the-art accuracy and efficiency. It is built on top of a flexible and adaptable framework, making it a good choice for a variety of tasks.

IV. COMPARATIVE PERFORMANCE ANALYSIS OF TWO-STAGE OBJECT DETECTORS

Two-stage object detectors are a type of object detector that uses two stages to detect objects in an image. The first stage typically involves generating a set of region proposals, and the second stage involves classifying and refining the bounding boxes of the proposed regions. Two-stage object detectors have been shown to achieve state-of-the-art performance on object detection tasks. However, there are a variety of different two-stage object detectors available, and it can be difficult to choose the best one for a particular task.

This paper presents a comparative performance analysis of several popular two-stage object detectors. It used the MSCOCO and PASCAL VOC 2012 datasets to evaluate the performance of the detectors. It also used the following metrics to evaluate the performance of the detectors:

- Average precision (AP): AP is a measure of the accuracy of an object detector. It is calculated by averaging the precision of the detector at different recall levels.
- AP0.5: AP0.5 is the AP when the predicted bounding box Intersection over Union (IoU) is greater than 0.5, and the ground truth.
- AP[0.5:0.95]: AP[0.5:0.95] is the average AP for IoU values from 0.5 to 0.95 in steps of 0.5.

The results of the comparative performance analysis are shown in Table 1 and Figure 1. The comparative performance analysis shows that DetectoRS is a state-of-the-art two-stage object detector. It achieves high accuracy on both the COCO and PASCAL VOC 2012 datasets, and it can handle a variety of object sizes and scales.

Other two-stage object detectors that performed well in the comparison include NAS-FPN, Mask R-CNN, and Cascade R-CNN. These models also use a variety of techniques to improve their performance, such as region proposal networks (RPNs), RoIAlign, and focal loss.

Overall, the results show that two-stage object detectors can achieve high accuracy on a variety of datasets and tasks. However, they can also be computationally expensive. As a result, it is important to choose the right model for the specific task at hand.

The following Table 1 and Figure 1 illustrate the comparative parameter values for different detectors using MSCOCO and Pascal VOC 2012 datasets using the average precision (AP) metric. The results showed that DetectoRS outperformed all other two-stage models in both AP0.5 and AP[0.5:95] on both datasets.

 ISBN: 978-1-68558-104-6

TABLE I. TWO-STAGE OBJECT DETECTORS PERFORMANCE COMPARISON ON MS COCO AND PASCAL VOC 2012 DATASETS AT SIMILAR INPUT IMAGE SIZES FOR THE TWO-STAGE OBJECT DETECTORS.

Detector & year	Backbone	Image Size	AP[0.5:0.95]	AP0.5	Merit and Limitations
R-CNN [7] 2014	AlexNet	224	-	58.50%	**Merit:** Faster R-CNN has improved performance on the PASCAL VOC datasets than HOG-based methods. **Limitation:** Faster R-CNN is slow to train because of its sequentially trained multistage pipeline, and training is expensive in terms of storage and time.
SSP-NET [11] 2015	ZFNet	Variable	-	59.20%	**Merit:** SPP-Net accelerates R-CNN without sacrificing performance. **Limitation**: SPP-Net inherits the disadvantages of R-CNN and only provides a small improvement in results.
Fast-R-CNN [5] 2015	AlexNet, VGGm, VGG16	Variable	-	65.70%	**Merit: Faster R-CNN enhances performance over SPPNet by designing RoI pooling layer and eliminating disc storage for features.** **Limitation: External RP computation becomes a bottleneck, making real-time applications sluggish.**
Faster-R-CNN [3] 2016	ZFNet, VGG	600	-	67.00%	**Merit**: Faster R-CNN proposes RPN and introduces multi-scale regression anchor boxes, making it faster than Fast RCNN without sacrificing performance. **Limitation:** Real-time detection is slow, and training is hard due to the sequential training process.
R-FCN [12] 2016	ResNet101	600	31.50%	53.20%	**Merit:** Mask R-CNN is a fully convolutional detector network that is faster than Faster R-CNN. **Limitation**: Mask R-CNN is still too slow for real-time use, and the training process is not streamlined.
FPN [13] 2017	ResNet-101	800	36.20%	59.10%	**Merit:** FPN is significantly faster and improved over several competition winners by using densely sampled image pyramids. **Limitation**: FPN is computationally expensive due to the use of densely sampled image pyramids.
Mask-R-CNN [14] 2018	ResNeXt101, ResNet101, FPN	800	39.80%	62.30%	**Merit:** Mask R-CNN is a refined version of the Faster R-CNN framework that can perform instance segmentation with an additional branch for mask detection in parallel with the BB prediction branch. **Limitation**: Mask R-CNN falls short of real time applications due to its computational complexity.
NAS-FPN [22] 2019	ResNet-50	1280	48.3	-	**Merit**: NAS-FPN exceeds Mask R-CNN with less computation time and achieves 2mAP accuracy in mobile detection, thanks to its combination of top-down and bottom-up connections. **Limitation**: NAS-FPN is still slow for real-time applications.
DetectoRS [24] 2020	ResNeXt-101	1333	53.30%	71.60%	**Merit**: DetectoRS makes a significant difference in terms of efficiency and effectiveness by achieving state-of-the-art accuracy for object identification and instance segmentation. **Limitation**: DetectoRS is still unsuitable for real-time detections due to its computational complexity.

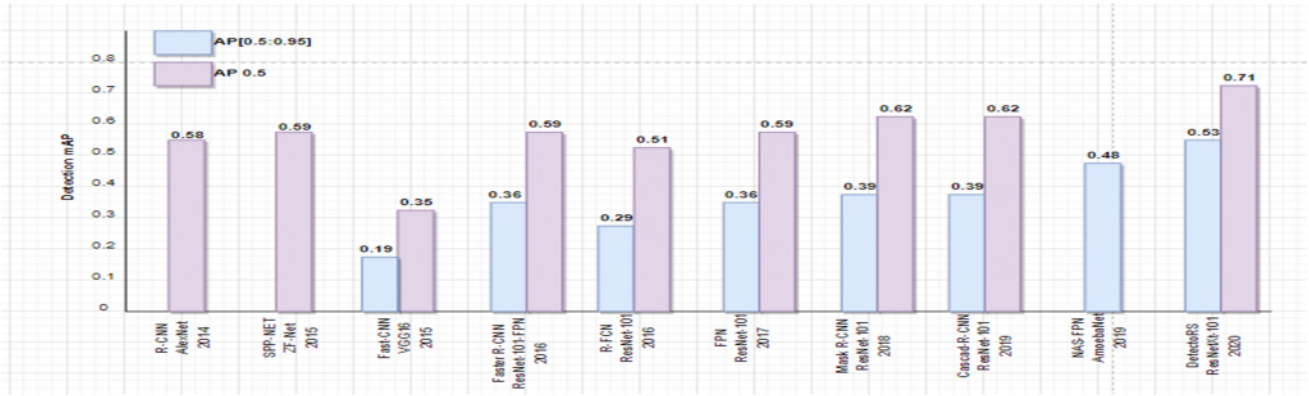

Figure 1. On the MSCOCO and PASCAL VOC2012 datasets, A comparative analysis bar graph of the performance of various two-stage object detectors.

V. CONCLUSION AND FUTURE WORK

This paper presented a comparative performance analysis of two-stage object detectors, which are state-of-the-art in object detection accuracy. The paper evaluated the performance of different detectors on two different datasets, MSCOCO and PASCAL VOC 2012, using the average precision (AP) metric. The results showed that DetectoRS outperformed all other two-stage models in both AP0.5 and AP[0.5:95] on both datasets. DetectoRS achieved an AP0.5 of 53.30% and an AP[0.5:95] of 71.60% on MSCOCO, and an AP0.5 of 83.00% and an AP[0.5:95] of 90.30% on PASCAL VOC 2012. However, it is also more complex.

Other two-stage object detectors that performed well in the comparison include NAS-FPN, Mask R-CNN, and Cascade R-CNN. These models also use a variety of techniques to improve their performance, such as region proposal networks (RPNs), RoIAlign, and focal loss.

Future research in object detection and recognition should focus on improving the speed of two-stage detectors without sacrificing accuracy, developing anchor-free detectors that are as accurate as anchor-based detectors but more computationally efficient.

REFERENCES

[1] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (voc) challenge," International Journal of Computer Vision. vol. 88, pp. 303–338, Jun 2010.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, pp. 1137–1149, June 2017.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788, June 2016.

[4] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks in Proc. Adv. Neural Inf.Process. Syst., 2015, pp. 91_99.

[5] R. Girshick, "Fast R-CNN," ICCV in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 1440_1448.

[6] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," Int.J. Comput. Vis., vol. 104, no. 2, pp. 154_171, Sep. 2013.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," In 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587, June 2014.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, Image net classification with deep convolutional neural networks," in Proc. Adv. Neural Inf. Process Syst., 2012, pp. 1097_1105.

[9] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2 Dpose estimation using part af_nity_elds," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jul. 2017, pp. 7291_7299.

[10] Z. Cai, and N. Vasconcelos, "Cascade R-CNN: High-Quality Object Detection and Instance Segmentation, arXiv:1906.09756 [cs.CV], June 2019.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition," IEEE Trans. Pattern Anal. Mach. Intel., vol. 37, no. 9, pp. 1904_1916, Sep. 2015.

[12] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and<0.5MB model size," arXiv:1602.07360v4, 2016.

[13] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in Proc. IEEE Conf.Comput. Vis. Pattern Recognit.," Jul. 2017, pp. 2117_2125.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in Proc.IEEE Int. Conf. Comput. Vis., Oct. 2017, pp. 2961_2969.

[15] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," Int.J. Comput. Vis., vol. 104, no. 2, pp. 154_171, Sep. 2013.

[16] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. "The Pascal Visual Object Classes Challenge 2012 (voc2012) Results (2012)," http://www.pascalnetwork.org/challenges/VOC/voc2011/workshop/index.html.

[17] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in Proc. 13th Eur. Conf. Comput. Vis. (ECCV). Zürich, Switzerland: Springer, Sep. 2014, pp. 740_755.

[18] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2016, pp. 3150_3158.

[19] J. Nan and L. Bo, "Infrared object image instance segmentation based on improved mask-RCNN," Proc. SPIE, vol. 11187, Nov. 2019, Art. no. 111871E.

[20] A. O. Vuola, S. U. Akram, and J. Kannala, "Mask-RCNN and U-Netensembled for nuclei segmentation," in Proc. IEEE 16th Int. Symp.Biomed. Imag. (ISBI), Apr. 2019, pp. 208_212.

[21] J. Li, X. Liang, J. Li, Y. Wei, T. Xu, J. Feng, and S. Yan, "Multistage object detection with group recursive learning," IEEE Trans. Multimedia, Vol. 20, no. 7, pp. 1645_1655, Jul. 2018.

[22] G. Ghiasi, T. Y. Lin, and Q. V. Le, "NAS-FPN: Learning scalable feature pyramid architecture for object detection," in Proc. IEEE Conf. ComputVis. Pattern Recognit., Jun. 2019, pp. 7036_7045.

[23] L. Aziz, M. S. B. Haji Salam, U. U. Sheikh, and S. Ayub, "Exploring Deep Learning-Based Architecture, Strategies, Applications and Current Trends in Generic Object Detection: A Comprehensive Review, in IEEE Access, vol. 8, pp. 170461-170495, 2020, doi: 10.1109/ACCESS.2020.3021508.

[24] S. Qiao, L.-C. Chen, and A. Yuille, "DetectoRS: Detecting objects with recursive feature pyramid and switchable atrous convolution," http://arxiv.org/abs/2006.02334, 2021.

[25] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deep Lab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," available: http://arxiv.org/abs/1606.00915.

[26] I. Krylov, S. Nosov, and V. Sovrasov, "Open Images V5 Text Annotation and Yet Another Mask Text Spotter, https://doi.org/10.48550/arXiv.2106.12326.

[27] B. Hariharan, R. Girshick, K. He, and P. Dollar, "Scalable, high-quality object detection using deep learning," In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 418–426).

[28] C. L. Zitnick, P. Dollár, "Edge boxes: Efficiently detecting salient object edges," "In European conference on computer vision. Springer.

[29] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, Mask R-CNN. arXiv preprint arXiv:1703.06870.

On-Demand Clock Boosting for Secure Remote Work System

Justus von der Beek
School of Computation, Information and Technology
Technical University of Munich
Munich, Germany
e-mail: beek@in.tum.de

Atsushi Shinoda
Graduate School of Informatics
Nagoya University
Nagoya, Japan
e-mail: shinoda@net.itc.nagoya-u.ac.jp

Hajime Shimada
Information Technology Center
Nagoya University
Nagoya, Japan
e-mail: shimada@itc.nagoya-u.ac.jp

Hirokazu Hasegawa
Center for Strategic Cyber Resilience Research and Development
National Institute of Informatics
Tokyo, Japan
e-mail: hasegawa@nii.ac.jp

***Abstract*—The global data center and networking infrastructure is projected to become the largest energy consumer by 2025, with high energy consumption contributing significantly to the climate due to greenhouse gas emissions. The trend of increased digitization accelerated this further, in particularly by Virtual Private Network and Video Conferencing network traffic, leading to higher CO2 emissions. In this paper, we address this challenge by analyzing and reducing the energy consumption of a secure remote working system. We propose a custom clock boosting mechanism using Dynamic Voltage and Frequency Scaling. Our two implementations, utilizing Extended Berkeley Packet Filter and eXpress Data Path, passively listen for new Transmission Control Protocol connections and adjust Central Processing Unit frequency when new employees connect. During idle periods, the frequency is minimized. Through this approach, we achieve up to 28% reduction in energy consumption during high load scenarios, while maintaining virtually no impact on consumption during idle phases. Additionally, the Quality of Service is improved, validating the effectiveness of our strategy.**

Keywords-VPN; eBPF; XDP; computer networking; energy efficiency

I. Introduction

Due to an increase in remote working and learning since 2019 [1], [2], Virtual Private Network (VPN) connections have been widely adopted, with VPN traffic experiencing a >200% increase during 2020 lockdowns [2]. This highlights the importance of secure data transfer for businesses and educational institutions. Shinoda et al. have developed an Access Control List (ACL) management mechanism for existing VPN software, which enhances the security of these VPN environments [3], [4]. The system achieves this by restricting access to critical files or servers for inexperienced users, thereby reducing the potential for hacking incidents.

This increase in digitalization poses both a challenge and an opportunity. On the one hand, the new demand of more internet users must be met with larger and more data centers. This also applies to company networks which were expanded during the pandemic [5]. However, it is no news that large data centers require high amounts of energy to operate [6]. Not only is the energy consumption a huge factor in the operating bill, but it also has a potentially high impact on the climate [7]. Liu et al. have estimated that data centers will be the largest global energy consumer in 2025 [8].

On the other hand, digitalization can lead to a decrease in greenhouse gas emissions [9]. In this paper, we deal with this challenge by analyzing the power consumption of our VPN system with dynamic ACL and attempting to minimize its operational energy consumption. In doing so, we strive to reduce energy costs and the environmental impact. To achieve this, we implement an on-demand Dynamic Voltage and Frequency Scaling (DVFS) controlling scheme for our VPN system and measure the potential energy savings and the impact on Quality of Service (QoS).

In Section II, related work is discussed and the optimized system introduced. Section III designs the clock boosting system and introduces two implementation ideas. The Section IV discusses the two implementation ideas in more detail. In Section V, the system is evaluated, followed by a conclusion and further work in Section VI.

II. Related Work

Because energy consumption is one of the primary cost factors in data centers, there exists a considerable amount of research focused on reducing it.

Krzywda et al. conducted experiments with DVFS schemes for data centers demonstrating its potential to reduce maximal energy consumption by up to 14% [10]. However, their approach involved setting a fixed frequency for a single experiment, which did not allow for dynamic updates at runtime based on application feedback. Additionally, the percent of energy savings they achieved came at the expense of an average performance reduction, which was at least twice as high.

Kasture et al. developed and implemented Rubik, a statistical performance model that utilizes DVFS to reduce the energy consumption in data centers hosting web search functionality [11]. They adapted the frequency to the lowest possible level while maintaining threshold latency. However,

 ISBN: 978-1-68558-104-6

their implementation is not available online to reproduce and evaluate against. Rebuilding this approach is out of scope for this paper. To keep the power consumption low, our idea can operate by keeping a counter of active Transmission Control Protocol (TCP) connections, making the expensive calculations of the paper needless.

The summary paper by Zhu et al. emphasizes the potential of DVFS schemes in conjunction with advancements in AI-driven prediction algorithms [12]. At present, the main disadvantages lie in the costs and computation difficulty of the prediction algorithm. In our work, we aim for a simple feedback solution with minimal calculation cost to reduce the energy consumption impact on the system.

A. Secure Remote Work System Overview

In the following, we provide a brief introduction to the secure remote work system that aim to optimize [3]. Figure 1 offers an overview of the key components for our paper, found in the system. This setup involves simulating a company network constructed by a Software Defined Networking (SDN). When an employee connects to the company network via VPN, the system applies a fine grained ACL to the SDN controller. The ACL's parameters depend on the user's reliability, enabling the blocking of access to high-risk data for unreliable users. The primary objective is to enhance the system's security by additional layers of security.

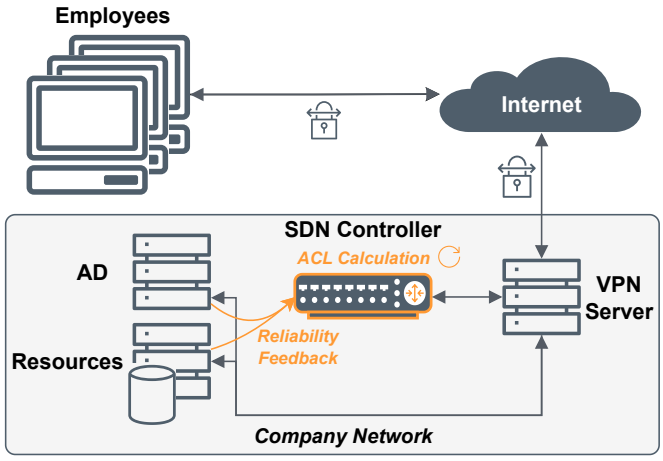

Figure 1. The mock up company network of Shinoda et al. We focus on the machine marked in orange, hosting the SDN-Controller because the CPU bound ACL computations allow for power savings.

To achieve this, a miniature network with VPN server, so-called VPN clients, active directories containing company mock data, and other resources is built. While not all parts of the network are relevant for our analysis, we have omitted them here for the sake of simplicity. If a user is not considered reliable enough, their access to files or servers will be blocked through the ACL permissions.

The main objective of this paper, is to reduce the overall system energy consumption by implementing a dedicated DVFS mechanism. The method also tries to minimize any adverse effects on performance. Given that the calculation of the ACL requires some time and involves CPU-heavy computations, our focus was mainly directed towards exploring potential power savings in the SDN-Controller.

III. Design of On-Demand Clock Boosting for Secure Remote Work System

Our SDN-Controller only operates at the beginning and end of a VPN connection. One significant challenge for our use-case lies in the uncertainty of when a new client will connect. Acknowledging that we cannot know the future, we opted for a reactive approach. Developing an extensive network prediction algorithm or model was beyond the scope of this work. Especially, as subsequent results revealed that the system achieved satisfactory latency with on-demand boosting. Nevertheless, we expect achieving better result by responding swiftly to any networking event. In Linux terms, our aim was to find a method positioned as low in the network stack as possible. Linux kernel version 3.15 and later features Extended Berkeley Packet Filter (eBPF) programs, which perfectly align with our idea. These programs can be attached at a low level in the network stack and have low overhead, making them an ideal fit for our requirements.

A. Extended Berkeley Packet Filter (eBPF)

Extended Berkeley Packet Filters enable the execution of special sandboxed programs within the kernel space, all without requiring any kernel modifications or modules [13]. These eBPF programs are coded in a C-like syntax and are compiled into what is known as eBPF bytecode. Upon loading, this code undergoes thorough verification and checks for potential errors such as out-of-bounds memory accesses or potential infinite runtime scenarios. The presence of these checks ensures that eBPF programs cannot crash or cause deadlocks, making the execution inherently safe [13]. Furthermore, the eBPF loader provides the flexibility to switch the specific hook to which an eBPF program is attached. For instance, one such hook could be the Linux networking stack, while other attachment points include kernel tracepoints or system calls.

B. eXpress Data Path (XDP)

Due of the flexibility and advantages of eBPF, the eXpress Data Path (XDP) hook point was developed. It enables the attachment of eBPF programs at the driver or hardware level, bypassing most of the network stack. As a result, this approach offers improved bandwidth and higher packet rates compared to the default kernel [14]. Nonetheless, it is important to note that not all devices and drivers support the XDP hook point. To address this limitation, an emulation mode was introduced after the Linux sockets module, allowing eBPF programs to be attached at an XDP hook without offering the advantage in speed.

For easy communication between the program running in the kernel space and user space, eBPF maps are available [13]. These data structures are allocated in shared memory regions,

 ISBN: 978-1-68558-104-6

facilitating read and write operations from both locations. Utilizing these maps, both programs can effectively communicate with each other, or the XDP program can maintain and store state information.

It is worth mentioning that we deliberately chose not to use the Data Plane Development Kit (DPDK). While DPDK's busy polling design might lead to even faster reaction times than XDP, it would come at the cost of significantly increased energy consumption. At least one CPU core would constantly operate at 100% utilization [15], rendering any potential energy savings irrelevant.

C. On-Demand Clock Boosting Design

The basic idea behind the clock boosting service is simple: we aim to provide the best available QoS while consuming as little energy as possible. To achieve this, we focus on utilizing the lowest feasible CPU frequency. In our test environment, the SDN switch is the only entity that connects to the SDN controller. No other connections are made to the machine. Consequently, all incoming TCP connections to the SDN controller originate from the SDN switch and are followed by the ACL calculation. Due to this fact, we can employ a straightforward check for new TCP connections directed to our controller. Upon detecting a new connection, we can boost the CPU frequency, thereby increasing the processing speed of the ACL calculation. Once the connection is reset, indicating that the processing is complete, we can then reduce the frequency back to the energy-saving level. Given that the system servers exclusively as an SDN-Controller, we have the condition to aggressively switch the CPU frequency back to power-saving mode without any performance implications. In the following, we propose two ideas to realize this design. One is based on the BPF Compiler Collection (BCC) [16] library while the other idea uses XDP.

D. Idea 1: BCC and cpupower

In our first approach, we have devised a system that attaches itself to the TCP/User Datagram Protocol (UDP) hook in the network stack, as illustrated at label 2 (Trigger on packet) in Figure 2. The process involves a new packet arriving on the network interface (label 1 in Figure 2), passing through the Linux network stack, and reaching the TCP/UDP module (label 2). Each time a packet passes through this module, it triggers the `tcpaccept` program which passively listens here. The `tcpaccept` program then invokes a shell script located in user space at label 3 in Figure 2. The shell script, temporarily increases the system's frequency for a short duration of 0.7 seconds to accelerate the calculations. We found this duration to be sufficiently long enough to maintain an equal QoS level. After this duration elapses, the performance governor is reverted to powersaving mode and the CPU frequency set to the minimal achievable frequency.

E. Idea 2: XDP, eBPF Maps and C-based Frequency Switch

The second approach utilizes XDP and an architecture overview is presented in Figure 3. Instead of employing

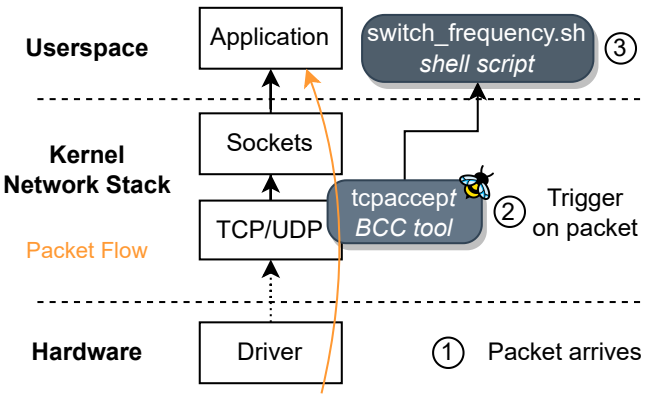

Figure 2. The eBPF BCC-based solution involves the `tcpaccept` program passively listening for incoming connections. When a connection is detected, it triggers a shell script to increase the frequency. After a set duration, the frequency is lowered again.

`tcpaccept` at the TCP/UDP module, we attach an eBPF program at the XDP hook, as shown in Figure 3 at label 1.

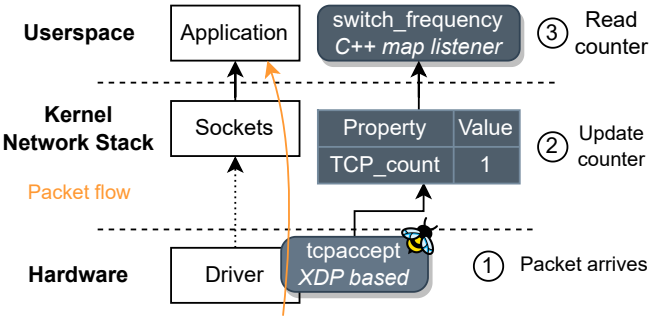

Figure 3. The XDP design involves an eBPF program that passively listens for new TCP connections and communicates with the user space frequency switch through shared memory maps. This communication also enables the system to detect disconnects.

Every time a packet arrives, the eBPF program is triggered and updates a counter variable in a shared memory map (labeled 2 in Figure 3). At the user space level, marked with label 3, we read the counter value and increase the frequency when at least one client is connected. Conversely, if all clients are disconnected, the frequency is decreased to the powersaving level.

IV. Implementation

In the following section, we will discuss the implementation details of the two solutions in greater detail and highlight advantages and disadvantages of each.

For our first idea we built upon Brendan Gregg's `tcpaccept` program [16]. It was originally designed to print statistics each time a new TCP connection is made. Since the executed user space program can be easily modified, we decided to base our implementation on this tool. For each new connection, we execute a shell script that utilizes `cpupower frequency-set` to switch to the highest available frequency and change the power governor to the `performance`

```
1: if packet is TCP then
2:     if packet.flags is SYN then
3:         map[TCP_count] ← map[TCP_count] + 1
4:     else if packet.flags is FIN then
5:         map[TCP_count] ← map[TCP_count] − 1
6:     end if
7: end if
```

Figure 4. The XDP TCP packet listener program

```
1: prv_count ← 0
2: while true do
3:     count ← map[TCP_count]
4:     if count ≥ 1 and prv_count = 0 then
5:         boost_frequency()
6:         prv_count ← count
7:     end if
8:     if count = 0 and prv_count ≥ 1 then
9:         reset_frequency()
10:        prv_count ← count
11:    end if
12: end while
```

Figure 5. The XDP program user space map listener

mode [17]. After a 0.7-second interval, the frequency is reverted to the lowest possible value and the governor is reset to `powersave` mode. To allow for changes to the frequency, this script must be run as root. In total, the implementation cost of this approach is less than 40 lines of code.

In contrast, the second solution utilizes XDP to parse incoming Ethernet packets and identify those carrying the TCP protocol. If a TCP packet is found, it is further checked for the SYN or FIN flags. When a SYN flag is detected, the number of active TCP connection is increased, while a FIN flag results in a decrease of active connections as shown in Figure 4. We maintain a counter in the shared memory map between XDP and user space which the XDP program updates (lines 3 and 5 in Figure 4). This counter informs the user space program about the current active connections and enables it to make decisions about the frequency to use. If there is one or more TCP connection, we switch to the performance governor and the highest frequency as shown by the if at line 4 in Figure 5. However, if there are none, we reset the governor to powersave mode and switch back to the lowest frequency (line 8 onward in Figure 5). Both of these actions are accomplished by writing the frequency and governor directly into the *sysfs*. This solution requires root privileges to load the XDP program and roughly 600 lines of new implementation in total. The user space listener can be loaded without root privileges required.

A. Disadvantages of Proposed Systems

One disadvantage of the first system is that in only reacts to TCP connections and does not monitor disconnects. Additionally, the solution introduces an extra of indirection through the shell script instead of directly writing to the MSR registers. This indirection adds time from the connection detection to the actual frequency increase. On top, the user space handling of the BCC tool is written in python which is an interpreted language with a higher overhead than C. It requires more resources than a C program and has a slower reaction time.

The major drawback of the XDP-based system, in our case, lies in the lack of XDP driver support from our Ethernet card. As a result, we have to rely on the emulation mode after the Socket module in the Linux network stack. Unfortunately, this leads to a reduction in reaction time instead of the desired improvement.

B. Advantages of Proposed Systems

Both systems share the advantage of low impact during idling since the programs are only triggered when an event occurs and they do not execute otherwise. This idle-aware design reduces the energy consumption making it an efficient solution.

Additionally, the XDP solution provides the advantage that it listens to disconnect events. This capability enables us to determine the duration for which we need to increase the CPU frequency. When multiple connections are active, we increase the frequency further to help handling all clients. Given that connections are only established when SDN-Controller decisions need to be made, the higher frequency will accelerate the processing enabling us to maintain a better QoS.

V. Evaluation

This section is split into the evaluation setup and the testing results.

A. Evaluation Setup

The SDN-Controller system runs on a HP ENVY x360 laptop with AMD Ryzen 3700U CPU with 4 cores, SMT enabled, TDP of 15 W and 16 GB RAM. We tested both with Turbo Core enabled and disabled to see the impact on the power consumption and performance.

The driver (`acpi-cpufreq`) allows for 3 frequencies with 1.4 GHz, 1.7 GHz and 2.4 GHz. Furthermore, the `performance`, `powersave` and `ondemand` governors are available [18]. We are using RockyLinux running kernel version 6.3.9-1. The ryu-manager version is 4.34. The Ethernet card is a Buffalo LUA4-U3-AGTE-NBK USB3.1 Gigabit Ethernet Card with driver version ax88179_178a. The VPN-Server deploying the VPN functionality is a NEC IX2310 running firmware version 10.6.63. Energy Consumption is measured via the Running Average Power Limit (RAPL) interface using the `turbostat` tool. Different research has shown that this interface is accurate enough for comparison of benchmark runs on the same system [19]. We are using 9 clients that simulate employees wanting to connect to the company internal network via VPN. All clients are using Windows 10.

The benchmark consists of all clients running a powershell script that uses `rasclient` to connect and disconnect to the company network as fast as possible. The time measurement

 ISBN: 978-1-68558-104-6

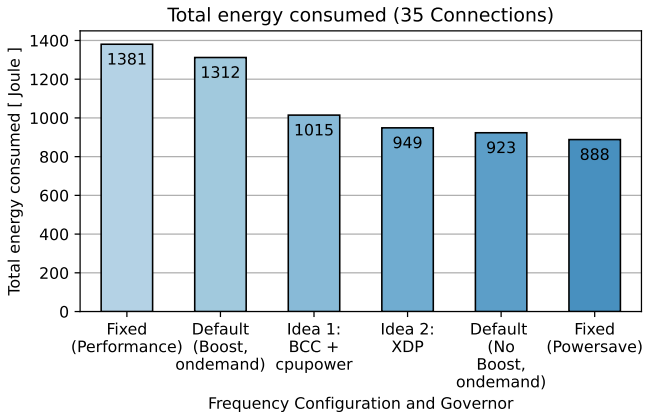

Figure 6. The VPN connection test total energy consumption. Both ideas can reduce the consumption by more than 23% compared to the default case with ondemand governor and Turbo Core enabled (named Boost in the Figure).

is done via the powershell internal `Measure-Command`. With the VPN connection and disconnection time, we try to measure a relevant QoS metric because users will notice a delayed VPN connection time as soon as they start working. Furthermore, this test should show the impact on the performance due to modified CPU frequency handling. File transfers, however, are not impacted by the VPN-controller and are therefore not tested.

Currently, the Ethernet network card of our SDN-Controller does not support XDP driver offload. Therefore, the slower SKB attachment mode was used during development and evaluation. We would expect better results for the native XDP mode because of a faster reaction time.

B. Evaluation Results

The accumulated power consumption of our benchmark runs for different frequency configurations is plotted in Figure 6. Both the BCC and XDP version of our proposed system can reduce the energy consumption during the high load scenario by more than 23% compared to the `ondemand` governor with Turbo Core enabled. We consider this the default version because this configuration is loaded on start up without any modifications to the system. The XDP version is less resource demanding than the python implementation, decreasing the consumption further by 5%. We make two noticeable observations. First, the performance mode only increases the power consumption by 5% hinting a very aggressive CPU frequency selection by the Operating System (OS). Secondly, disabling the frequency boosting technology saves 2% more energy than our XDP implementation. Krzywda et al. [10] approach would likely achieve results similar to the `ondemand` governor with Turbo Core disabled in Figure 6.

When looking at the idle consumption in Table I the proposed system does not increase the total consumption. Due to the event based triggers of eBPF and XDP the impact on energy consumption is only during high load phases. There are no energy savings during the idle periods but also no additional costs. This is important because most of the time the system will stay in an idle phase. An increase in the consumption during this phase would therefore outweigh savings during high load.

TABLE I
THE IDLE POWER CONSUMPTION (120S IDLE)

Mode	Default	XDP	BCC
Energy Consumption	256.62 J	262.02 J	266.62 J
Relative Consumption	100%	102%	104%

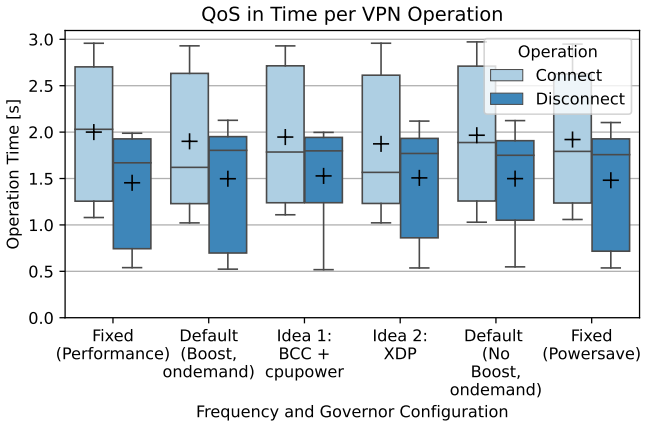

Figure 7. The VPN-Connection test connection time. Outliers larger than 4s and smaller than 0.5s were excluded. The + denotes mean, the horizontal bar median. The whiskers show the 95th percentile.

Switching the CPU frequency has an impact on the time it takes to calculate the ACL rules and inform the SDN-Switch about the final decision, which determines whether access is allowed or blocked. Figure 7 presents the mean connection and disconnection times in seconds, accumulated from all 9 clients, in a boxplot. The x-axis displays the different configuration modes, while outliers greater than 4 s or smaller than 0.5 s are excluded from the analysis as they indicate errors. The mean value is indicated by the + marker, the horizontal line indicates the median value. Interestingly, the results show that there are no significant differences in connection and disconnection times among the various configuration modes. However, our first implementation idea increases the connection times by 2.4% while achieve less power savings than the `powersave` and no boost configuration, as evidenced in Figure 7. For the default powersaving mechanisms a 1-3% increase in mean connection time can be observed. For the XDP version, the mean connection time is 1.87 s compared to 1.90 s in the default no boost case and 1.97 s in the powersaving configuration. The difference in median and mean value for the XDP configuration seem to come from slower outliers which degrade the median connection time of 1.55 s. Effectively, we spend 5% more energy for a 3% mean connection time decrease compared to the no boost configuration. For the XDP implementation we would expect better results when switching to the driver offloaded eBPF program. This could decrease the reaction time and lead to faster connection times.

 ISBN: 978-1-68558-104-6

The chosen benchmark marks a best case scenario for our proposed system. Because the CPU frequency in idle phases is automatically reduced by the operating system itself we improve due to the more aggressive frequency switching. Furthermore, we benefit from the knowledge, that the CPU is not required in the intervals between connections allowing for the fast frequency reduction. Because the system is solely used as a SDN-Controller the power saving is not impacting performance of any other program.

VI. Conclusion and Future Work

In this paper, we presented two implementations aimed at reducing the energy consumption of a secure remote work system. The BCC-based design, along with the cpupower tool, achieved a decrease in energy consumption compared to the default power governor, but better results were obtained by disabling Turbo Core. The XDP version, due to its lower resource impact in user space, further reduced power consumption. Compared to the default configuration, a 1.5% decrease in connection time was achieved, and compared to the disabled Turbo Core configuration, connection times were 5% faster. However, this required approximately 3% more energy compared to the ondemand governor with disabled Turbo Core.

Future work in the system could involve migrating the SDN-Controller into a virtualized environment to reduce required components and idle power consumption, while still meeting QoS requirements under high workload scenarios. Additionally, we expect the frequency switch to work on the SDN-Switch, further reducing consumption during file transfers by throttling the frequency to match maximum network bandwidth. Another potential method of power reduction would be to adapt devices to workday patterns and implement high power-saving sleep modes for infrequently used computers.

Acknowledgment

This research was supported by JSPS KAKENHI Grant Numbers JP23H03396, JP19K20268.

References

[1] T. Favale, F. Soro, M. Trevisan, I. Drago, and M. Mellia, "Campus traffic and e-learning during COVID-19 pandemic," vol. 176, article 107290.

[2] A. Feldmann *et al.*, "The lockdown effect: Implications of the COVID-19 pandemic on internet traffic," in *Proceedings of the ACM Internet Measurement Conference*. ACM, pp. 1–18.

[3] A. Shinoda, H. Hasegawa, Y. Yamaguchi, H. Shimada, and H. Takakura, "Implementation of access control method for telecommuting communication based on users' reliability," in *Proceedings of Computer Security Symposium*, pp. 840–847.

[4] H. Hasegawa and H. Takakura, "A dynamic access control system based on situations of users," in *Proceedings of the 7th International Conference on Information Systems Security and Privacy*, pp. 653–660.

[5] F. Almeida, J. Duarte Santos, and J. Augusto Monteiro, "The challenges and opportunities in the digitalization of companies in a post-COVID-19 world," vol. 48, no. 3, pp. 97–103.

[6] Z. Song, X. Zhang, and C. Eriksson, "Data center energy and cost saving evaluation," vol. 75, pp. 1255–1260.

[7] J. Ferreira, G. Callou, A. Josua, D. Tutsch, and P. Maciel, "An artificial neural network approach to forecast the environmental impact of data centers," vol. 10, no. 3, p. 113.

[8] Y. Liu *et al.*, "Energy consumption and emission mitigation prediction based on data center traffic and PUE for global data centers," vol. 3, no. 3, pp. 272–282.

[9] J. Sutton-Parker, "Determining commuting greenhouse gas emissions abatement achieved by information technology enabled remote working," vol. 191, pp. 296–303.

[10] J. Krzywda, A. Ali-Eldin, T. E. Carlson, P.-O. Östberg, and E. Elmroth, "Power-performance tradeoffs in data center servers: DVFS, CPU pinning, horizontal, and vertical scaling," vol. 81, pp. 114–128.

[11] H. Kasture, D. B. Bartolini, N. Beckmann, and D. Sanchez, "Rubik: fast analytical power management for latency-critical systems."

[12] H. Zhu *et al.*, "Future data center energy-conservation and emission-reduction technologies in the context of smart and low-carbon city construction," vol. 89, p. 104322.

[13] What is eBPF? an introduction and deep dive into the eBPF technology. [Online]. Available: https://ebpf.io/what-is-ebpf/

[14] D. Scholz *et al.*, "Performance implications of packet filtering with linux eBPF," in *2018 30th International Teletraffic Congress (ITC 30)*. IEEE, pp. 209–217.

[15] S. Gallenmuller, P. Emmerich, F. Wohlfart, D. Raumer, and G. Carle, "Comparison of frameworks for high-performance packet IO," in *2015 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*. IEEE, pp. 29–38.

[16] B. Gregg, S. Goldshtein, T. Qin, and H. Chen, "BPF compiler collection (BCC)." [Online]. Available: https://github.com/iovisor/bcc

[17] Linux User's Manual, *cpupower-frequency-set(1)*.

[18] D. Brodowski. CPU frequency and voltage scaling code in the linux(TM) kernel. [Online]. Available: https://www.kernel.org/doc/Documentation/cpu-freq/governors.txt

[19] K. N. Khan, M. Hirki, T. Niemi, J. K. Nurminen, and Z. Ou, "RAPL in action: Experiences in using RAPL for power measurements," vol. 3, no. 2, pp. 9:1–9:26.

 ISBN: 978-1-68558-104-6

The Past and Possible Future Development of Password Guessing

Ze-Long Li
University Of Jinan
UJN
Shandong Province, China
Email: 202121200928@stu.ujn.edu.cn

Teng Liu
University Of Jinan
UJN
Shandong Province, China
Email: 202121200914@stu.ujn.edu.cn

Lei Li
Jinan Blue Sword Junxin Information Technology Co., Ltd
Shandong Province, China
Email: leilimoon@hotmail.com

***Abstract*—According to the China Internet Network Information Center (CNNIC), by December 2022, the number of Internet users in China has reached 1.067 billion. Recently, consulting firm Kepios pointed out that nearly 5 billion people worldwide are currently active on social networks. Nowadays although there are many methods of identity authentication: fingerprint recognition, facial recognition and static password, static password is still the most widely used identity authentication method. Most people usually set passwords too simple and easily cracked. This allows attackers to crack their passwords with less cost. Password guessing technology can generate large-scale password dictionaries, which can be used to evaluate the strength of passwords and encourage users to change their own passwords. With the development of deep learning, password guessing technology is also constantly breaking through. But few people provide systematic surveys, which allow us to systematically review the most advanced methods and avoid repetitive research. Firstly, we will conduct a comprehensive analysis of the development of password guessing technology to this day. Secondly, we will propose future feasibility research methods based on the latest technology to address the shortcomings of password guessing models.**

Keywords-deep learning; generative models; neural networks; normalization methods; network and information security.

I. INTRODUCTION

In the real world, there are three basic methods for authenticating users: 1) proving your identity based on what you know; 2) proving your identity based on what you have; 3) directly proving your identity based on unique physical characteristics. Currently, identity authentication based on passwords, especially static passwords, is still one of the most widely used authentication methods. The password set by users is always related to personal information, as it is easy to remember and can be set repeatedly for different accounts [30]. For example, according to Dojo's 2023 cracked password list, many people prefer to use pet names, lover birthday, and other information as passwords. Attackers can directly access this information through social media and public personal information, thereby stealing passwords. Therefore, although password settings are simple, their security is still an important issue.

Password security issues have long been a concern, and various websites have adopted different password setting rules to prevent users from using weak passwords. Forcing users to follow password rules has little impact on improving password strength, as evidenced by the three points (in the first paragraph) mentioned earlier. Therefore, research on password guessing attacks is necessary. Our goal is not to crack user passwords, but rather to provide password strength detection, allowing users to understand the strength of their passwords and prompting them to modify them. Password guessing can be divided into offline password guessing and online password guessing [38]: offline mode requires stealing password files in advance, conducting unrestricted attack attempts, and does not require cracking speed; Online mode must use the same login portal as the user, with a limit on the number of times. This article conducts a systematic investigation of password guessing technology and provides a detailed explanation of most models.

The main contributions of this article are as follows:

- A systematic review was conducted on the password guessing methods mentioned in the references, with some models providing method details.
- Introduce improvement methods based on the original model by class, and each method improves the original method.
- Discuss the limitations of password guessing and propose feasible future research directions based on new technologies.
- Mention three methods for optimizing password guessing.

The rest of this paper is organized as follows. Section II is the background and related work. Section III describes the models. Section IV proposes several future research directions. The conclusion closes the article.

II. BACKGROUND AND RELATED WORK

As early as 1979, Robert Morris and Ken Thompson mentioned two attacks that are very familiar in the field of information security: violent cracking (a method of cracking passwords by calculating them one by one until the true password is found.) and Dictionary attack in their paper on UNIX password security [6]. The disadvantage of the former is that it is very time-consuming, while the latter requires a large amount of memory. According to the shortcomings of the two methods, in 1980, Hellman proposed a time-memory trade-off (TMTO) method, which allows people to balance

 ISBN: 978-1-68558-104-6

time and memory costs [12]. In 2003, in order to reduce the number of calculations in the cryptanalysis process, Oechslin proposed a precomputation method - Rainbow table [16]. The Rainbow table is an improvement on the TMTO method.

The above is the most original method for password guessing. In 2005, Narayanan and Shmatikov proposed to apply the Markov model to password guessing [1], which is better than the Rainbow table method. Since then, password guessing has entered a "new era". The Markov model is a statistical model, and its most widespread application is speech recognition. Since 2005, with the continuous development of artificial intelligence, more and more experts have begun to pay attention to how Markov models can be optimized for better password guessing [3][14][18][19][47].

Probabilistic Context Free Grammar (PCFG) was originally used for syntactic analysis and is another traditional password guessing method after the Markov model. It was proposed by Weir et al. [4] in 2009. PCFG checks grammar structures (combinations of special characters, numbers, and alphanumeric sequences) and generates distribution probabilities, which are then used to generate candidate passwords.

When using PCFG, we need to consider the password structure, that is, the password setting rules. Therefore, we need to understand people's setting habits and website requirements, which are aimed at domestic and foreign users [31][32][44]. Most of the literature is about English-speaking users, and only a few studies have examined how non-English users choose passwords. In 2019, Wang et al. [32] conducted a comparative analysis of 73.1 million domestic passwords and 33.2 million English websites in real life, emphasizing the structural and semantic features of domestic password settings. Compared with foreign users, the passwords set by domestic users are less resistant to online guessing attacks, but better resistant to offline guessing attacks. Wang et al. [32] systematically discussed several basic attributes of passwords, such as the relationship between passwords and language, and found that there are great differences in letter distribution, structure and semantic patterns between domestic and foreign. PCFG and Markov models are used to attack, with the main purpose of enabling users to protect personal accounts more deeply. In the same year, Kaevrestad et al. [44] conducted research on the classification of password creation strategies. The main purpose was to better understand the password setting rules and better understand passwords. According to the survey summary provided by 21 experts, the password categories are divided into 7 categories: phrases; biographical passwords; leetspeak; dates; words; combination of words and numbers; random passwords. More specifically, it can be divided into four categories: only numbers; alpha numeric characters (numbers, small and large letters); special characters.

After 2011, the field of artificial intelligence has entered a booming period. Recursive Neural Network (RNN) model has been widely used in the field of Natural language processing. It can model based on time series data to process data, such as text prediction. In 2016, considering the inaccuracy of modeling password guessing at the time, Melicher's team [5] proposed using neural networks to simulate the resistance of text passwords to guessing attacks. This is another major progress after applying the Markov model to the field of password guessing in 2005. Traditional RNN may cause gradient explosion, so Melicher's team [5] chose to use Long Short Memory Network (LSTM) to solve the gradient explosion problem. Two years later, Zhang et al. [8] proposed a password cracking method based on structural partitioning and BiLSTM recurrent neural network. It is also the use of neural networks. In 2022, Ye et al. [11] applied time domain Convolutional neural network (TCN) to password guessing and added tag learning method. After a year, in 2023, Wu et al. [15] once again used TCN for password guessing and named it PGTCN, which can automatically study the structure and characteristics of passwords and generate new passwords based on the knowledge learned.

In 2014, Ian Goodfellow et al. [37] proposed Generative Adversarial Networks (GANs), which have powerful functions and have been studied and applied since their introduction [40][41][42][43]. Considering that the GANs model is a Generative model, it can be used for Natural Language processing. Therefore, in 2019, Hitaj et al. [9] applied the GANs model to password guessing, and its performance was superior. The GANs model has obvious drawbacks: it is difficult to train due to unstable training, vanishing gradients and pattern collapse when processing text data. Although the Hitaj team used wasserstein distance to slightly improve, its shortcomings are still evident.

III. MODEL EXPLANATION

This section provides a detailed explanation of password guessing models related to Markov, PCFG, and deep learning.

A. *Markov model family*

Markov chain can be traced back to 1906-1912, which was proposed by Markov and is an important concept in machine learning. Markov chain is a Stochastic process in the state space through the transition from one state to another. The probability distribution of the next state can only be determined by the current state, and the events before it in the time series are independent of it. This specific type of "memoryless" is called Markov property.

The following are the basic elements of a Markov chain:

1) State space: Let $X_n=i$ indicate that the state at time n is i, and the set of values of all states is called the "state space".

2)Transition probability: the Conditional probability from the state at the current time to a state at the next time is called "transition probability".

$$P_{ij}=P(X_n=j|X_{n-1}=i) \quad (1)$$

The above equation represents the probability of transitioning from state i to state j.

3) State-transition matrix: there may be more than one state at each time, so the transition probability between all states is formed into a matrix, which is called "State-transition matrix", and the size of the matrix is set to |I| * |I|. It should be noted that this matrix does not change over time.

4) Initial state: p_0.

In the Markov hypothesis, we need to use the N-Gram algorithm, which assumes that the occurrence of the nth word is only related to the first N-1 word and not to any other word. The probability of the entire sentence is the product of the probabilities of each word's occurrence. These probabilities can be obtained by directly counting the number of times N words appear simultaneously from the corpus.

In the zero order Markov model, the generation of the current character is independent of the previously generated character. In the first-order Markov model, each diagram (ordered pair) of characters is assigned a probability, and the current character is generated by looking at the previous character. Mathematically, in the zero-order model [1]:

$$P(\alpha)=\pi_{x\in\alpha}v(x) \quad (2)$$

In the first-order model:

$$P(x_1x_2...x_n)=v(x_1)\pi_{i=1}^{n-1}v(x_{i+1}|x_i) \quad (3)$$

The reason why Markov model can be used for password guessing is that a Markov model defines a probability distribution on a symbol sequence. In other words, it allows for sampling of character sequences with certain attributes.

The drawbacks of the Markov model are also evident, as it generates a large amount of duplicate data when cracking passwords, resulting in high repetition rates and low coverage, resulting in resource waste (as shown in Table I). A new method based on Markov model has been proposed. In 2015, Dürmuth et al. [3] proposed a method using an ordered Markov enumerator (OMEN) based on the idea proposed by Narayanan, considering orderliness. Simply put, they generate candidate passwords based on the probability of their occurrence, and the first output is the one with the highest probability. OMEN has improved the speed of password guessing, and it is worth noting that it only approximates the likelihood of passwords.

The main parameters of OMEN include n-gram size, alphabet size, and Number of levels.

The main algorithm enumPwd (): At a high level, enumPwd () will discretize all probabilities into multiple bins, iterate each bin in descending order of probability, and output the password that matches the probability of the bin in each bin. For specific password lengths ℓ and level η. EnumPwd (η, ℓ) executes as follows:

Firstly, we need to calculate a vector **a**= (a_3,..., a_l) with a length of ℓ-1. Each a_i represents an integer within [0, nbLevel-1], and the sum of all elements is η. Because there are ℓ-1 elements, when using 3-grams, it is necessary to have ℓ-2 transition probabilities and 1 initial probability to determine the probability of a string of length ℓ. For example, the probability of a password with a length ℓ=7 is calculated as follows:

$$P(loveyou)=P(lo)P(v|lo)P(e|ov)P(y|ve)P(o|ey)P(u|yo) \quad (4)$$

For each such vector a, select 2-grams x_1x_2(all) and iterate through all x_3, with the aim of 3-grams $x_1x_2x_3$ to obtain the level value a_3. Next, iterate x_4 for each 3-gram to obtain the level value a_4, with the aim of 3-gram $x_2x_3x_4$ Continue this process until the expected length is reached, and the final output is a set of candidate passwords (length of ℓ, level of η).

When setting parameters ℓ and η, the setting of ℓ is quite difficult, as the problem arises from the password length during training and people's guesses about a specific length. Therefore, Dürmuth et al. [3] added an adaptive algorithm to track the success rate of different password lengths.

Although this method effectively improves the speed of guessing and the coverage rate, its results will not change no matter how many training parameters are determined, and it always generates the same password in the same order, which is a deterministic algorithm. Fully considering the advantages and disadvantages of ordinary Markov and OMEN, Guo et al. [18] proposed a dynamic mechanism called the dynamic Markov model in 2021. Compared with ordinary Markov and OMEN, this model reduced the repetition rate from 75.88% to 66.50% and increased the coverage rate from 37.65% to 43.49%.

The purpose of the dynamic mechanism is to reduce the repetition rate and to improve coverage.

1)This method is only suitable for random sampling.

2)Every time a password is generated, a dynamic mechanism is used to reduce the probability of its subsequent occurrence.

3)For any string C: m≤m_{MAX}, set the string space to S_S . Form a set of strings with probability values greater than 0 and define this set as a support set S_M , which is a subset of S_S.

4)For strings outside of S_M, we believe they have no cracking significance.

Dynamic mechanism principle: For any original distribution, we set it to $D_{original}$ and represent the number of passwords in $S_M$with N. We randomly select a password P from S_M, which needs to meet the following requirements:

$$p_i^{original}\times N\geq 1 \quad (5)$$

By a small parameter α reduces its probability, based on α/ (N-1) increasing the probability of other passwords, and the probability distribution D_{new} of the new password is closer to a uniform distribution D_{uni}.

Although the S_M size is assumed to be N, N is unknown. In practical operations, we cannot directly handle the entire password probability. The authors provide a simplified method to replace it, which is to only consider n-gram fragment. Figure 1 shows the process of dynamic Markov generating passwords.

There is also an application method of the Markov model, which combines the GAN model and will be explained in

TABLE I. THE NUMBER AND RATE OF REPETITIONS IN PASSWORD GENERATION BY MARKOV MODELS

password generation	10^6	10^7	10^8
Number of duplicate passwords	4.79×10^5	5.94×10^6	6.86×10^7
Repetition rate	47.93%	59.37%	68.61%

 ISBN: 978-1-68558-104-6

section D. Table II compares three Markov models for the coverage number of different probability ciphers.

B. Probabilistic Context Free Grammar family

Note that rule processing in dictionary-based password guessing is a difficult task. Therefore, Weir et al. [4] proposed a method based on PCFG to generate password structures in the highest probability order, which fundamentally considers the structure of passwords, i.e., the rules for password settings.

PCFG is an extension of Context Free Grammar (CFG), which is a method of Rule Based Natural Language Processing (NLP). The main function of CFG is to verify whether the input string conforms to a certain grammar G, which is similar to regular expressions, but CFG can express more complex grammars. CFG is a set of replacement rules, for example: 0->O indicates that variable 0 can be replaced by variable O.

PCFG only adds the probability associated with each generation, and all associated productions add up to 1. Weir et al. [4] used only L_n, D_n *and* S_n (L represents letters, D represents numbers, and S represents special characters.) for the specified n-value in grammar, except for the starting symbol. They call these variables alpha variables, digit variables and special variables respectively. Table III is an example of PCFG, based on which the pre-terminal structure can be further derived:

$$S \rightarrow L3D1S1 \rightarrow L34S1 \rightarrow L34! \quad (6)$$

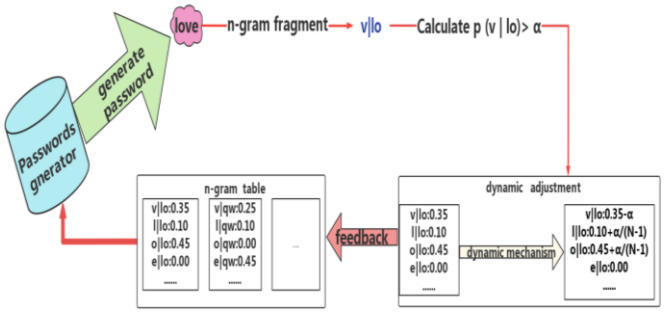

Figure 1. Dynamic Markov model generating password process.

TABLE II. THE COVERAGE OF THREE MARKOV MODELS

Probability	Total	Markov Model	OMEN	Dynamic Markov Model
$<10^{-12}$	310024	161	817	68
$[10^{-12},10^{-11})$	231826	1168	26988	980
$[10^{-11},10^{-10})$	334797	13507	1409272	13275
$[10^{-10},10^{-9})$	410065	95250	287433	122463
$[10^{-9},10^{-8})$	390731	272115	340801	350988
$[10^{-8},10^{-7})$	284911	271945	268945	284897
$[10^{-7},10^{-6})$	116095	115973	112954	116094
$[10^{-6},10^{-5})$	19827	19826	19456	19827
$[10^{-5},10^{-4})$	2701	2701	2671	2701
$[10^{-4},10^{-3})$	243	243	243	243
$[10^{-3},10^{-2})$	4	4	4	4

Keyboard order (keyboard mode) and multi word strategy can greatly improve the PCFG based password guessing method proposed by Weir et al. [4], which is an important supplement to PCFG [45]. This method was proposed by Houshmand et al. [45] in 2015. By learning the new model, it can achieve 22% improvement over PCFG, and the authors also defined metrics to help analyze and improve the dictionary. Increase the coverage of standard attack dictionary, achieving an additional increase of~33%. The keyboard mode does not consider the characters actually entered but is a shape that is easy to remember.

For example, “asdfg” is a series of keys from “a” to the next four letters on the right. Therefore, the keyboard mode is considered as a series of keys on the keyboard, and passwords can be created according to various combinations. The keyboard mode uses the symbol K as a new nonterminal character to be introduced into PCFG. Table IV is a comparison between Weir et al.'s [4] method and the other author's method. There are two considerations on how to determine whether it is keyboard mode or original (L, D, S) structure:

1)Pure numbers or special characters are classified as original structures and as components of D/S.

2)When it does not belong to the first item and the substring contains at least 3 characters, the keyboard mode requires the maximum length. For example, “asdfghui12” is classified as keyboard mode K8D2, not L6D4/K6D4.

Assuming a set of words is W $\{w_1...w_n\}$, considering it as a dictionary, and R is the cipher set $\{p_1...p_m\}$. If w is an L-structure in R, the password in R must have at least one w. If w is found in R, I (w, R) =1, otherwise I (w, R) =0. The accuracy definition of W for R is as follows:

TABLE III. PCFG EXAMPLE

Left-Hand Side	Right-Hand Side	Probability
S→	$D_1L_3S_2D_1$	0.75
S→	$L_3D_1S_1$	0.25
D_1→	5	0.60
D_1→	2	0.20
D_1→	1	0.20
S_1→	!	0.65
S_1→	%	0.30
S_1→	*	0.05
S_2→	&&	0.70
S_2→	$$	0.30

 ISBN: 978-1-68558-104-6

$$P(W,R)=\frac{1}{|W|}\sum_{i=1}^{n} I(w_i, R) \quad (7)$$

Assuming a password has k different L-structures, letting the count be c (w, p), where p is the number of L-structures, and the value is w. The coverage of word w is:

$$C(w, p)=\frac{c(w, p)}{k} \& C(w, R)=\sum_{i=1}^{m} C(w, p_i) \quad (8)$$

Set to a subset of passwords with at least one L-structure in R. The coverage of dictionaries W and R is as follows:

$$C(W,R)=\frac{1}{|R_L|}\sum_{i=1}^{n} C(w_i, R) \quad (9)$$

Through its development, PCFG not only allows for guessing passwords in probabilistic order, but also fully considers keyboard mode, resulting in higher cracking coverage. However, in practical applications, low probability passwords are still difficult to crack because they often lack semantic structure. Although lacking semantic structure, low probability ciphers also have a large search space and certain semantic information [13]. Therefore, considering the importance of improving the low probability password hit rate for offline attack efficiency, Guo et al. [13] proposed a degenerate distribution collection method in 2022 and designed a corresponding Low Probability Generator Probabilistic Context Free Grammar (LPG-PCFG) model based on PCFG. Compared with PCFG, LPG-PCFG aims to increase the distribution of low probability passwords, and when generating 10^7 and 10^8 passwords respectively, the number of hits increases by 50.4% and 42.0%.

Assuming the degenerate distribution as D_{deg} is the intermediate state between modeling and D_{uni}. The closer the degenerate distribution is to a uniform distribution, the better the distribution for generating low probability ciphers. However, passwords generated very close to each other may lack learning features, and measuring quality and low probability passwords is a challenge. There will be an optimal degenerate distribution D^{*}_{deg}, which can achieve a balance between modeling and uniform distribution, thus effectively generating low probability ciphers. Table V shows several probability correction rules aimed at obtaining a degenerate distribution, mainly through the following mechanisms:

Sampling to obtain password x^+, by sampling the generated model, modification probability:

TABLE IV. KEYBOARD BASE STRUCTURES VS PCFG

Passwords	PCFG	Keyboard
1234	D_4	K_4
w2w2	LDLD	K_4
ASD1234QW	$L_3D_4L_2$	$K_3D_4L_2$
Q1!2	LDSD	K_4

$$p(x^+) - \alpha \quad (10)$$

Support the probability of other passwords (x^-) in the set, where N_S represents the number of passwords supported in the set:

$$p(x^-)+\alpha(N_S-1) \quad (11)$$

Guo et al.'s [13] method enables low probability ciphers to also have good guessing performance, greatly improving the hit rate. The article mentions the semantic structure and low probability password semantic information but has not conducted in-depth research on them. It still uses passwords created by English or Chinese users for research. In June 2023, due to insufficient investigation of cryptographic semantic information, Wang et al. [46] proposed a general framework for PCFG based on semantic enhancement, named SE # PCFG. 43 types of semantic information are allowed to be considered for password analysis, which is by far the most abundant set. In addition, a Semantically Enhanced Password Cracking Architecture (SEPCA) was proposed by combining SE # PCFG with a smoothing method.

For better semantic analysis of passwords, the authors define four levels of password structures:

1)Character: The lowest level information about a password.

2)Semantic factor (SF): Some consecutive characters together form a semantic unit, which can be a word and carry semantic information called semantic factor type (SFT).

3)Semantic Pattern (SP): Consisting of one or more semantic factor types semantically, considering the entire password.

4)Semantic Structure (SS): Reflects the collective behavior of users, mapping shared and semantic attributes (language, website type).

Three step calculation process: preprocessing, identifying SFTs fragments, and post-processing.

The authors define a general PCFG as:

$$G=(M,T,R,S,P) \quad (12)$$

M and T represent non-terminal and terminal symbols. S is the beginning. R is the set of rules, and P is the probability contained in each rule R.

TABLE V. MODIFICATION RULES OF DEGENERATION DISTRIBUTION

Rule	Adjust $p(x^+)$	Adjust $p(x^-)$
Rule1	$p(x^+) - \alpha$	$p(x^-) + \alpha/(N_s - 1)$
Rule2	$p(x^+) - \alpha$	$p(x^-) + \alpha/(1 - p(x^+))p(x^-)$
Rule3	$\beta p(x^+)$	$p(x^-) + (1 - \beta)p(x^+)(1 - p(x^+))p(x^-)$
Rule4	$\beta p(x^+)$	$p(x^-)(1 - \beta)p(x^+)(1 - p(x^+))p(x^-)$
Rule5	$1-\gamma(1 - p(x^+))$	$\gamma p(x^-)$

In SEPCA, T is the set of all semantic factors, and M is the union of T and S. The rules are divided into two groups: from S to a certain SP; From one SFT to a certain SF.

At this point, the traditional password guessing methods, namely the Markov and PCFG models, as well as the improvements based on the two models, have been explained. Starting from section C, the application of neural networks in the field of password guessing is discussed. Table VI in section E summarizes the password guessing model.

C. Neural Network model family

Using Neural Networks to simulate the resistance of passwords to guessing attacks can be more effective than Markov models and PCFG. Neural network modeling uses less space than Markov models, and neural networks can transfer knowledge from a task to related tasks. Elements used in neural network models [5]:

1)Model architecture: Using recursive neural networks, character level text can be generated.

2)Alphabet size.

3)Password Context: Predicts characters related to the context (similar to Markov models).

4)Model size: Implementing with LSTM requires determining how many parameters are present in the model.

5)Transfer learning: Train a model about all passwords but adjust and guess longer passwords.

6)Data during training.

Melicher et al. [5] used an LSTM network, which could solve the gradient explosion problem caused by long text, including memory gates, forgetting gates, and output gates. Determine which information is discarded and which information is left through three "gates". The article proved that neural networks could guess passwords faster and more accurately. Since Melicher et al. [5] applied Neural Networks to password guessing, various Neural Network models, including various variants of LSTM, have been used to improve the hit rate and efficiency of password guessing [8][11][15][33].

In 2018, Zhang et al. [8] combined the advantages of PCFG and neural networks to propose a password guessing method based on structural partitioning and Bidirectional Long Short-Term Memory Recursive Neural Networks, named the SPRNN model. Firstly, divide the password into substructures, and then use the BiLSTM model to generate substrings based on the substructures, considering the accuracy and generalization ability of the model. The article points out that the SPRNN model performs well across datasets, with a hit rate of 25% to 30% higher than the general Markov model and 10% higher than the Weir et al. [4] method. In 2019, Li et al. [49] also applied BiLSTM to password guessing. In 2022, Chang et al. [33] addressed the difficulty of selecting sequence length in traditional LSTM models for password generation, and it is unclear whether there is a relationship between sequences of different lengths. Chang et al. [33] considered user personal information and proposed a multi sequence length LSTM password guessing model. Compared with traditional PCFG models, the hit rate has increased by 68.2%, and there is also an improvement of 7.6%~42.1% compared to traditional LSTM models. MLSTM consists of two stages: training stage and generation stage.

The method proposed by Chang et al. [33] addresses the length limitation of LSTM sequences, but the datasets used are all analyses of Chinese ciphers, and other datasets should also be considered in order to be more representative.

Ye et al. [11] proposed a password guessing model based on Time Convolutional Neural Network (TCN) (PassTCN).

Figure 2 introduces some Recursive Neural Networks and Convolutional Neural Networks.

TCN is an algorithm used to solve time series prediction. In order to further improve the performance of password generation, a new password probability label learning method is also proposed. Figure 3 shows the password guessing structure based on the TCN model.

The password probability label learning proposed by Ye et al. [11] is based on the probability distribution of the password and constructs a unique password label based on the probability distribution in the training set. Firstly, it is necessary to calculate the probability of different characters with known password prefixes in the training set, and construct password labels based on the probability values. Assuming it is any possible password prefix, set the ground truth label of the next character to y_i and thus obtain the probability of the next arbitrary character c:

$$P(c|prefix)=\frac{Count(prefix+c)}{Count(prefix)} \quad (13)$$

The PassTCN-PPLL method proposed by Ye et al. [11] effectively improves password coverage by combining time convolutional neural networks and password probability distribution labels. In 2023, Wu et al. [15] also proposed a password guessing model PGTCN improved by feature

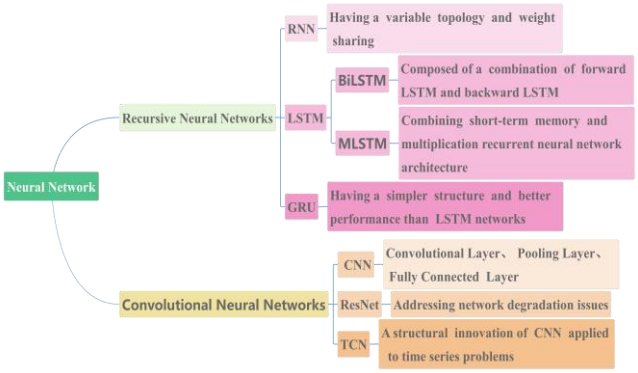

Figure 2. Partial Recursive Neural Networks and Convolutional Neural Networks.

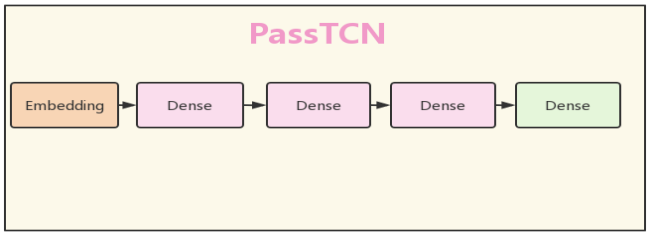

Figure 3. Structure Based on TCN.

fusion technology based on TCN combined with residual learning, with the aim of improving model performance and thus improving guessing efficiency. Relatively speaking, PGTCN is more stable. In order to reduce the probability of repetition, PGTCN also adopts a label like approach. Wu et al. [15] randomly introduce some labels when sampling the next label, rather than always selecting the most likely one. It is worth noting that PGTCN can thoroughly extract high level structures and low-level characteristics, which helps in password generation.

Neural networks are widely used in password guessing, with the main network models being LSTM and TCN. This only introduces the past two years and the original methods.

D. GAN model family

In 2014, Goodfellow et al. [37] proposed Generative Adversarial Nets (GANs). GANs are inspired by the zero-sum game theory, which consists of two parts: a generative model (G) and a discriminant model (D). The generative model captures the data distribution of samples, which is used in password guessing to generate passwords that can deceive the discriminant model. The discriminant model is actually a binary classifier used to distinguish whether the input data is true or false, whether it is real data or generated samples by the generative model. The emergence of GAN has enabled networks to learn more precise losses through adversarial learning, prompting generators to generate higher quality results, greatly promoting the development of this field and entering the vision of more popular.

The optimization objective function of GAN is as follows:

$$\min_{G} \max_{D} V(D,G) \tag{14}$$

$$V(D,G)=E_{x\sim P_{data}(x)}[\log D(x)]+E_{z\sim P_{z}(z)}[\log(1-D(G(z)))] \tag{15}$$

Equation (15) represents the loss function of GAN. Train network G to minimize log(1-D(G(z))), i.e., to maximize the loss of D. Training network D to maximize log D(x) and log(1-D(G(z))). In the G network, log(1-D(G(z))) represents loss. Under the D network, - log D(x) + log(1-D(G(z))) represents loss.

In 2019, Hitaj et al. [9] proposed applying GAN to password guessing and named it PassGAN. PassGAN does not rely on password analysis like Markov models, PCFG, and neural networks, but instead uses GAN to automatically learn the true password distribution from publicly leaked passwords. In other words, we do not need any professional knowledge related to cryptography, and applying GAN can generate high-quality passwords for guessing. Hitaj et al. [9] used Improved training of Wasserstein GANs (IWGAN) [42][43], with the optimizer using ADAM, which IWGAN relies on to minimize training errors.

The PassGAN generator G structure consists of 5 residual blocks, a one-dimensional convolutional layer, and activation functions using Linear and SoftMax; The discriminator D structure includes 5 residual blocks, a one-dimensional convolutional layer, and an activation function using Linear. The positions of G and D convolutional layers and linear activation functions are different. Figure 4 shows the residual module structure, and Figure 5 shows the PassGAN structure.

GAN does not require complex Markov chains to perform well in password guessing, but it has problems with unstable training, vanishing gradients, and mode collapse. Although Hitaj et al. [9] applied IWGAN, the problem still exists. Nam et al. [27] proposed a candidate password for optimizing guessing, named REDPACK using a relativistic GAN method. REDPACK effectively combines multiple generation models to generate passwords. Generator G can effectively optimize candidate password selection by selecting different models, such as OMEN, PCFG, etc. Nam et al. [27] improved the performance of cracking through custom rules, and there is still room for further improvement in the future.

In 2022, Jiang et al. [14] and Yu et al. [10] proposed a password generation model based on ordered Markov enumeration and discriminant networks (OMECDN) for PassGAN and added gradient normalization to PassGAN [10][21][22]. Jiang et al. [14] changed generator G to OMEN and discriminator D to critical discriminant network. OMECDN can sort based on the probability of password combinations, match the true password distribution, and reduce repetition rates. Yu et al. [10] found that the combination of IWGAN and gradient penalty is not an ideal method to solve the shortcomings of GAN, so they added gradient normalization counting to discriminator D and named it GNPassGAN. GNPassGAN guessed 88.03% more passwords than PassGAN, reducing repetition by 31.69%. Zhou et al. [17] proposed a new structure based on PassGAN, which uses LSTM network for generator G and multiple convolutional layers in discriminator D, based on the non-differentiability of discrete data sampling process and the impact on backpropagation. In addition, the biggest contribution is the addition of Gumbel SoftMax, named G-Pass. Dynamically adjust parameters during the training process, balancing sample diversity and sample quality.

Gumbel SoftMax assumes that the vocabulary size is v and $h \in R^{v}$ is the output of the last layer of the generator; P specifies the distribution of categories, Y

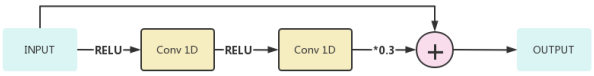

Figure 4. Residual Block's Architecture.

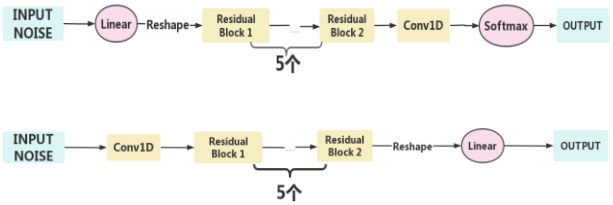

Figure 5. PassGAN's Architecture.Upper generator G, lower discriminator D.

follows a distribution $P(p_1, p_2, \ldots, p_v)$, and p_i is the probability calculated by SoftMax; Using the reparameterization technique to reconstruct random sampling into a deterministic (h_i) and a random element combination(g_i), let F (x) represent a random distribution; We can obtain the inverse function of F (x), from which we can calculate g_i when u~U [0,1]; At this point, one_Hot (argmax (.))is still non-differentiable, using SoftMax as an approximation.

E. Others

This section mainly summarizes the applications of other neural networks and deep learning models proposed in the field of password guessing in recent 2018 and beyond. Table VI at the end of this section displays all the models mentioned in this article.

The various models described in sections A to D rarely consider cross site nature, and the characteristics of different datasets are different because different websites require different password setting rules and target different user groups. In 2018, Liu et al. [7] proposed a universal password guessing model GENPass for cross site nature. Its generator is PCFG+LSTM, and under10^{12}guessing times, the cross-site performance of this model is 20% higher than that of a simple mixed dataset hit rate. In the final analysis, this model is also an application of adversarial thinking, and Xia et al. [48] proposed a similar idea in 2020. In 2022, He et al. [28] proposed a password reuse model called PassTrans based on transformer. The attention mechanism of transformer can be calculated according to the following equation:

$$\text{Attention}(Q,K,V)=\text{softmax}\left(\frac{QK^T}{\sqrt{4iQ_k}}\right) \qquad (16)$$

Q, K and V represent query, key, and value, respectively. Q. K and V calculate the similarity between the current query and all keys and obtain a set of weights by passing this similarity value through the Softmax layer. Q, K and V are both weight matrices.

Sanjay et al. [29] proposed a password generation technique based on a bidirectional generative adversarial network algorithm (BiGAN) using classification and guessing strategy methods, with the aim of generating passwords that improve convergence speed, named PassMon. BiGAN structure: generator, encoder, and discriminator. Pagotta et al. [34] proposed a stream-based generation model for password guessing. The stream-based method was first proposed, providing a representation of the latent space, making it possible to explore specific subspaces and interpolation operations of the latent space, named PassFlow. The PassFlow application has a smaller training set and performs better than PassGAN. The generated password quality is good, and even if it does not match, its rules are very similar to people's password habits. The PassFlow training set is small, in other words, even a subset of the training set, PassFlow, can be effectively used, so it is less affected by the limited number of datasets due to domain specificity. In 2023, Rando et al. [36] proposed a password guessing method based on large language models (LLMs) that can successfully model natural language from a large amount of text without explicit supervision, named PassGPT.

TABLE VI. VARIOUS PASSWORD GUESSING MODELS

Model Name	Basic Generation Model Types	Publication Year
Markov	Markov	2005
PCFG	PCFG	2009
OMEN	Markov	2015
Next Gen PCFG	PCFG	2015
FLA	RNN,LSTM	2016
PassGAN	GAN,IWGAN	2017,2019
GENPass	PCFG,LSTM	2018,2020
SPRNN	BiLSTM	2018
BiLSTM	BiLSTM	2019
REDPACK	PCFG,GAN,etc.	2020
Dynamic Markov	Dynamic Markov	2021
GNPassGAN	GAN	2022
PassTCN-PPLL	TCN	2022
LPG-PCFG	PCFG	2022
G-Pass	GAN	2022
Passtrans	Transformer	2022
OMECDN	Markov,GAN	2022
PassMon	BiGAN	2022
MLSTM	MLSTM	2022
PassFlow	Flow	2021,2022
WordMarkov	Markov	2022
SE#PCFG	PCFG	2023
PassGPT	GPT-2	2023
PassTCN	TCN	2023

IV. DISCUSSION AND FUTURE RESEARCH

Table VI shows that password guessing technology has developed rapidly in the past two years. Although various models have performed well, there are also various defects and deficiencies. Based on the newly proposed optimization methods and model structures in recent years, this chapter proposes several future research directions in the field of password guessing to address limitations and unresolved work:

- Public password data is not easy to find. We can consider data augmentation technology (DA) to obtain more data. Note that the application of DA technology should not aimlessly expand the data, as

obtaining poor data can lead to worse results. We need to clean the obtained data and eliminate bad data. According to research, some users will set their passwords based on the topic of the website [24]. Password guessing often requires a dictionary, and we can use the DA method to obtain as many websites with the same topic as possible. Based on the special password generation strategy of website themes, a dictionary is generated for password guessing.

- Spectral Normalization (SN) can improve the stability of discriminator D in GAN [35], which is also a variant of GAN. The work we are doing not only applies SN to discriminator D, but also adds SN to generator G. Multiple variants of GAN for password guessing may achieve better results. Of course, model training requires the use of optimizers [20][21][22].
- There are already models in the literature other than Markov models, PCFG, GAN, etc. applied to password guessing. We hope that more types of neural networks and deep learning models can be applied to password guessing [2][23][25][26].
- Password rules cannot be ignored, as most literature does not consider password rules, and the rules required by different websites may vary. Considering the combination of password setting rules and the topic dictionary mentioned above, we hope to apply them simultaneously to password guessing.
- We need to detect password leaks, and Honeywords is a type of bait password used to provide feedback on password leaks [39]. As an effective method for detecting whether passwords have been cracked, how to generate Honeywords better has become a research direction. We can consider using the basic models of various password guessing models mentioned in Table 6, such as PCFG, GAN, etc., to generate Honeywords, with the main goal of making it difficult to distinguish between Honeywords and real passwords.

V. CONCLUSION

With the development of technology, other authentication methods have emerged, but passwords were still a widely used authentication method. In this article, we introduced various methods of password guessing, most of which were based on Markov models, PCFG, NN, and GAN. In other words, we could divide the models mentioned in the article into two categories: probability-based models and deep learning-based models. Markov and PCFG were both related to probability, with the difference being that Markov predicted the next character based on the previous character in the password, while PCFG predicted the next character based on the structure of the password (numbers, letters, special characters). PCFG could be regarded as an optimization of Markov methods, but both had the problem of high computational complexity. RNN, GAN and other related models belonged to deep learning models. There were many types of models in this part. For optimization under the same model, basically, the later model performed better than the previous model.

Some experts have also proposed different types of models for application in password guessing. The field of password guessing, as a relatively new research field, has also benefited from the rapid growth of neural networks and deep learning in the past two years. In this article, we mentioned several feasible future research directions and hoped that researchers could pay attention to and find feasible solutions.

REFERENCES

[1] A. Narayanan and V. Shmatikov, “Fast dictionary attacks on passwords using time-space tradeoff,” Computer and Communications Security, pp. 364–372, 2005.

[2] D. Suleiman, A. Awajan, and W. Al Etaiwi, “The Use of Hidden Markov Model in Natural ARABIC Language Processing: a survey,” Procedia Computer Science, vol. 113, pp. 240-247, 2017.

[3] M. Dürmuth, F. Angelstorf, C. Castelluccia, D. Perito, and A. Chaabane, “OMEN: Faster password guessing using an ordered markov enumerator,” Lecture Notes in Computer Science, vol.8978, pp. 119–132, 2015.

[4] M. Weir, S. Aggarwal, B. De Medeiros, and B. Glodek, “Password cracking using probabilistic context-free grammars,” In Proceedings of the 2009 30th IEEE Symposium on Security and Privacy, pp. 391–405, 2009.

[5] W. Melicher, et al., “Fast, lean, and accurate: Modeling password guessability using neural networks,” In Proceedings of the 25th USENIX Security Symposium, pp. 175–191, 2016.

[6] R. Morris and K. Thomson, “Password security: A case history,” In Communications of the ACM, vol. 22, pp. 594–597, 1979.

[7] Y. Liu, et al., “GENPass: A general deep learning model for password guessing with PCFG rules and adversarial generation,” In Proceedings of the 2018 IEEE International Conference on Communications (ICC), pp. 1–6, 2018.

[8] M. Zhang, Q. Zhang, X. Hu, and W. Liu, “A Password Cracking Method Based On Structure Partition and BiLSTM Recurrent Neural Network,” In Proceedings of the Eighth International Conference on Communication and Network Security, pp. 79–83, 2018.

[9] B. Hitaj, P. Gasti, G. Ateniese, and F. Perez-Cruz, “PassGAN: A Deep Learning Approach for Password Guessing,” Applied Cryptography and Network Security, Lecture Notes in Computer Science, vol.11464, pp. 217-237, 2019.

[10] F. Yu and M. Vargas Martin, “GNPassGAN: Improved Generative Adversarial Networks For Trawling Offline Password Guessing,” 2022 IEEE European Symposium on Security and Privacy Workshops, pp. 10-18, 2022.

[11] J. Ye, M. Jin, G. Gong, R. Shen, and H. Lu, “PassTCN-PPLL: A Password Guessing Model Based on Probability Label Learning and Temporal Convolutional Neural Network,” Sensors 2022, vol.22(17), Article Number: 6484, 2022.

[12] M. Hellman, “A cryptanalytic time-memory trade-off,” IEEE Trans, vol.26, pp. 401–406, 1980.

[13] X. Guo, K. Tan, Y. Liu, M. Jin, and H. Lu, “LPG–PCFG: An Improved Probabilistic Context- Free Grammar to Hit Low-Probability Passwords,” Sensors 2022, vol.22(12), Article Number: 4604, 2022.

[14] J. Jiang, A. Zhou, L. Liu, and L. Zhang, “OMECDN: A Password-Generation Model Based on an Ordered Markov Enumerator and Critic Discriminant Network,” Applied Sciences, vol.12(23), Article Number: 12379, 2022.

[15] Y. Wu, X. Wan, X. Guan, T.Ji, and F.Ye, “PGTCN: A Novel Password-Guessing Model Based on Temporal Convolution

Network," Journal of Network and Computer Applications, vol.213, pp. 103592, 2023.

[16] P. Oechslin, "Making a Faster Cryptanalytic Time-Memory Trade-Off," Lecture Notes in Computer Science, vol. 2729, pp. 617-630, 2003.

[17] T. Zhou, H. Wu, H. Lu, P. Xu, and Y. Cheung, "Password Guessing Based on GAN with Gumbel-Softmax," Security and Communication Networks, vol. 2022, Article Number: 5670629, 2022.

[18] X. Guo, Y. Liu, K. Tan, W. Mao, M. Jin, and H. Lu, "Dynamic Markov Model: Password Guessing Using Probability Adjustment Method," Applied Sciences, vol. 11(10), Article Number: 4607, 2021.

[19] J.Chen and J. S. Rosenthal, "Decrypting classical cipher text using Markov chain Monte Carlo," Statistics and Computing, vol. 22, pp. 397-413, 2012.

[20] L. Liu, et al., "On the Variance of the Adaptive Learning Rate and Beyond," International Conference on Learning Representations, arxiv. 1908.03265, 2019.

[21] Z. Chen, V. Badrinarayanan, C. Y. Lee, and A. Rabinovich "GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks," Proceedings of Machine Learning Research, vol.80, pp. 794-803, 2018.

[22] Y. L. Wu, H. H. Shuai, Z. R. Tam, and H. Y. Chiu, "Gradient Normalization for Generative Adversarial Networks," International Conference on Computer Vision, pp. 6353-6362, 2021.

[23] L. Rabiner and B. Juang, "An introduction to Hidden Markov Models," IEEE ASSP Magazine, vol.3, pp. 4-16, 1986.

[24] A. Kanta, I. Coisel, and M. Scanlon, "A Novel Dictionary Generation Methodology for Contextual-Based Password Cracking," IEEE Access, vol. 10, pp. 59178-59188, 2022.

[25] X. Li, J. Thickstun, I. Gulrajani, P. Liang, and T. Hashimoto, "Diffusion-LM Improves Controllable Text Generation," Advances in Neural Information Processing Systems, vol. 35, pp. 4328-4343, 2022.

[26] J. Lovelace, V. Kishore, C. Wan, E. Shekhtman, and K. Q. Weinberger, "Latent Diffusion for Language Generation," arxiv. 2212.09462, 2022.

[27] S. Nam, S. Jeon, and J. Moon, "Generating Optimized Guessing Candidates toward Better Password Cracking from Multi-Dictionaries Using Relativistic GAN," Applied Sciences, vol. 10(20), pp. 1-19, 2020.

[28] X. He, H. Cheng, J. Xie, P. Wang, and K. Liang, "Passtrans: An Improved Password Reuse Model Based on Transformer," 2022 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 3044-3048, 2022.

[29] S. Murmu, H. Kasyap, and S. Tripathy, "PassMon: A Technique for Password Generation and Strength Estimation," Journal of Network and Systems Management, vol. 30(1), Article number: 13, 2022.

[30] M. Siponen, P. Puhakainen, and A. Vance, "Can individuals' neutralization techniques be overcome? A field experiment on password policy," Computers and Security, vol. 88(C), 2020.

[31] R. Veras, C. Collins, and J. Thorpe, "A Large-Scale Analysis of the Semantic Password Model and Linguistic Patterns in Passwords," ACM Transactions on Privacy and Security, vol. 24(3), pp. 1-21, 2021.

[32] D. Wang, P. Wang, D. He, and Y. Tian, "Birthday, name and bifacial-security: understanding passwords of Chinese web users," In Proceedings of the 28th USENIX Conference on Security Symposium, pp. 1537–1554, 2019.

[33] G. Chang, L. Zhao, and W. Chen, "MLSTM:A Password Guessing Method Based on Multiple Sequence Length LSTM," Computer Science, vol. 49(4), pp. 354-361, 2022.

[34] G. Pagnotta, D. Hitaj, F. D. Gaspari, and L. V. Mancini, "PassFlow: Guessing Passwords with Generative Flows," International Conference on Dependable Systems and Networks, pp. 251-262, 2022.

[35] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral Normalization for Generative Adversarial Networks," International Conference on Learning Representations, arxiv. 1802.05957, 2018.

[36] J. Rando, F. Pérez-Cruz, and B. Hitaj, "PassGPT: Password Modeling and (Guided) Generation with Large Language Models," arxiv. 2306.01545, 2023.

[37] I. J. Goodfellow, et al., "Generative adversarial nets," International Conference on Neural Information Processing Systems, vol. 2, pp. 2672‑2680, 2014.

[38] X. Zhang, X. Zhang, J. Hu, and Y. Zhu, "A New Targeted Online Password Guessing Algorithm Based on Old Password," International Conference on Computer Supported Cooperative Work in Design, pp. 1470-1475, 2023.

[39] D. Wang, Y. Zou, Q. Dong, Y. Song, and X. Huang, "How to Attack and Generate Honeywords," 2022 IEEE Symposium on Security and Privacy (SP), pp. 966-983, 2022.

[40] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," Computer Science, arxiv. 1411.1784, 2014:.

[41] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," International Conference on Learning Representations, arXiv. 1511.06434, 2015.

[42] Y. Chen, Q. Gao, and X. Wang, "Inferential Wasserstein Generative Adversarial Networks," Journal of the Royal Statistical Society Series, vol. 84(1), pp. 83-113, 2022.

[43] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," International Conference on Machine Learning, vol. 70, pp. 214-223, 2017.

[44] J. Kaevrestad, F. Eriksson, and M. Nohlberg, "Understanding passwords ‑ a taxonomy of password creation strategies," Information and Computer Security, vol. 27(3), pp. 453-467, 2019.

[45] S. Houshmand, S. Aggarwal, and R. Flood, "Next Gen PCFG Password Cracking," In IEEE Transactions on Information Forensics and Security, vol. 10(8), pp. 1776-1791, 2015.

[46] Y. Wang, W. Qiu, W. Zhang, H. Tian, and S. Li, "SE#PCFG: Semantically Enhanced PCFG for Password Analysis and Cracking," arxiv. 2306.06824, 2023.

[47] J. Xie, H. Cheng, R. Zhu, P. Wang, and K. Liang, "WordMarkov: A New Password Probability Model of Semantics," IEEE International Conference on Acoustics, pp. 3034-3038, 2022.

[48] Z. Xia, P. Yi, Y. Liu, B. Jiang, W. Wang, and T. Zhu, "GENPass: A Multi-Source Deep Learning Model for Password Guessing," In IEEE Transactions on Multimedia, vol. 22(5), pp. 1323-1332, 2020.

[49] H. Li, M. Chen, S. Yan, C. Jia, and Z. Li, "Password Guessing via Neural Language Modeling," In Machine Learning for Cyber Security, Lecture Notes in Computer Science, vol. 11806, pp. 78‑93, 2019.

 ISBN: 978-1-68558-104-6