Sample Selection Search to Predict Elephant Flows in IXP Programmable Networks

Marcus Vinicius Brito da Silva[1,2], André Augusto Pacheco de Carvalho[2], Arthur Selle Jacobs[1], Ricardo José Pfitscher[1], and Lisandro Zambenedetti Granville[1]

[1] Institute of Informatics, Federal University of Rio Grande do Sul
Av. Bento Gonçalves, 9500, Porto Alegre, Brazil
{mvbsilva, asjacobs, rjpfitscher, granville}@inf.ufrgs.br
[2] Federal Institute of Pará, IFPA, Cametá, Brazil
andre.pacheco@ifpa.edu.br

Abstract. Internet eXchange Points (IXPs) are high-performance networks that allow multiple autonomous systems to exchange traffic. As in any network, IXP operators face management challenges to promote better usage of the services provided by the network. Among these, a critical problem lies in the identification of elephant flows, which are characterized by having traffic size and duration significantly higher than other flows. We explore the periodic pattern of IXP network traffic to predict the new flows' size and duration by observing the previous flows' temporal behavior. One of the critical parameters of success for periodicity-based predictions is the sample selection, with the quality and size of samples directly influencing results. In this paper, we present a sample selection strategy, based on the Cuckoo Search Algorithm, to match it with our mechanism. Our approach uses a Sample Selection Module based on *views* updated from an objective function adapted to the IXP network traffic. Thus, we optimize in $\approx 32\%$ the predictions processing time and increased the mechanism accuracy by $\approx 20\%$, using conservative tolerance for the prediction interval, compared to previous approaches.

Keywords: Network Management, SDN, P4, Cuckoo Search Algorithm

1 Introduction

Internet eXchange Points (IXPs) are high-performance networks and perform an essential role in the Internet ecosystem [3], accounting for at least 20% of all traffic exchanged between Internet Autonomous Systems (ASes) [5]. As in any network, IXP operators face daily management challenges to promote better usage of the services provided by the network [1]. Among these, the identification of elephant flows is a critical problem for IXP management, which are characterized by having traffic size and duration significantly higher than other flows (*i.e.*, small/mice flows) [12]. Thus elephant flows tend to deplete the network devices' resources rapidly and can significantly impact smaller flows traffic that share

the same path on the IXP infrastructure, compromising the overall perceived network Quality of Service (QoS) [11].

Most of state-of-the-art approaches for elephant flow identification combine Software-Defined Networking (SDN) [17] with periodicity-based flow behavior prediction. In efforts to rapidly identify and mitigate the effects of elephant flows on IXP networks, recent research has used the SDN and network programmability paradigm [4] as alternatives to traditional monitoring and identification approaches. Proposed solutions such as DevoFlow [7], OpenSample [24], and SDEFIX [15] present mechanisms for identifying elephant flows in IXP networks using sFlow [21] and managing paths (as a reaction to elephant flows) using Openflow [17]. In IDEAFIX [22], we analyzed flow's size and duration for each ingress packet immediately in the data plane, and compared to thresholds for flow classification. Then, in a first improvement of IDEAFIX, we proposed a periodicity-based mechanism [23] to predict flows' behavior and anticipate the elephant flows identification process in the control plane.

Although the state-of-the-art approaches that use the periodicity of flows can detect Elephant flows occurrences, the sample selection process is neglected. For instance, our prediction mechanism [23] is based on a Locally Weighted Regression (LWR) [20, 6] model, and the samples used in the prediction mechanism were selected only based in the flow 5-tuple. That is, the sample set was not optimized for run-time prediction. We argue that this can delay the elephant flows identification and mitigation time. Thus, in this paper, we present a bio-inspired sample selection strategy based on the metaheuristic Cuckoo Search Algorithm (CSA) [25], to improve the predictions in terms of prediction time and accuracy. Our approach uses a Sample Selection Module based on *views* generated/updated from an objective function adapted to the IXP network traffic, as described in Subsection 3.1. In particular, this paper makes the following contributions: (*i*) a sample selection mechanism for IXP networks traffic, optimized by a metaheuristic strategy (based on CSA); (*ii*) a prototype implementation of the proposed sample selection method, coupled with our previous elephant flow prediction mechanism for improved performance; and (*iii*) perform the elephant flows identification and reaction process fast and efficient.

Experimental results show that our sample selection strategy optimizes up to 41% the predictions processing time and, consequently, identifies and mitigates elephant flows faster than the state-of-the-art. In the best and worst cases from an software emulated P4 switch [4], the reaction time was $\approx$ 32 ms and $\approx$ 102 ms, respectively. However, the packet processing time of the software-emulated P4 switch also influences the reaction time (*i.e.*, it would take less time in a hardware switch). In addition, the mechanism accuracy has increased by $\approx$ 20% compared to previous approaches, using conservative tolerance to the prediction interval. That is, the network operator can define a tolerance for the calculated prediction interval over each inference. The prediction interval determines the fluctuation margin for the inferred value. These results represent, at least, 90% of success in elephant flows identification, out of the total number of elephant flows inserted in the network at each test, with approximately 5% of false positives.

The remainder of this paper is organized as follows. In Section 2, we present background on IXP network traffic and the Cuckoo Search Algorithm. Our optimized sample selection strategy for the elephant flow prediction is described presented in Section 3. The evaluation of our proposal and the main results is presented in Section 4. We discuss related work on Section 5. Finally, in Section 6, we present the main conclusions of the study and future work perspectives.

2 Background

In this section, we first discuss the IXP traffic behavior and elephant flow concepts associated with its management challenges to then introduce our previous solution for elephant flow prediction/identification. Next, we briefly review the Cuckoo Search Algorithm (CSA) on which we base our sample selection proposal.

2.1 IXP Traffic Behavior and Elephant Flows

In the Internet traffic, most of the flows have a small size and/or lifetime (*i.e.*, small flows) [1, 26], but there is a small number of flows that accounts for the majority of the traffic volume, also having a longer lifetime; these are the elephant flows [12, 18]. Elephant flows are common in networks, and they may cause performance issues that demand proper management actions from network operators. For example, elephant flows can significantly impact the small flow traffic that occasionally shares the same network data path. Also, elephant flows tend to deplete the network devices' resources rapidly, this also can lead to undesirable delays, queuing, and packet losses [12, 15, 19]. In the case of IXP, elephant flows impact are more critical, because of the traffic amount that IXP networks must deal with. Therefore, the faster elephant flows are identified, the smaller are their effects on network performance [15, 23].

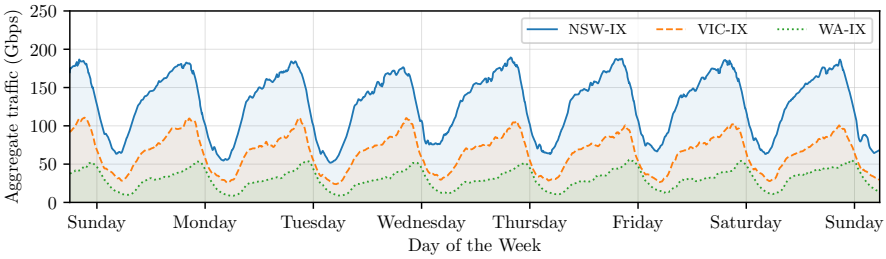


Fig. 1. Australia IXPs aggregate traffic [23].

Figure 1 [23] presents the traffic behavior in Australia IXPs [13] over the course of a week. IX-Australia offers peering in Western Australia (WA-IX),

New South Wales (NSW-IX), and Victoria (VIC-IX). Although this is an aggregated traffic behavior, it is possible to recognize a periodic pattern, with discrete variations. This IXP's behavior is also seen in the largest IXPs worldwide (*e.g.*, AMS-IX, DE-CIX, MSK-IX) [1, 2]. Application-based IXP traffic analysis shows that HTTP/HTTPS is the most dominant application, accounting for more than 50% of bytes exchanged between ASes [1]. Moreover, elephant flows traffic has a substantial, but not perfect, correlation with traffic in total flows [16].

In our previous work [23], we explored the IXP network traffic periodicity to predict the new flow instance size and duration by observing the previous flows' temporal behavior. That is, as the traffic pattern exhibits a periodicity (as shown in Figure 1), our previous prediction mechanism uses the events that previously resulted in elephant flows to predict new elephant flows. Although our prediction mechanism allows us to considerably anticipate elephant flow identification, it is based on a Locally Weighted Regression LWR model and makes predictions at run-time. Therefore, the quantity and quality of the prediction sample base influences results directly. In this paper, we rely on a sample selection strategy for our prediction mechanism (see Section 3), based on the Cuckoo Search Algorithm, described in the next subsection.

2.2 Cuckoo Search Algorithm

Cuckoo Search (CSA) [25] is a metaheuristic algorithm developed to solve optimization problems by simulating the behavior of the cuckoo bird. Biologically, cuckoo birds do not create their nests, and they lay their eggs in other birds' nests at random. In this case, there is a probability that the host bird will find the intruder egg and throw it away. So the eggs that survive are considered the fittest, and so begins the new generation. At the computational scope, for a maximization problem, the quality or adequacy of a solution may be proportional to the value of the objective function. For this, CSA admit the following representations: each egg in a nest represents a solution, and a cuckoo egg represents a new solution, the goal is to use the new and potentially better solutions (cuckoos) to replace a "not so good" solution in the nests.

In this paper, the objective function purpose is to validate elephant flow samples (based on flow's size and duration) obtained from the IXP network and compose an optimized sample set to use in the prediction mechanism, as described in the Section 3. As already mentioned, our prediction mechanism [23] uses the LWR [6] model to predict the behavior of new network flows at run time. Therefore, the quantity and quality of the prediction sample base influences results directly, as presented in the evaluation section (see Section 4).

Based on the Cuckoo Search Algorithm, our Sample Selection Module (refer to Subsection 3.1) creates solution sets from data plane reports, identifies the best solutions by our objective function, and passes them on to new generations. Thus, we seek to optimize the set of samples used in the prediction mechanism in order to improve processing time and accuracy. Therefore, from this, we wish to perform the elephant flows identification and reaction process fast and efficient in run-time.

3 Proposed Approach

In this section, we describe the architecture for elephant flows prediction and identification, and the Sample Select Module, introduced in this paper. Figure 2 shows the proposed architecture, composed of an IXP infrastructure abstraction with a programmable data plane, a historical database, the sample selection module, and a prediction module, in the control plane. When an edge switch receives the flow's first packet, the controller is notified to compute the path by which the flow will be routed in the network. Then, running at the control plane, the mechanism uses the LWR model to predict the flow size and duration [23]. When predicted values are validated, based on tolerated prediction intervals, the mechanism characterizes the flow as an elephant or not, according to thresholds.

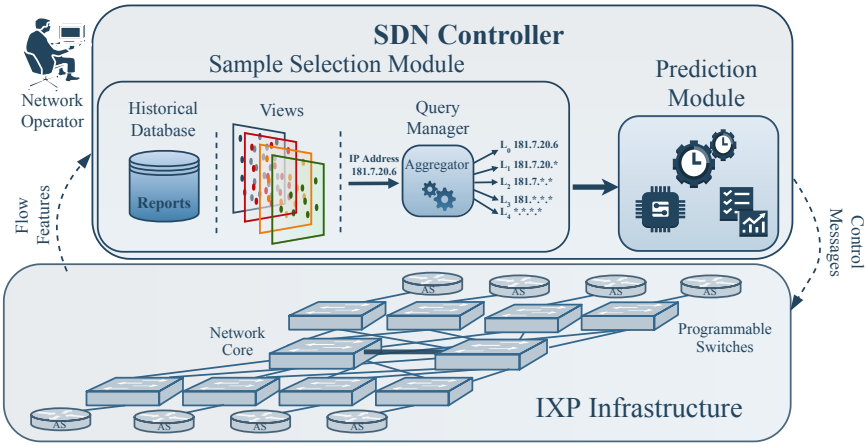


Fig. 2. The Mechanism Architecture.

The prediction mechanism, presented in our previous work [23], relies on a historical database with information about the previous flows' behavior. This information is extracted directly from the data plane, where an agent uses P4 to account, in each edge switch, the volume, and duration of each flow that ingresses in the IXP network. When a flow is finalized (*e.g.*, TCP FIN flag is valid, or timeout exceeded), a notification (on top of UDP) is sent from the data plane to the control plane to report the flow size and duration along with its 5-tuple and start flow time (*i.e.*, *timeStamp*). In our previous work [23], the samples used in the prediction mechanism were selected only based in the flow 5-tuple. That is, the sample set used in the predictions was not optimized for run-time prediction. Consequently, the elephant flows identification and mitigation time are also affected. In contrast, in this paper, we introduced a sample selection strategy based on the Cuckoo Search Algorithm, to minimize the prediction time, improve the prediction mechanism's accuracy, and consequently reduce the reaction time to elephant flows.

3.1 Sample Selection Module

The Sample Selection Module operation is based on a query manager and on views composed by data plane reports stored in the historical database. The views composition/update uses the metaheuristic Cuckoo Search Algorithm (described in the Subsection 2.2) defined by the objective function described below. Each view V_i is associated with a time of day i, resulting in 24 views (0-23 hours). When V_i is active, the mechanism can update the previous view V_{i-1}. Consequently, V_i can only be updated when the next view is active V_{i+1}. Thus, it aims to follow the periodicity in IXP network traffic behavior.

The objective function (Algorithm 1) has the role of updating views' samples from two combined rules: *reduction* and *validation*. In *reduction*, the view's older samples are discarded if they exceed the time window w defined by the network operator (*e.g.*, seven days). In the *validation* rule, the view's current samples and each flow reported by the data plane, in the view time interval, have their volume and duration characteristics compared with the thresholds used in that interval. Therefore, only the samples validated by the objective function will be in the next view (next generation). It allows us to keep up with the latest changes in IXP network behavior, reducing both search space, and sample variation, as evidenced by the results obtained from the experimental evaluation, Section 4. The worst-case complexity of this algorithm is given by $\mathcal{O}(n)$, where n is the number of samples in the *View* plus data plane *reports*, in the *View* interval.

Algorithm 1 Objective Function

Input: *View*, *reports* by the data plane (in the *View* interval), $threshold_{size}$, $threshold_{duration}$
Output: *View* (next generation)
$View.append(reports)$
s is flow sample (*5-tuple, ingressTimeStamp, size, duration*)
w = time window
forall $s \in View$ **do**
  if $s.ingressTimeStamp < w$ ***or*** $s.size < threshold_{size}$ ***or*** $s.duration < threshold_{duration}$ **then**
    $View.remove(s)$
  end
end

Therefore, if a flow was reported by the data plane as an elephant one, it is kept in the views to anticipate the identification of recurring occurrences. However, if the flow is not reported as elephant successive times for the view range, its oldest occurrences should be deleted with the view update process, and the controller understands that this flow is unlikely to be an elephant. Thus, an elephant flow reported in a sporadic situation, in updates (next generations), it will have no match in the analysis view. Such an approach allows the mechanism to be flexible to non-periodic elephant flows occurrences.

The query manager is responsible for verifying whether the analyzed flow's tuple matches on the current view. If there is not a match, the prediction module is not triggered, and the controller understands that this flow is unlikely to be an elephant. Otherwise, when there are matches for this tuple in the current view, the controller realizes that this flow may be an elephant. In this case, the prediction module is triggered to predict the flow's behavior, using samples obtained by the current view, based on the flow's tuple. That is, assuming that the samples on the current view correspond to the more recent flow's occurrences, selected by our objective function, it is expected that the prediction model demands less time to predict the flow's volume and duration, with more accuracy. Finally, if the predicted values exceed the predefined thresholds, then, the flow is classified as an elephant, and control messages are sent to the data plane to mitigate its effects following the IXP network traffic policy.

To prioritize time requirements in an IXP network, in which decisions/actions need to be taken quickly, a proactive approach has been developed. That is, when a flow is started, the controller inserts the routing rules into the default path; also, it adds the routing rules for an alternative path. When an elephant flow is identified, the controller sends control messages to insert a rule into the edge switch to mark the packets of the identified elephant flow (*i.e.*, set an IP header flag) and route them by an alternative path. This approach demands more memory resources for the routing tables. However, this allows reacting to an elephant flow more quickly, as will be shown in the following results.

4 Evaluation

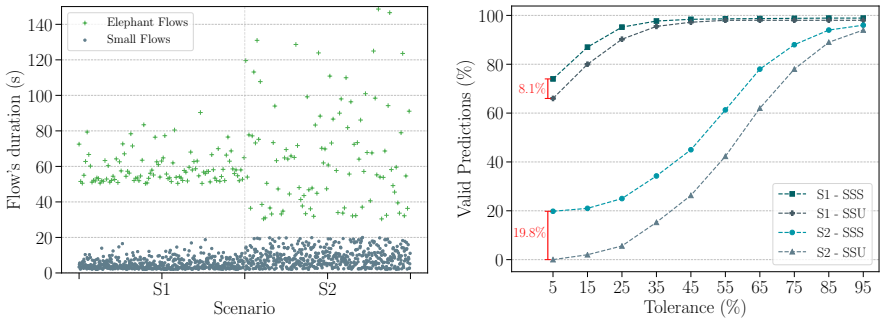
To assess the elephant flow prediction and identification mechanism based on our sample selection search strategy (abbreviation "SSS"), we compared the previous approach [23], without sample selection optimization (abbreviation "SSU"), and focused on evaluating four main aspects: (*i*) mechanism accuracy, *i.e.*, the percentage of valid elephant flows predictions and false positives; (*ii*) reaction time, *i.e.*, the interval between the ingress time of the flow first packet in the IXP network, the query to Sample Selection Module, Prediction Module time, and the identified elephant flow management time; (*iii*) excess data, *i.e.*, the number of bytes that transited in default path until the flow has been identified as an elephant one and a reaction occurs; and (*iv*) resource utilization, *i.e.*, the memory and CPU usage by the Predict and Sample Selection Module in our mechanism; For metrics (*ii*), (*iii*), and (*iv*), lower values are better.

We rely on a topology based on the AMS-IX [2] infrastructure (see Figure 2), as also used in the related work [15, 22], with 8 ASes connected by edge switches to a programmable data plane IXP. Each edge switch runs IDEAFIX to analyze and report flows to the control plane and has at least two IXP network core connection paths. Switches were implemented in the language $P4_{16}$ and by using the software switch BMv2 [3]. The infrastructure was emulated using *Mininet* 2.3, with a bandwidth of 1 Gbps per link and no propagation delay.

[3] https://github.com/p4lang/behavioral-model

We generated a workload in two scenarios with distinct sizes of TCP flows between all connected ASes using *iPerf*. The first scenario (S1) with low traffic behavior variation and regular periodicity, *i.e.*, flows follow a well-defined frequency. Scenario two (S2) with more significant variation in the flows behavior and periodicity. The flow bandwidth was established at 10 Mbps, and the duration was determined through a exponential distribution [14], with a mean of 60 seconds, and the rate parameter ($\lambda = 1/\beta$) with $\beta = 20$ and 40 seconds (S1 and S2, respectively), for elephant flows. For small flows, we used a mean of 5 seconds and $\beta = 5$ and 10 seconds (S1 and S2, respectively) [15, 22]. The IXP network traffic was distributed periodically over nine weeks (*i.e.*, $\approx$ 1,600 hours). The thresholds were defined in 10 MB and 20 seconds [15, 22]. Each experiment lasted 10 minutes, repeated 32 times, and 2,048 flows were generated, of which 12% were elephant flows [1]. We used a computer with Intel Core i7-4790 processor, 16 GB of RAM, and *Ubuntu 16.04 LTS*.

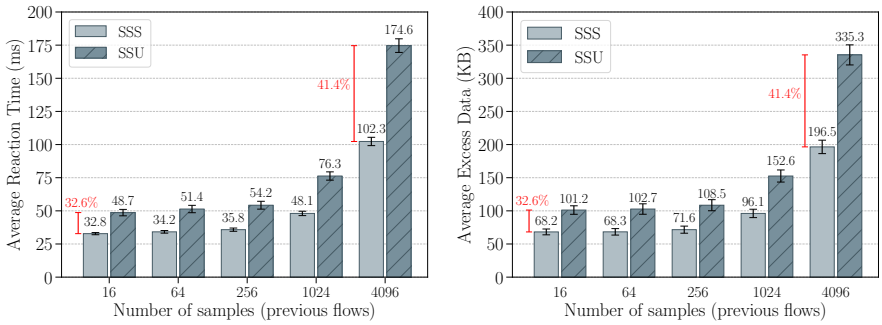
The mechanism accuracy (in Figure 3(b)) was evaluated from the valid predictions (true positives) of the elephant flows (based on the tolerance defined by the network operator for the prediction interval) and false positives, in two scenarios (Figure 3(a)). We observed that the prediction mechanism allows for greater accuracy when using samples from our sample selection search (SSS) strategy, compared to the sample selection unoptimized (SSU) approach. These results are a consequence of the filter applied by our objective function (see subsection 3.1). Results show an increase of $\approx$ 8% and 20% between the two approaches, in both scenarios (S1 and S2, respectively), using conservative tolerance for the prediction interval. It shows that the method can predict and validate values even in non-regular scenarios. However, it requires more flexibility in prediction tolerance. These results represent, at least, 90% of success in elephant flows identification, out of the total number of elephant flows inserted at each test (*i.e.*, $\approx$ 245 flows), with approximately 5% of false positives. Elephant flows not identified by the prediction mechanism were identified directly in the data plane by IDEAFIX [22], after exceeding the thresholds.



(a) Variation of the traffic behavior. (b) Accuracy.

Fig. 3. Scenario and prediction mechanism accuracy.

Figure 4(a) shows the reaction time to elephant flows influenced by the sample number used in the predictions. The amount of samples influences these results because the prediction mechanism needs to process them in run-time. This allows us to observe the time it takes for a reaction to occur with the prediction model being generated/trained at run-time. In the y-axis, the average reaction time (in milliseconds) has a significant difference, when applied the sample selection on the optimized database relative to a search performed on an unoptimized database. Results show a difference up to 32% and 41% in the best (*i.e.*, ≈ 32 ms) and worst (*i.e.*, ≈ 102 ms) case, respectively, at a 95% confidence level.



(a) Reaction time.

(b) Excess data.

Fig. 4. Reaction time and excess data analysis.

The same behavior observed for excess data (Figure 4(b)), *i.e.*, the number of bytes that transited in default path until the flow has been identified as an elephant one and a reaction occurs. This is because the prediction time influences the mitigation process, allowing more packets to be routed through the default path. Results show a difference up to 32% and 41% in the best (*i.e.*, ≈ 68.2 KB) and worst (*i.e.*, ≈ 196.6 KB) case, respectively, at a 95% confidence level.

Finally, Table 1 shows the resource utilization of the mechanism, *i.e.*, the memory and CPU usage by the Predict and Sample Selection Module in our mechanism. We observe that the Sample Selection Module incurs in a significant increase in CPU (*i.e.*, ≈ 45%) and memory (*i.e.*, ≈ 32%) usage in the control plane. It is because the view-based strategy (SSS) needs more processing for its updates and more memory to replicate information compared to SSU.

Table 1. Resource usage of the prediction mechanism.

Approach	SSU	SSS
CPU Used	32.8%	78.7%
Memory Used	15.6%	47.1%

In summary, the results show that we can predict and identify elephant flows faster and more accurately by using a Sample Selection Module. Our view-based strategy updated by an objective function based on the Cuckoo Search Algorithm allows us to choose the best samples from the global data plane reports (*i.e.*, historical database). This improves the predictions' confidence interval and accuracy. In addition, search and prediction times are reduced. Consequently, also reduced the elephant flow identification and reaction time. However, our mechanism incurs a significant increase in CPU and memory using in the control plane, compared to previous approaches. Thereby, the IXP network operator can evaluate which strategy is most appropriate, according to its gains and cost, based on the results presented in this study.

5 Related Work

In DevoFlow [7], OpenSample [24], and SDEFIX [15], an SDN/OpenFlow-based identification module is used to classify elephant flows, analyzing the collected data by sFlow according to predefined rules. When size and duration thresholds are exceeded, the flow is classified as elephant and it is mitigated according to policies written by the network operator. However, these approaches performs elephant flow analysis and identification integrally in the control plane and a reaction occurs only when thresholds are exceeded. In IDEAFIX [22], our previous work, we present the first attempt to identify elephant flows in IXPs faster by relying on programmable data planes. To do that, IDEAFIX takes advantage of P4 [4] features to store and analyze the information about the size and duration of the flows entirely in the data plane. Although IDEAFIX reduces the detection delay when compared to controller-based approaches, it still requires that flows size and duration to reach the thresholds for identification to occur.

Traffic behavior prediction strategies are alternatives to the approaches described above to identify elephant flows. Although it is not perceptible initially, elephant flows are elephant ones since their first packet. Thus if the predicted flow behavior, its characterization may be anticipated. To network flows behavior predict, recurrently strategies use Artificial Neural Networks, Bayesian Networks [9], Hidden Markov [8], and Machine Learning techniques [10]. However, they are limited as to generate run-time prediction models. That is, a training period is required to make predictions whenever changes in network behavior occur. Given this, we introduced a first mechanism for predicting elephant flows in IXP networks in run-time [23] using a Locally Weighted Regression model (LWR) [20, 6]. The sample weights, in the LWR model, are attributed from a Gaussian distribution adjusted by the network operator according to the desired sample time window. In addition, the network operator can define a tolerance range to validate estimations according to the calculated prediction interval for each of them. To make this strategy even more optimized, in this paper, we combine it with a prediction mechanism that relies on the bio-inspired sample selection strategy, based on the Cuckoo Search Algorithm, to perform the elephant flows identification and reaction process fast and efficient in run-time.

6 Conclusions

In this paper, we present a sample selection strategy, based on the metaheuristic Cuckoo Search Algorithm, for predicting elephant flows in Internet eXchange Point programmable networks. We explored the periodicity pattern of IXP network traffic to predict the new flow instance size and duration by observing the previous flows' temporal behavior. In additional we use a Sample Selection Module based on *views* generated/updated from an objective function adapted to the IXP network traffic context. Thus, we optimize in $\approx 32\%$ the predictions processing time and, consequently, identify and mitigate elephant flows faster (*i.e.*, 32.8 ms), and increased the mechanism accuracy by $\approx 20\%$ using conservative tolerance for the prediction interval, compared to previous approaches. As future work, we will consider other methods, based on machine learning to predict flows' behavior, analyze the trade-off between control-data plane processing, and we also plan to deploy our solution in real IXP networks.

Acknowledgement

We thank CNPq for the financial support. This research has been supported by call Universal 01/2016 (CNPq), project NFV-MENTOR process 423275/2016-0.

References

1. Ager, B., Chatzis, N., Feldmann, A., Sarrar, N., Uhlig, S., Willinger, W.: Anatomy of a large European IXP. In: ACM SIGCOMM Computer Communication Review. vol. 42, pp. 163–174. ACM (2012)
2. AMS-IX: Amsterdam Internet Exchange Infrastructure. https://ams-ix.net/technical/ams-ix-infrastructure (2019)
3. Augustin, B., Krishnamurthy, B., Willinger, W.: IXPs: Mapped? In: ACM SIGCOMM Conference on Internet Measurement. pp. 336–349. IMC '09, ACM, NY, USA (2009)
4. Bosshart, P., Daly, D., Gibb, G., Izzard, M., McKeown, N., Rexford, J., Schlesinger, C., Talayco, D., Vahdat, A., Varghese, G., et al.: P4: Programming protocol-independent packet processors. In: ACM SIGCOMM Computer Communication Review. pp. 87–95. ACM (2014)
5. Cardona Restrepo, J.C., Stanojevic, R.: IXP traffic: a macroscopic view. In: The 7th Latin American Networking Conference. pp. 1–8. ACM (2012)
6. Cleveland, W.S., Devlin, S.J.: Locally weighted regression: an approach to regression analysis by local fitting. In: Journal of the American statistical association. vol. 83, pp. 596–610. Taylor & Francis (1988)
7. Curtis, A.R., Mogul, J.C., Tourrilhes, J., Yalagandula, P., Sharma, P., Banerjee, S.: DevoFlow: Scaling flow management for high-performance networks. In: ACM SIGCOMM Computer Communication Review. vol. 41, pp. 254–265. ACM, NY, USA (2011)
8. Dainotti, A., De Donato, W., Pescape, A., Rossi, P.S.: Classification of Network Traffic via Packet-level Hidden Markov Models. In: Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE. pp. 1–5. IEEE (2008)

9. Dalmazo, B.L., Vilela, J.P., Curado, M.: Performance analysis of network traffic predictors in the cloud. Journal of Network and Systems Management **25**(2), 290–320 (Apr 2017)
10. Erman, J., Mahanti, A., Arlitt, M.: Qrp05-4: Internet Traffic Identification using Machine Learning. In: Global Telecommunications Conference, 2006. GLOBECOM'06. IEEE. pp. 1–6. IEEE (2006)
11. Gregori, E., Improta, A., Lenzini, L., Orsini, C.: The impact of IXPs on the AS-level topo-logy structure of the Internet. In: Computer Communications. pp. 68–82. Elsevier (2011)
12. Guo, L., Matta, I.: The war between mice and elephants. In: Ninth International Conference on Network Protocols. pp. 180–188. IEEE (2001)
13. IX Australia: Australia Internet Exchange Point. https://www.ix.asn.au/ (2018)
14. Karagiannis, T., Molle, M., Faloutsos, M., Broido, A.: A nonstationary Poisson view of Internet traffic. In: IEEE Conference on Computer Communications (INFOCOM). vol. 3, pp. 1558–1569. IEEE (2004)
15. Knob, L.A.D., Esteves, R.P., Granville, L.Z., Tarouco, L.M.R.: SDEFIX—Identifying elephant flows in SDN-based IXP networks. In: IEEE/IFIP Network Operations and Management Symposium (NOMS). pp. 19–26. IEEE (2016)
16. Li, Y., Liu, H., Yang, W., Hu, D., Wang, X., Xu, W.: Predicting inter-data-center network traffic using elephant flow and sublink information. In: IEEE Transactions on Network and Service Management. vol. 13, pp. 782–792. IEEE (2016)
17. McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J.: OpenFlow: enabling innovation in campus networks. In: ACM SIGCOMM Computer Communication Review. pp. 69–74. ACM, NY, USA (2008)
18. Mori, T., Kawahara, R., Naito, S., Goto, S.: On the characteristics of Internet Traffic variability: Spikes and Elephants. In: IEICE TRANSACTIONS on Information and Systems. vol. 87, pp. 2644–2653 (2004)
19. Mori, T., Uchida, M., Kawahara, R., Pan, J., Goto, S.: Identifying elephant flows through periodically sampled packets. In: ACM SIGCOMM Conference on Internet Measurement. pp. 115–120. IMC '04, ACM (2004)
20. Schaal, S., Atkeson, C.G.: Robot juggling: implementation of memory-based learning. IEEE Control Systems **14**(1), 57–71 (1994)
21. sFlow: sFlow.org. http://www.sflow.org (2018)
22. da Silva, M.V.B., Jacobs, A.S., Pfitscher, R.J., Granville, L.Z.: IDEAFIX: identifying elephant flows in P4-based IXP networks. In: IEEE Global Communications Conference (GLOBECOM). pp. 1–6. IEEE (2018)
23. da Silva, M.V.B., Jacobs, A.S., Pfitscher, R.J., Granville, L.Z.: Predicting Elephant Flows in Internet Exchange Point Programmable Networks. In: International Conference on Advanced Information Networking and Applications. pp. 485–497. Springer (2019)
24. Suh, J., Kwon, T.T., Dixon, C., Felter, W., Carter, J.: Opensample: A low-latency, sampling-based measurement platform for commodity sdn. In: 34th IEEE International Conference on Distributed Computing Systems (ICDCS). pp. 228–237. IEEE (2014)
25. Yang, X.S., Deb, S.: Cuckoo search via lévy flights. In: 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC). pp. 210–214. IEEE (2009)
26. Zhang, Y., Breslau, L., Paxson, V., Shenker, S.: On the Characteristics and Origins of Internet Flow Rates. In: ACM SIGCOMM Computer Communication Review. vol. 32, pp. 309–322. ACM, New York, NY, USA (Aug 2002)