Detecting Robotic and Compromised IPs in Digital Advertising

Rohit R. R.*
Rakshith V.*
Agniva Som
rohitrrs@amazon.com
urakshi@amazon.com
agnivsom@amazon.com
Amazon Ads
Bangalore, India

ABSTRACT

Irrespective of the intent, malicious or benign, behind the origin of non-human traffic on sponsored advertising pages, failure to detect such unwanted traffic results in deterioration of advertiser performance metrics. Invalid (i.e., robotic) ad traffic is frequently driven by IP addresses (or address ranges) that are exclusively dedicated to VPNs, hosting or proxy services, Tor networks, as well as by unknown or residential IPs that comprise of bot networks set up to inflict maximum damage on a targeted group of advertisers. Sophisticated invalid traffic distributes ad activity across millions of IPs, switches back-and-forth between residential IPs with extremely short-lived dwell time, and disguises behind genuine human traffic to operate from compromised or *mixed* (sending both human and bot traffic) IPs. In order to mitigate rapidly evolving bot IP traffic, we propose an unsupervised model to generate robust IP embeddings from a mixture of autoencoder network experts, which can be segregated by basic heuristics for flagging entirely invalid IPs. Our contribution further includes the development of a new proxy label and a supervised network harnessing IP, search query and product embeddings, for the purpose of detecting mixed IPs sourcing invalid traffic only to specific sponsored search or product listing pages. Our proposed two-component IP detection system enhances suspicious IP traffic detection rate by 25% over a classical supervised model baseline.

KEYWORDS

mixture of experts, autoencoders, weak supervision, neural subnetworks, unsupervised learning, IP reputation, bot IP detection

ACM Reference Format:
Rohit R. R., Rakshith V., and Agniva Som. . Detecting Robotic and Compromised IPs in Digital Advertising. In *3rd Workshop on AI-Enabled Cybersecurity Analytics 2023.* ACM, New York, NY, USA, 6 pages.

1 INTRODUCTION

Sponsored advertising is a favored choice among advertisers to offer their vast product catalogue to the enormous and immensely diverse visitor population on popular e-commerce web pages and apps. Sponsored Ads *aka* Sponsored Search *aka* Promoted Listings refer to digital performance advertising programs that enable advertisers to optimize on their ROIs (return on investment) by bidding on search keywords for prime real estate on these sites so that they can increase their product visibility and sales. However, due to the enormous revenue opportunity (millions to billions of US dollars) generated by these large online marketplaces, sponsored advertising is a fertile ground for fraudsters attempting to artificially inflate their own earnings. We define invalid traffic (IVT) or robotic traffic on sponsored ad pages as ad impressions and clicks that are fraudulent, involuntary or non-human, and are usually generated by automated or coerced methods. Generally, the main purpose to drive IVT is pernicious by nature, where robotic traffic is routed by rogue sellers and advertisers to specifically target their legitimate competition to deplete competitor budget, boost own search rankings etc. However, some IVT may also be benign in motive (nevertheless harmful for advertiser performance), like undeclared crawlers and price grabbers that scrape page content of online marketplaces to gather product information without any direct intent to harm advertisers.

In recent times, sophisticated IVT has evolved to adopt a variety of complex fraud *modus operandi* to avoid detection by the in-house or third party traffic quality (i.e., ad fraud detection) solutions employed by ad networks to prevent IVT. There are many illegitimate online services that claim to boost advertiser ROIs and KPIs (key performance indicators) on sponsored ads, which actually run bot networks and drive IVT in the background in exchange for a fee. Most of these bot operators route IVT through disreputable IP address ranges associated with VPNs, hosting services, data centers, online proxies etc., making invalid IP detection a key business problem for traffic quality. Since IPs get reassigned periodically, and more genuine IP addresses enter into the fray to source suspicious ad traffic, robotic IP lists need to updated dynamically according to rotating IP behavioral patterns. One of the biggest challenges in bot IP detection, however, is the lack of human and non-human/robot labels for training and measuring the efficacy (precision/recall) of models. We show that this problem can be circumvented to a large extent by shifting to modern unsupervised and self-supervised modeling approaches [1, 12, 18], or by training supervised models with weak/partial supervision with incomplete proxy labels [4, 10, 14].

In this paper, we propose autoencoder networks [8] trained on long-term IP behavioral features to generate unsupervised IP vector representations (embeddings) that are subsequently segregated by a simple anomaly detection technique into human and bot IP clusters. In order to remove training bias on under-represented traffic slices and to further enhance detection, we build a Mixture of Experts (MoE) model [6, 9, 16] where each expert is an individual

[*] Contributed equally to the paper

AI-Enabled Cybersecurity Analytics 2023, August 7, 2023, Long Beach, CA
.

autoencoder network. We extend the use of the MoE-autoencoder model IP embeddings to build supervised networks for surgically detecting IVT from IPs with *mixed* (simultaneously sending both human and invalid) traffic. We justify the shortcomings of a fully unsupervised approach to *mixed IP* detection, and develop a high coverage, high quality proxy human label for supervised training. Our work demonstrates that fine-tuning of unsupervised IP embeddings along with pre-trained embeddings of the advertised products and search queries in a single combined network achieves exceptional IVT detection capability, as reinforced by excellent recall metrics on derived bot signals in an online advertising application.

The paper is structured as follows. Section 2 discusses recent related research in autoencoder networks and mixtures of deep neural experts, with a focus on anomaly detection applications. We describe IP embedding generation with autoencoder mixtures in Section 3, and show downstream application for suspicious IP detection both with unsupervised and supervised techniques. Section 4 demonstrates superlative performance on suspicious IP traffic detection by our modeling framework in a sponsored advertising application, while we conclude in Section 5 with future enhancements in IP detection with this framework.

2 RELATED WORK

There has been great research progress made on autoencoder [8, 12, 18] based deep neural networks in a variety of real-world applications that rely on unsupervised learning [15]. Anomaly detection is a common use case with autoencoders, and is usually carried out following one of two methods. The first method takes high reconstruction error as an indicator of anomaly, while the second clusters the network generated latent embeddings to segregate "normal" observations from the anomalies. Sometimes multiple autoencoders are used for anomaly detection by creating ensembles [3, 11], which are trained end-to-end and their reconstruction errors are then combined together to identify data anomalies. Modeling approaches using the reconstruction error for anomaly detection strongly depend on training only using "regular" (i.e., non-anomalous) observations to remain free of training bias creeping in from the anomalies themselves [20].

Application of MoEs [9] on top of autoencoders has been explored by [17] and [19]. [17] uses mixture of experts in a variational autoencoder setup where the experts are combined through a static gating mechanism, with the objective to integrate different modality features efficiently rather than perform anomaly detection. On the other hand, [19] stacks only two encoders and decoders to find anomalies in images using reconstruction error. Another approach to merge output from multiple autoencoders is to have separate autoencoders for different clusters/buckets created within the data [2, 13]. Interestingly, these approaches do not scale well when the number of clusters is too large, and the highly parameterized networks can further overfit to the small volumes of data within each cluster.

3 MODEL OVERVIEW

In this section, we describe our formulation for the mixture of autoencoder experts to encode historical behavioral patterns into unsupervised IP embeddings. IVT from mixed traffic IPs are notoriously hard to isolate, since the IPs themselves show reasonable degree of "regular" behavior from the human portion. We surmise that malicious IVT targets specific product listings and search queries on sponsored advertising programs, and consequently there is a strong interaction between IPs and product/search pages that are targeted. We design another supervised network trained with a new form of proxy labels, which learns from the encoder generated IP embeddings as well as product ID and search query embeddings to discover (IP, query) and (IP, product) tuple traffic pattern anomalies due to the mixed IPs. This two-component modeling framework is able to detect IVT from both fully robotic and mixed traffic IPs.

3.1 Mixture of experts IP encoder model

The IP encoder is an unsupervised model ingesting long-term aggregated statistics of IP traffic to output a single representation for each IP. The objective of this model is to encode historical IP behavioral patterns into a vector using aggregate features, so the autoencoder model [8] makes a sensible choice. Our goal is to comprehensively detect suspicious IPs and IP-focused bots, for which we rely on proxy robotic signals (e.g. IPs with many ad clicks and abnormally low purchase rates) for directional measurement of progress. In a normal autoencoder there is no organic way to encourage greater recall on IP bots. So we modify the autoencoder setup with MoE based training to encode this inductive bias into the model architecture.

With Mixture of Experts (MoE) [9], the input IP features are fed into multiple encoder networks. The autoencoder experts decide on how the IP feature space gets mapped to an embedding representation space. We apply the MoE method to take different encoder outputs and to combine them probabilistically to form a single embedding representation. We apply DSelect-k [7] as the MoE gating function, which picks exactly k of n experts to produce the final embedding. The sparsity in selection of experts encourages better regularization and reduces computational overhead. We perform per-example gating wherein the features of the input sample determine the weights for mixing the experts under consideration.

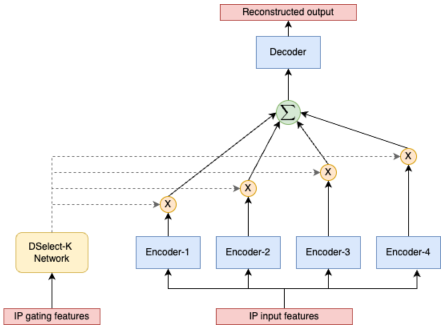

Figure 1: DSelect-K Autoencoder Mixture of Experts architecture

The output from the encoder MoE gate is passed through a decoder network in order to reconstruct the input features, which are then evaluated with a reconstruction loss. Note that MoE is generally applied to the final model output, where output is combined and then passed to a loss function. In our design, we instead apply it on the encoder outputs and have a single decoder. This formulation has two advantages. First, the IP embeddings are used to discover robotic IPs by clustering the computed embeddings and identifying the irregular clusters. If we have separate decoders it is not guaranteed that the encoders will learn representations across different experts in the same vector subspace for clustering. Having a single decoder enforces the encoders to output representations relative to the same decoder. Second, mixing the encoder embeddings to create a single representation avoids the sub-optimal process of manually assigning weights to the chosen encoder representations. Our model architecture is visually presented in Figure 1, while the optimization details are explained in Appendix A.1.

It follows as a natural extension of various real-world autoencoder applications to identify the suspicious IPs as anomalous observations that have high reconstruction errors. However, this approach does not perform well in the IVT detection space as highlighted in Section 4, because sophisticated robotic traffic try to emulate human behavior and often do not manifest as data anomalies in the original or the latent feature space. In order to identify robotic IPs, instead we cluster the IP embeddings using K-Means, and then the robotic clusters are segregated by simple heuristics (like unusually low purchase rate at a cluster level). All IPs from the segregated bot clusters are collected to form the final list of suspicious IPs. Further, the IP embeddings are shared for downstream modeling purposes and consumed in the mixed IP detection problem as described in Section 3.2.

3.2 Mixed IP detection model

With only IP embeddings, it is unrealistic to detect IPs whose behavior appears robotic only in certain component sessions. This can happen when an IP is compromised or has many IP sub-networks, few of which are used by authentic human traffic and others by malicious IVT. Due to its pernicious nature, IVT generally has a specific motive and a clear agenda to hit pre-determined search keywords, ad campaigns or product listings. As a consequence, mixed IP IVT can mostly be segregated as abnormalities on any such (dimension, IP) tuples. Our IP encoder model in Section 3.1 does not require labels, as it generates unsupervised embeddings and utilizes simple heuristics to sequester robotic IPs. However, this approach is inadequate for mixed IP detection, as the combination of (IP, search query) and (IP, product) tuples are combinatorially large and result in an extremely high cardinality joint embedding space. It is nearly impossible to cluster these joint embeddings at scale, or to avoid overfitting to a large number of inadequately small sized clusters. To address this limitation, we resort to supervised learning with proxy labels as a fallback strategy, constructing a new and high fidelity label in the process.

3.2.1 Labels for supervised training. Proxy labeling is the only choice for supervision in this problem, since there is no known process to get the ground truth for identifying bad or mixed IPs. Traditionally, in sponsored advertising applications, any user action (like ad click) that leads to a product purchase on the relevant online marketplace is considered as a strong proxy label of human activity. Our labeling logic is different, which marks all traffic events from users with *decent* purchase rates (i.e., purchase to total traffic ratio) in the past month as human, and the rest as non-human[1]. The proposed labeling scheme has close to 7 times more human label coverage as compared to the traditional label, and aids in avoiding overfitting to sparse labels. Our goal is to generate mixed IP deny-lists to segregate (IP, search query) or (IP, product) tuples, thereby identifying the invalid portion of IP traffic targeting these ad supply slices. To achieve this, we train a supervised model at traffic event (sponsored ad click or impression) level with numerical and categorical ad click/user features. Additionally, the model accepts unsupervised IP embeddings from the IP encoder model described in Section 3.1, and pre-trained embeddings for search queries and product pages from another modeling application (out of scope for discussion in this paper).

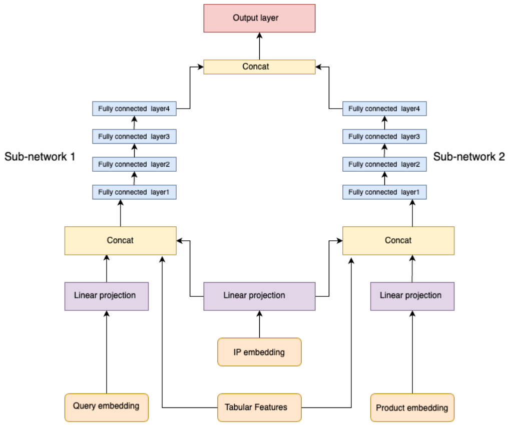

Figure 2: Mixed IP detection model neural architecture

3.2.2 Model architecture. As shown in Figure 2, the proposed architecture is a neural network composed of two neural sub-networks, which are designed to focus on mixed IP bots with varying *modus operandi*. Sub-network 1 and Sub-network 2 learn to identify (IP, query) and (IP, product) targeted IVT attacks respectively. The aim of this joint architecture is to achieve the best of both worlds, which offers a couple of advantages. Firstly, it eliminates the need to build separate models for individual problems, reducing redundant production maintenance overhead. Secondly, by sharing some layers, the network can focus on common aspects of the two IVT detection problems, improving accuracy and reducing overfitting. Figure 2 illustrates how the autoencoder IP model embeddings and pre-trained query/product embeddings are passed to linear projection layers, concatenated with one-hot encoded categorical features, and passed through four fully-connected layers to jointly optimize on binary cross-entropy loss. Our model generates pRobot (model) scores between 0 and 1 for all IP traffic events, where 1 indicates a strong robotic event prediction. pRobot scores are then aggregated across (IP, query) and (IP, product) tuples to generate mixed IP deny-lists, adding any tuple as robotic when average pRobot score is over a certain threshold.

[1] one-sided weak labeling where non-human indicates a mix of true valid and true invalid; human label represents (almost) entirely true valid, aside from trivial mistakes

4 EXPERIMENT RESULTS

In this section, we demonstrate quantitatively the usefulness of the proposed two-component IP detection system in a real-world application of suspicious IP traffic detection. All metrics for the detection system have been reported for the sponsored advertising program of a major online marketplace in the US region. We carried out extensive experiments with various modeling and hyperparameter options for each model, and listed results for the best model/parameter choices.

We compare between the mixture of experts (MoE-AE), vanilla, sparse [12] and denoising [18] autoencoders (AE) with respect to a baseline Random Forest model. All the robotic IP detection models (supervised or unsupervised) have the same input feature set. Each training observation is a single IP with long-term (4 months) aggregated ad traffic behavioral patterns to create a total of 33 IP features. We calibrate each of the models to attain a fixed target value for a proxy measure of false positive rate (FPR). Model architectural and hyperparameter choice details are relegated to Appendix A.2.

4.1 Model results

We evaluate the results of our experiments on the basis of important IVT detection metrics. These metrics include a) detection rate (DR) depicting how much of invalid traffic an algorithm is catching, (b) a proxy measure of false positive rate (FPR) that measures algorithm precision in catching robotic traffic, and (c) a set of very confident robotic coverage (RC) signals derived from simple, interpretable heuristics according to domain expert knowledge. Improving on RC signal coverage gives a directionally positive indicator of increasing bot IP detection (unknown) *true* recall metric.

Model variant	Click DR	Impression DR
Vanilla-AE	-0.85%	45.90%
Sparse-AE	-0.37%	45.73%
Denoising-AE	-0.44%	44.98%
MoE-AE	**-0.17%**	**45.97%**

Table 1: Percentage improvement in detection rates over baseline model for different IP encoder models

Table 1 shows relative improvement in bot IP detection rate of the different autoencoder models over a baseline Random Forest model trained with the traditional click-attributed purchase based *short-term labels* called out in Section 3.2.1, given that each model is calibrated at the same proxy FPR. The results indicate that the MoE-AE has the best performance, having similar detection as Random Forest in terms of click bot IP detection and vastly outperforming in terms of crawler IP detection. One important point to mention here is that the unsupervised IP models can be simultaneously applied for both click bot IP and crawler (impression bot) IP detection by the downstream tasks, whereas the Random Forest model is limited to only click bot IP detection due to lack of good training labels in the impression data. Moreover, as briefly touched upon in Section 3.1, reconstruction error based bot IP detection based on anomalous latent space embeddings from the autoencoder models performs rather poorly, resulting in a drop of 86% in Vanilla-AE (reconstruction) click DR relative to the baseline model. This observation justifies the need for the clustering step, where groups of *similar* IPs with human-like, regular IP features can get correctly segregated as IVT.

Model variant	Click DR
Mixed IP: Proposed	**$1.051x\%$**
Mixed IP: ml-BERT query embeddings	$1.027x\%$
Mixed IP: short-term labels	$1.015x\%$
Mixed IP: one-hot IP embeddings	$x\%$

Table 2: Detection metrics for different mixed IP model variants w.r.t. one of the models as baseline (data obfuscated for confidentiality)

We performed ablation studies on the importance of different components of the mixed IP detection network. We used search query and product embeddings developed by our organization in the proposed model, and also experimented with multi-lingual BERT embedding (base model) [5] as an alternative source of query embeddings. We observe that our pre-trained query embeddings trained and optimized on online marketplace data performs marginally better than multi-lingual BERT as shown in Table 2. Experiments on the usefulness of IP embeddings from the IP encoder model as opposed to training IP embeddings from scratch (consider each IP as a one-hot categorical feature) also show that there are relative improvements with pre-trained embeddings. Our choice of new labels with high coverage, which we refer to as *long-term labels*, for supervised training is also justified by the higher detection rate at same model FPR. The final model is thus built on *long-term labels*, marketplace specific pre-trained query and product embeddings, and the IP embeddings from the MoE-autoencoder model.

Metrics	MoE-AE + mixed IP	Vanilla AE + mixed IP
Click DR	**25.19%**	24.59%
Impression DR	**50.75%**	50.67%
RC1	**17.92%**	17.60%
RC2	**11.81%**	11.19%
RC3	18.67%	**18.82%**
RC4	**24.56%**	24.56%
RC5	79.07%	**79.36%**
RC6	**151.17%**	150.53%

Table 3: Detection rate and robotic recall metrics percentage improvement over the baseline model

For the final two-component IP detection system, we compare systems with IP embeddings generated by any specific autoencoder model and then the same embeddings fed to the mixed IP detection model, for various model choices. For brevity, we only present the relative improvement shown by our choice, the MoE-autoencoder model system versus the vanilla autoencoder model system, as compared to the baseline Random Forest IP detection model. The detection rate (at the same FPR) for both systems are very close (MoE-AE is only slightly higher), whereas the robot signal coverage metrics show that the MoE-AE system is marginally superior with higher Gains on more signals. Among the robot coverage signals (RC), RC1-RC4 are click bot IP detection signals while RC5-RC6 are crawler bot IP detection signals (refer to Appendix A.3 for specifics). We observe that the most significant IP detection metric improvements are visible on the crawler signals.

5 CONCLUSION AND FUTURE WORK

Invalid ad traffic is routinely directed through suspicious IP ranges that are either entirely or partly taken over by bot operators for perpetrating fraud. In this paper, we motivate the need for an advanced IP detection model with the combined objective of mitigating entire IP traffic if needed or surgically removing specific traffic slices within an IP when the IP has mixed behavior. Our proposed Mixture of Experts autoencoder model is able to fulfill this dual role with overall IVT detection (on clicks and impressions) and robot signal coverage far exceeding the performance of the supervised model. Unsupervised IP embeddings from the MoE-AE model form a crucial input component to a new mixed IP detection model that also learns from pre-trained query and product ID embeddings to successfully isolate robotic components within compromised or shared IPs, as part of our proposed two-component IP detection system. In the future, by adding a multitude of IP metadata features such as ISP, ASN, IP domain, IP location, organization description etc. spanning over many feature classes (numerical, tabular, natural language, embeddings) to the IP encoder, we wish to make even bigger step-function advancements in robotic and mixed IP detection.

REFERENCES

[1] Agarwal, R., Muralidhar, A., Som, A., and Kowshik, H. Self-supervised representation learning across sequential and tabular features using transformers. In *NeurIPS 2022 First Table Representation Workshop* (2022).

[2] Chazan, S. E., Gannot, S., and Goldberger, J. Deep clustering based on a mixture of autoencoders. In *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)* (2019), IEEE, pp. 1–6.

[3] Chen, J., Sathe, S., Aggarwal, C., and Turaga, D. Outlier detection with autoencoder ensembles. In *Proceedings of the 2017 SIAM international conference on data mining* (2017), SIAM, pp. 90–98.

[4] Chitlangia, S., Muralidhar, A., and Agarwal, R. Self supervised pre-training for large scale tabular data. In *NeurIPS 2022 First Table Representation Workshop* (2022).

[5] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[6] Eigen, D., Ranzato, M., and Sutskever, I. Learning factored representations in a deep mixture of experts. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings* (2014), Y. Bengio and Y. LeCun, Eds.

[7] Hazimeh, H., Zhao, Z., Chowdhery, A., Sathiamoorthy, M., Chen, Y., Mazumder, R., Hong, L., and Chi, E. Dselect-k: Differentiable selection in the mixture of experts with applications to multi-task learning. *Advances in Neural Information Processing Systems 34* (2021), 29335–29347.

[8] Hinton, G. E., and Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *Science 313*, 5786 (2006), 504–507.

[9] Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. Adaptive mixtures of local experts. *Neural computation 3*, 1 (1991), 79–87.

[10] Karamanolakis, G., Mukherjee, S. S., Zheng, G., and Awadallah, A. H. Self-training with weak supervision. In *NAACL 2021* (May 2021), NAACL 2021.

[11] Kieu, T., Yang, B., Guo, C., and Jensen, C. S. Outlier detection for time series with recurrent autoencoder ensembles. In *IJCAI* (2019), pp. 2725–2732.

[12] Ng, A., et al. Sparse autoencoder. *CS294A Lecture notes 72*, 2011 (2011), 1–19.

[13] Opochinsky, Y., Chazan, S. E., Gannot, S., and Goldberger, J. K-autoencoders deep clustering. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2020), pp. 4037–4041.

[14] Ratner, A., Bach, S. H., Ehrenberg, H. R., Fries, J. A., Wu, S., and Ré, C. Snorkel: Rapid training data creation with weak supervision. *Proc. VLDB Endow. 11*, 3 (2017), 269–282.

[15] Ruff, L., Kauffmann, J. R., Vandermeulen, R. A., Montavon, G., Samek, W., Kloft, M., Dietterich, T. G., and Müller, K.-R. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE 109*, 5 (2021), 756–795.

[16] Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q. V., Hinton, G. E., and Dean, J. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings* (2017), OpenReview.net.

[17] Shi, Y., Paige, B., Torr, P., et al. Variational mixture-of-experts autoencoders for multi-modal deep generative models. *Advances in Neural Information Processing Systems 32* (2019).

[18] Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning* (2008), pp. 1096–1103.

[19] Yu, Q., Kavitha, M. S., and Kurita, T. Mixture of experts with convolutional and variational autoencoders for anomaly detection. *Applied Intelligence 51* (2021), 3241–3254.

[20] Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., and Chen, H. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations* (2018).

A APPENDIX

A.1 Optimization equations for IP encoder

Let the encoder input feature space be $\mathcal{X}$ and the gating feature space be $\mathcal{Z}$. Let's denote the set of training samples of size N by $\mathcal{D} = \{(x_i, z_i) \in \mathcal{X} \times \mathcal{Z}\}_{i=1}^{N}$. Assume that we have n encoder models with the encoder model i represented by the function $E_i : \mathcal{X} \rightarrow \mathbb{R}^d$, whereas the decoder model is represented by the function $D : \mathbb{R}^d \rightarrow \mathcal{X}$ and the gating function is represented by $G : \mathcal{Z} \rightarrow \mathbb{R}^n$. We train using the reconstruction loss function $\ell : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$. Out of the n experts we would like to select k experts to contribute to the final encoded representation. The encoder, decoder and gating functions are neural networks parameterized by different weights.

Through our MoE setup we want to solve the following optimization problem,

$$\begin{aligned} &\min_{E_1,\ldots E_n,D,G} \ \frac{1}{N} \sum_{(x,z)\in\mathcal{D}} \ell\left(x, D\left(\sum_{i=1}^{n} E_i(x)G(z)_i\right)\right) \\ &\text{s.t. } \forall j \in [N], \quad \|G(z_j)\|_0 \leq k, \ \sum_{i=1}^{n} G(z_j)_i = 1, \ G(z_j) \geq 0 \end{aligned} \tag{1}$$

where $\|w\|_0$ is the L_0 norm of a real vector w, and the gating function G is implemented as a DSelect-k layer here. Equation 1 is a constrained optimization problem and using DSelect-k, a continuously differentiable gating network, allows us to approximately solve for it by optimizing the entire network with gradient descent. Once the network is trained, the IP embedding of an input sample (x, z) is given by the output of the gated encoder models,

$$IP_{embedding}(x, z) = \sum_{i=1}^{n} E_i(x)G(z)_i$$

A.2 Network training details

IP encoder model: All the encoders in the vanilla, MoE, denoising and sparse autoencoders were set to be feed forward networks with 2 hidden ReLU activated layers with dimensions $[48, 24, 10]$ for the MoE, denoising and vanilla autoencoders and $[64, 48, 20]$ for the sparse autoencoders. The decoders are mirror networks of the encoders. MoE-AE has 4 encoders along with a DSelect-k layer with $k = 1$. All autoencoders are optimized on MAE loss with Adam optimizer. A total of five gating features related to counts and ratios of ad clicks and ad conversions have been used in the MoE experiments.

Mixed IP detection model: As described in Section 3.2 (and Figure 2), each sub-network in the mixed IP detection model has four fully-connected layers consisting of 2048, 1024, 512 and 256 nodes respectively with ReLU activation function and L2 regularization. The model is trained and calibrated on 1 week of Sponsored Ads clicks data.

A.3 Robot coverage heuristic signals

In order to calculate recall on known bot traffic, we use the following set of very confident heuristics in the form of robot coverage (RC) signals.

- **RC1** : Sessions with more than 20 times the average number of ad clicks in a single hour
- **RC2** : User Agents (UA) with 70 times lower than average purchase rates
- **RC3** : (IP, Query)-tuples with 70 times lower than average purchase rates
- **RC4** : (IP, Product ID)-tuples with 70 times lower than average purchase rates
- **RC5** : User Agents with 40 times lower than average click-through-rates and 35 times lower than average purchase rates
- **RC6** : Sessions with 80 times higher than average ad impressions in a single hour and 4 times lower than average click-through-rates