AISec2024 Paper #48 Reviews and Comments
===
Paper #48 AI2DS: Advanced Deep Autoencoder-Driven Method for Intelligent
Network Intrusion Detection Systems

Review #48A
===

Overall merit
----
1. Reject

Reviewer expertise
----
4. Expert

Paper summary

The paper tackles the problem of Network Intrusion Detection (NID) by means of machine learning (ML). Specifically, the paper proposes a deep learning (DL) autoencoder that seeks to improve existing NID methods by means of unsupervised learning: the autoencoder is trained only on "normal" samples, and the proposed method automatically adjusts the distance threshold through which any given sample is determined to be "normal" or "anomalous" -- the latter ideally denoting an attack. The proposed method, named AI2DS, is tested on a publicly available dataset: NSL-KDD, and allegedly outperforms existing methods.

Strenghts:

+ A universal solution for NID has yet to be found...

Weaknesses:

- ...but the research described in this paper is _not_ a step in the right direction
- No source code disclosure
- Outdated and flawed dataset
- Inappropriate baselines
- Poor writing quality
- Unclear how the "intelligent thresholding" technique is implemented

Comments for authors

This paper cannot be accept to AISEC. I am certain that the authors carried out extensive experiments and that put a lot of effort behind this submission. However, the paper commits various "pitfalls" that are well-known in this domain, and which denote a superficial treatment/understanding of prior work. As a consequence, the main takeaways are trivial and the results of this paper cannot be used as a foundation to advance the state of the art in the NID context.

I will provide abundant references below, pointing to works that I strongly recommend the authors to carefully read if they wish to pursue impactful research in this domain. Regardless, I will also elaborate on the above mentioned weaknesses, as well as provide avenues that the authors can explore in the NID context.

Outdated and flawed dataset

The paper is evaluated on the NSL-KDD dataset, which has long been known for being inherently flawed, as well as not representative of modern network environments (it was collected over 25 years ago!). See [H, Q]. There are many more datasets available today, e.g., UNSW-NB15, CICIDS17, CTU13 (to name a few). See [M]. However, be aware that also these datasets have limitations (see [J,K,T]). As such, the best way to carry out a convincing evaluation is to use these datasets only as a "preliminary" use case: after using them to see if the proposed method "works", the best thing the authors can do is carry out an evaluation in the real world, and use real data to validate whether the method "reall works". Even an assessment on a small network environment would provide much more convincing results than a 99.9% accuracy on a benchmark dataset.

Inappropriate baselines

The paper compares with [10, 23, 29], but there are no criteria mentioned as to why these three works have been chosen. Unfortunately, there are many more works that propose autoencoders for NID (the most notable one is the work by Mirsky et al.[O]; see also [R]). Nevertheless, it has also been shown that DL techniques in general are much less effective than "traditional" ML methods (e.g., random forests), the latter also providing advantages in terms of computational runtime and explainability.

Poor writing quality

The paper is, in general, overstating a lot of claims. The impression is that the text is inflated with terms whose sole purpose is to "(over)sell" certain procedures. For instance, there is an abundant usage of the term "comprehensive" without any supporting argument. This is annoying to read.

I recommend changing the writing style and toning down the claims---especially if they cannot be backed-up with solid facts.

Finally, consider adding a "limitations" section.

Avenues for future work

The following are areas (and corresponding papers) that can inspire the "future work" mentioned at the end of this paper, as well as new ones.

Consider adding evaluations against "attackers that try to evade the proposed method" (e.g., adversarial ML attacks). See [E]

Consider adding evaluations on the "explainability" of the proposed method. See [A, B]

Consider adding evaluations on the "hardware" perspective. See [C]

Consider adding statististical tests to ascertain whether a method is truly superior than another one (which may take into account diverse metrics). See [C].

Consider adding evaluations on the "concept drift" effect. See [I, S]

Consider adding evaluations on the "preprocessing" of the network traffic (i.e., the conversion from packets to network flows). See [N]

Consider adding evaluations on the "amount of training data" (and quality of labeling) required to develop the proposed model. See [F, G]

Consider adding evaluations on the "transferability" of the proposed technique. See [M,L,N]

Consider looking into the "dos and donts" of ML in computer security evaluations. See [D].

Consider adding evaluations on the "false positives". See [O].

Some additional issues:

* In Section 3.7, the text states ```The experimentswere conducted meticulously on a high-performance computing system equipped with an Intel Core i5 7200U dual-core CPU and 32GB of RAM``. I recommend avoid using subjective adjectives such as "meticulously" to describe the experiments, and instead describe the experiments in a "meticulous" way. Neertheless, an Intel Core i5 is not a "high-performance computing system". It was released over 7 years ago, and it was not "high-performance" even at the time of its rollout. On what basis is this statement written?
* The paper is 25MB in size, and I presume this is due to the figures being very "heavy" -- potentially due to high dpi/resolution settings. My recommendation to avoid this problem is to use vectorized figures (e.g., svg format).
* In any case, release the source code.
* I found no specific assessment of the proposed "intelligent thresholding" mechanism.

EXTERNAL REFERENCES

[A]: Jacobs, Arthur S., et al. "Ai/ml for network security: The emperor has no clothes." Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. 2022.

[B]: Nadeem, Azqa, et al. "Sok: Explainable machine learning for computer security applications." arXiv preprint arXiv:2208.10605 (2022) [EuroS&P23].

[C]: Apruzzese, Giovanni, Pavel Laskov, and Johannes Schneider. "SoK: Pragmatic Assessment of Machine Learning for Network Intrusion Detection." EuroS&P (2023)

[D]: Arp, Daniel, et al. "Dos and don'ts of machine learning in computer security." 31st USENIX Security Symposium (USENIX Security 22). 2022.

[E]: Apruzzese, Giovanni, et al. "Modeling realistic adversarial attacks against network intrusion detection systems." Digital Threats: Research and Practice (DTRAP) 3.3 (2022): 1-19.

[F]: Van Ede, Thijs, et al. "Deepcase: Semi-supervised contextual analysis of security events." 2022 IEEE Symposium on Security and Privacy (SP). IEEE, 2022.

[G]: Apruzzese, Giovanni, Pavel Laskov, and Aliya Tastemirova. "SoK: The impact of unlabelled data in cyberthreat detection." 2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P). IEEE, 2022.

[H]: Apruzzese, Giovanni, et al. "The role of machine learning in cybersecurity." Digital Threats: Research and Practice 4.1 (2023): 1-38.

[I]: Andresini, Giuseppina, et al. "Insomnia: Towards concept-drift robustness in network intrusion detection." Proceedings of the 14th ACM workshop on artificial intelligence and security. 2021.

[J]: Engelen, Gints, Vera Rimmer, and Wouter Joosen. "Troubleshooting an intrusion detection dataset: the CICIDS2017 case study." 2021 IEEE Security and Privacy Workshops (SPW). IEEE, 2021.

[K]: Liu, Lisa, et al. "Error prevalence in nids datasets: A case study on cic-ids-2017 and cse-cic-ids-2018." 2022 IEEE Conference on Communications and Network Security (CNS). IEEE, 2022.

[L]: Sarhan, Mohanad, Siamak Layeghy, and Marius Portmann. "Evaluating standard feature sets towards increased generalisability and explainability of ML-based network intrusion detection." Big Data Research 30 (2022): 100359.

[M]: Sarhan, Mohanad, et al. "Netflow datasets for machine learning-based network intrusion detection systems." Big Data Technologies and Applications: 10th EAI International Conference, BDTA 2020, and 13th EAI International Conference on Wireless Internet, WiCON 2020, Virtual Event, December 11, 2020, Proceedings 10. Springer International Publishing, 2021.

[N]: Apruzzese, Giovanni, Luca Pajola, and Mauro Conti. "The cross-evaluation of machine learning-based network intrusion detection systems." IEEE Transactions on Network and Service Management 19.4 (2022): 5152-5169.

[O]: Mirsky, Yisroel, et al. "Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection." NDSS18

[P]: Alahmadi, Bushra A., Louise Axon, and Ivan Martinovic. "99% false positives: A qualitative study of {SOC} analysts' perspectives on security alarms." 31st USENIX Security Symposium (USENIX Security 22). 2022.

[Q]: Catillo, Marta, Antonio Pecchia, and Umberto Villano. "Machine Learning on Public Intrusion Datasets: Academic Hype or Concrete Advances in NIDS?." 2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S). IEEE, 2023.

[R]: Catillo, Marta, Antonio Pecchia, and Umberto Villano. "Simpler is better: On the use of autoencoders for intrusion detection." International Conference on the Quality of Information and Communications Technology. Cham: Springer International Publishing, 2022.

[S]: Yang, Limin, et al. "{CADE}: Detecting and explaining concept drift samples for security applications." 30th USENIX Security Symposium (USENIX Security 21). 2021.

[T]: Robert Flood et al "Bad Design Smells in Benchmark NIDS Datasets", IEEE EuroS&P 2024

Review #48B
==========

Overall merit
---
1. Reject

Reviewer expertise
---
3. Knowledgeable

Paper summary

The paper proposed the use of autoencoders for network intrusion detection. The proposed autoencoder is trained over normal traffic to minimize the reconstruction error. A threshold is set, then at test time, if an anomaly is present in the network features, the recontruction error will surpass the threshold and an anomaly is triggered. The proposed method is validated over the NLS-KDD dataset.

Comments for authors

Thanks for submitting your work to the AISec workshop. In my opinion the paper falls below the bar for acceptance to the workshop. The issue of the paper is the novelty of the proposed approach. The proposed method is not novel and the use of Autoencoders for anomaly detection has been proposed in the past, e.g., [1]. Also one of the methods that the paper is comparing against was using the same technique for anomaly detection [23].

The paper has the following limitations:
 - The paper's positioning fails to report the real state of the art of NIDS. Important references like [1] are missing.
 - The paper does not explain why the proposed approach would differ from well-established autoencoder-based NIDS. The claimed key contribution "adaptive thresholding" is not new. Prior work already discussed how to automatically compute the threshold based on the distribution of autoencoder residuals e.g., [2].
 - Parameter tuning is not described nor validated, the choice of the threshold, and learning rate batch size requires validation.
 - The false positive rate is non-negligible at around 0.15 (I derived it from the confusion matrix in Fig 5, $\frac{1415}{1415+8296}$), the proposed method would cause a high number of false alarms. This aspect is not discussed in the paper.
 - Figure 6 looks strange to me. The rows do not sum up to 1. How did you nromalize the confusion matrix? Starting from figure 5 the normalized confusion matrix is array([[0.85428895, 0.14571105], [0.0436375 , 0.9563625]]) (using sklearn confusion_matrix(y_true,y_pred, normalize='true')).
 - Writing style overemphasizes the novelty of the approach. The writing is to a certain extent exaggerating the novelty of what in practice is being proposed. I suggest the authors to change the tone of writing.
- Algorithm 1 looks like it was taken verbatim for prior work [23]. This is not an acceptable practice.

Ovarall my suggestion to imporve the paper for later submission is to restructure the paper to really make it clear to the reader what is new in the proposed approach, especially compared with [1], not only at an experimental level but also at a conceptual level. Which problem is your solution solving, compared to prior work? Comparing with the state-of-the-art in the field is essential to assess the novelty of a newly proposed approach, especially if it relies on the same idea (in this case autoencoders). I would also suggest to carefully consider the FPR when proposing a new anomaly detection method.

[1] Mirsky, Yisroel, et al. "Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection." Network and Distributed Systems Security (NDSS) Symposium. 2018.

[2] Taormina, Riccardo, and Stefano Galelli. "Deep-learning approach to the detection and localization of cyber-physical attacks on water distribution systems." Journal of Water Resources Planning and Management 144.10 (2018): 04018065.

[23] Z. Tauscher, Y. Jiang, K. Zhang, J. Wang and H. Song, "Learning to Detect: A Data-driven Approach for Network Intrusion Detection," 2021 IEEE International Performance, Computing, and Communications Conference (IPCCC), Austin, TX, USA, 2021, pp. 1-6, doi: 10.1109/IPCCC51483.2021.9679415.

Review #48C
===

Overall merit
----
1. Reject

Reviewer expertise
----
4. Expert

Paper summary

The authors propose using a deep autoencoder to implement a network intrusion detection system. The autoencoder is trained to reconstruct exclusively normal (i.e., benign) network traffic. After training, the reconstruction error of the autoencoder is used to estimate whether the unknown traffic is benign or malicious. The idea is that the autoencoder, trained on benign data, will have lower reconstruction error on normal traffic vs anomalous traffic. The authors evaluate their approach on the NSL-KDD dataset.

Comments for authors

The paper is well-written and well-presented. Overall, it is easy to follow. Performing classification by training only with benign data is an interesting research direction, that i believe should receive more widespread analysis. However, the the idea of utilizing autoencoders and reconstruction error has been proposed many times in the literature, also in the context of intrusion detection. Moreover, the experimental analysis is limited and does not do a good job of thoroughly evaluating the proposal. Below I discuss these points in more detail.

- Lack of novelty: the core idea of the paper is to use the autoencoder's reconstruction error to estimate whether traffic is normal or anomalous. This approach has been widely applied in the general area of intrusion detection, and also specifically to network intrusion detection. The contribution and novelty of the paper are, therefore, very limited.
- Poor RW analysis: the RW analysis is missing important papers on network intrusion detection that rely on autoencoders:
    - Mirsky, Yisroel, et al. "Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection." Network and Distributed Systems Security (NDSS) Symposium. 2018. This is probably the most well-known paper to utilize the autoencoder's reconstruction error to detect anomalous traffic.
    - Piskozub, Michal, et al. "Malphase: Fine-grained malware detection using network flow data." Proceedings of the 2021 ACM Asia conference on computer and communications security. 2021. This paper does not directly utilize autoencoder's reconstruction error for detection, but integrates autoencoder reconstruction information in a larger detection and classification pipeline.
These are just two papers that came to my mind. However, many more papers in this area exist, and the authors should do a much better job of covering them and comparing them to their approach. Kitsune especially (first reference above) is already doing what is proposed here.
- Evaluation issues: the authors limited their evaluation to the NSL-KDD dataset, which does not provide a good overview of the performance of the approach. Moreover, even on this dataset the performance is dubious, as shown in Figure 7. The figure shows that a 10% false positive rate is required to achieve a true positive rate of 80%. This clearly makes the approach not applicable to real world deployments.

Review #48D
===

Overall merit
----

1. Reject

Reviewer expertise
----
3. Knowledgeable

Paper summary

This study delves into utilizing a traditional autoencoder architecture to strengthen network security by improving the efficacy of intrusion detection mechanisms. The main contributions of this research involve developing a specialized deep autoencoder trained exclusively on normal network behavior.

Comments for authors

The contribution of the paper is minor. It shows a simple ML experiment widely proposed in various works in the literature. Although the idea is promising, I think the paper lacks innovative content compared to the state of the art. The contributions should be better emphasized to highlight the proposed technique. In addition, I would suggest that the authors test the technique on more up-to-date and defect-free datasets, also considering possible side aspects such as the representativeness of the attacks. The data contained in NSL-KDD is in no way representative of modern attacks and networks.

Moreover, the whole approach is described in a nuanced manner. There is no mention of the cardinality of the training/validation/test sets. This information would allow the threshold selection criterion to be better understood and evaluated. Furthermore, the reader cannot be asked to evaluate the FPR by looking at the confusion matrices. This metric should always be reported. The comparative analysis is practically 'stitched' on its own solution. There are plenty of solutions in the literature with much better performance. To date, a NIDS that does a recall of 91.24 (with a significant number of false positives) cannot be considered a noteworthy solution.

Review #48E
===

Overall merit
----
2. Weak reject

Reviewer expertise
----
3. Knowledgeable

Paper summary

The paper uses a custom auto-encoder architecture to classify network intrusions. The encoder is trained to reconstruct "normal" network samples, and intrusions are detected as flows with a reconstruction loss above a specific threshold. The threshold is set at training time by comparing the loss of normal samples versus intrusions. The solution is evaluated on the NSL-KDD dataset and outperforms three other works.

Comments for authors

Thank you for your submission! This is an important problem, and overall I like your approach. However, I believe your paper's core contribution should be strengthened.

What I liked in the paper:

* Your explanation are clear for the most part, which makes the paper easy to follow and read
* I appreciate your data analysis to understand the impact of different features
* Your idea is simple yet effective
* You provide details about training which is great for reproducibility.

What I think could be improved:

* My main concern with your work is the lack of novelty. Auto-encoders have been used previously

for network classification ([23] from your paper, or paper [B] below). Specifically, [B] trains the model in the same way as you do, by providing normal samples at training time, and using training intrusions to set the threshold, on the same dataset. In general, this would be ok if you had an additional insight, but I don't see anything else stand out. **You should explain how your work is different from [B].**

* The paper lacks a clear explanation of how you select the threshold. In section 3.5 you say "*samples with reconstruction errors surpassing the predefined threshold of 0.0223 for minimum validation loss are identified as potential intrusions.*", and in section 4.3 "*the model attained the minimum validation loss of 0.0223, signifying the optimal set of weights stored during the training process*". In my understanding, the validation loss is the average loss of a normal sample in the validation set, so setting this as the threshold would lead to many normal samples being misclassified. Considering the good performance of your classifier, I'm under the impression the threshold is set like other papers by comparing the reconstruction loss between normal samples and intrusions in the training set. **This setting of the detection threshold deserves a better explanation as to clarify if training time intrusions are used to set the threshold or not.** I assume for the rest of my review that they are. If they are not, a detailed analysis of the losses of different sets of data (training / testing, normal / intrusions) would be great.

* If you do use training-time intrusions to set the threshold, your framework might not generalize well, as you rely on specific intrusions.

* Using a single dataset limits the generalizability of your method. For instance, it would be interesting to see how a model trained on this dataset performs on another, to see if the definition of "normal" traffic is valid across different networks. If not (and I suspect the answer will be no) that's fine, but it would be good to know.

* Additional evaluations would be nice, in order to understand the following aspects of your models:
    * How does the amount of training data impact quality? What is the minimal number of training samples to get optimal performance?
    * How does concept drift impact the models? How does the model perform on samples that occur later in time than the training samples.
    * Are new intrusions harder to detect? **For this you can use your current data by comparing the performance of the model on intrusions seen at training time versus those not (cf. Table 2).**

* Performance isn't state of the art. F1-score is higher in [10], and accuracy is close to that of [23] using an auto-encoder, and similar to the performance reported in [B]. If you look on the Kaggle page for this dataset, there are solutions using simple ML that get close to 100% accuracy.

* The metrics hide the fact that in real deployment, there are much more "normal" flows than intrusions (base rate fallacy). **Metrics like true positive rate or false positive rate would be more adapted** to understand what is the expected number of false positives that would be observed in real networks.

* Other nits:
    * You could be more concise in your writing, which would allow you to fit more experiments in a conference paper.
    * Fig 1 is difficult to read, and not very informative. A feature could be the same for 99.99% of all values and still be informative.
    * You don't need to include the definitions of accuracy, precision, recall and F1-score.
    * You are missing the "anonymous author" authorship.
    * You should include other works from the network intrusion detection space in your related works section. [B] and any other relevant auto-encoder anomaly detectors should be discussed.

Reference

[B] W. Xu, J. Jang-Jaccard, A. Singh, Y. Wei and F. Sabrina, "Improving Performance of Autoencoder-Based Network Anomaly Detection on NSL-KDD Dataset," in IEEE Access 2021