The Intellecta International Journal of Research and Innovation
Volume 1 Issue 1 (Oct-Dec 2026) | e-ISSN: Applied | Page No: 24-34.

A PREVALENCE ADJUSTED AND PROTOCOL-SPECIFIC EVALUATION OF MACHINE LEARNING-BASED INTRUSION DETECTION SYSTEMS

Dr. Varsha
Department of Computer Science and Engineering
Mumbai, Maharashtra, India
Email Id: v71633098@gmail.com

ABSTRACT

Machine learning intrusion detection is evaluated almost entirely through metrics computed across an entire test partition. This paper argues that the practice has two defects and demonstrates both with measurements. The first defect is protocol-level concealment: network protocols carry attack traffic in very different proportions, so an aggregate metric averages over problems of unequal base rate, and a detector that fails on a minority protocol can still report a respectable aggregate figure. The second defect is prevalence distortion: precision depends on the base rate of the positive class, and benchmark corpora carry attack prevalences one to two orders of magnitude above those of operational networks, so reported precision overstates what a deployed system will achieve. Six experiments were conducted, crossing three detectors with two benchmark corpora and evaluating 72,856 held-out records. A Random Forest reporting an aggregate recall of 0.7626 was found to attain a recall of 0.0361 within the ICMP subset, a difference of a factor of 21.1 that no aggregate statistic exposes. Detection quality across the five assessable protocol subsets correlated with subset attack prevalence at r = 0.7773. Holding measured sensitivity and specificity fixed, a precision of 0.9793 at benchmark prevalence corresponds to 0.3415 at an operational prevalence of one percent. The paper closes with a reporting protocol that addresses both defects and costs no additional data, training, or computation.

Keywords: Intrusion detection, evaluation methodology, protocol stratification, base rate, precision, class prior, reporting standards.

1. INTRODUCTION

Four or five numbers calculated on a held-out partition are typically used as the criterion to judge an intrusion detection system. The accuracy, precision, recall, the F1 score and the area under the receiver operating characteristic curve are reported as single values, compared against the previous values in previous work and the model with the larger values is considered the better detector. The convention has become so settled a matter that it seldom is mentioned, much less discussed.

It is analyzed in this paper. The convention is not very reliable due to two properties of network traffic, which can be measured and not simply stated. Both are not a drawback of a specific model. Both are flaws in the evaluation of models.

The first property is protocols vary in the amount of attack traffic they are transmitting. In the NSL-KDD corpus, 49.5 percent of TCP records are anomalous, 19.3 percent of UDP records, and 85.0 percent of ICMP records. A more general measure among the three is the weighted average of three problems at varying base rates and at varying difficulty, dominated by which protocol has the greatest number of records. Detectors can thus completely break on a single protocol and fail to impress on the aggregate numbers. It will be seen in Section V that this occurs.

The second property is that accuracy is not a definite property of a classifier. They are sensitive and specific: they characterize the behavior of the model on attack records and normal records respectively and are independent of the mixing of the two records. Precision is a changing quantity, since it requires the availability of a sufficient number of normal records to create false positives against which the availability of a sufficient number of attacks can create true positives. Attack prevalences close to half are used to construct benchmark corpora. Operation networks are of an approximately order of magnitude lower attack traffic rates. Reported precision thus characterizes a population, which none of the deployed systems will encounter. This is the argument that was made by Axelsson [1] in principle. Section VI quantifies it for a contemporary architecture.

The paper has more than three components added up with a very simple structure. It expresses the two flaws accurately, it quantifies them over six experiments with two benchmark corpora, and it suggests a reporting protocol that eliminates them. The protocol does not engage any new data collection, retrain or extra computation. Only that the existing metrics be calculated within protocol subsets and reported precision is accompanied by its extrapolation to a given deployment prevalence is required.

2. RELATED WORK

2.1 Current evaluation practice

Reported results in this field are overwhelmingly aggregate. Tang et al. [2] report a single accuracy of 0.7575 on NSL-KDD. Yin et al. [3] report 0.8129. Aygun and Yavuz [4] report 0.8828 and 0.8865 for two autoencoder variants, with precision and recall but no disaggregation. Naseer et al. [5] report 0.8900. On UNSW-NB15 the pattern repeats: Khan and Gumaei [6] report 0.7639, Malaiya et al. [7] 0.8190, Zhou et al. [8] 0.8363, and Min et al. [9] 0.8530. Jeong et al. [10], who assembled these comparisons, report 0.8939 and 0.8662 on the two corpora respectively. None of these studies reports performance separately by transport protocol, and none states the attack prevalence of the population on which precision was measured.

2.2 Existing critiques

The evaluation practices of this field have been criticised before, though not in the terms used here. Sommer and Paxson [11] argued that machine learning results in intrusion detection frequently fail to survive scrutiny of the evaluation procedure, and identified the gap between laboratory conditions and operational deployment as the principal reason. Ring et al. [12], surveying the available corpora, observed that protocol composition varies substantially between datasets and that this variation is not reflected in how results are reported. Axelsson [1] showed that the low base rate of intrusive activity in real traffic means that even a detector with high sensitivity and high specificity will produce alerts dominated by false positives.

These three critiques are separate, and the present paper connects the first two of them to measurement. Ring et al. identified the protocol variation but did not show what it does to reported metrics. Axelsson derived the base-rate result analytically but predates the architectures now in use. What follows supplies the measurements.

3. THE TWO DEFECTS STATED

3.1 Protocol-level concealment

Let a held-out partition contain subsets indexed by protocol, with n records and prevalence p in each. Any metric computed across the whole partition is a function of the pooled confusion matrix, and the pooled matrix is the elementwise sum of the subset matrices. A subset contributing few records therefore contributes little to the pooled value, whatever its internal behaviour.

The consequence is that the aggregate metric can be dominated by the majority protocol to an arbitrary degree. If TCP contributes 81.86 percent of a corpus and ICMP 6.28 percent, as in NSL-KDD, then a model performing well on TCP and failing entirely on ICMP will report an aggregate figure close to its TCP figure. The failure is not merely understated. It is invisible, because no aggregate statistic carries information about the composition of the partition it was computed over.

A second mechanism compounds the first. Because each protocol subset has its own class prior, a single decision threshold cannot be optimal for all of them. A threshold calibrated against the pooled prior will be too high for a subset in which attacks are scarce and too low for one in which attacks are abundant. Both errors are systematic rather than random, and neither is visible in a pooled metric.

3.2 Prevalence distortion of precision

Sensitivity r and specificity s are conditional on the true class and are therefore invariant to the class prior. Precision is not. For a population of N records with attack prevalence p, the model produces Npr true positives and N(1 minus p)(1 minus s) false positives, so precision equals pr divided by the quantity pr plus (1 minus p)(1 minus s). As p falls, the denominator becomes dominated by its second term and precision falls with it, though neither r nor s has changed.

The magnitude of the effect depends on the false positive rate. A detector with a false positive rate of 1.92 percent will retain usable precision down to moderate prevalence and lose it at low prevalence. A detector with a rate of 9.54 percent will lose it much sooner. Two models with similar accuracy can therefore behave very differently in deployment, and the metric that predicts which is which is the false positive rate rather than accuracy. Section VI shows both cases in the same experiment.

4. EXPERIMENTAL SETUP

The detectors used here are instruments for demonstrating the two defects, not the subject of the paper. Three were used so that the defects could be shown to affect models of different families rather than one architecture.

4.1 Corpora

Table I: The two benchmark corpora

Property	NSL-KDD	UNSW-NB15
Source	[13]	[14]
Total records	148,517	257,673
After protocol filtering	148,517	215,757
Attributes used	27	42
Held-out partition	29,704	43,152
Attack proportion	48.12%	63.91%

Table II: Protocol composition and attack prevalence

Corpus	Proto.	Records	Share	Prevalence
NSL-KDD	TCP	24,314	81.9%	49.37%
NSL-KDD	UDP	3,518	11.8%	19.13%
NSL-KDD	ICMP	1,872	6.3%	84.24%

UNSW	TCP	24,596	57.0%	45.19%
UNSW	UDP	18,550	43.0%	76.20%
UNSW	ICMP	6	0.01%	0.00%

Table II sets out the conditions under which the first defect operates. Within NSL-KDD, attack prevalence ranges across the three subsets from 19.13 percent to 84.24 percent, and the largest subset holds thirteen times as many records as the smallest. Within UNSW-NB15 the ordering of the two large subsets is reversed, TCP being near balanced and UDP attack dominant. The final row records a limitation that constrains part of the analysis: the ICMP subset of UNSW-NB15 holds six records, all of the normal class, and supports no inference. Every ICMP statement in this paper rests on NSL-KDD.

4.2 Detectors

Three detectors were trained on each corpus. A Decision Tree and a Random Forest [15], both implemented in scikit-learn [16], were constrained to a depth of two, the ensemble comprising five estimators. A Linformer [17], a linear-complexity variant of the transformer [18], was implemented in PyTorch [19] with three encoder layers, eight attention heads, a model dimension of 256 and 1,294,008 learnable parameters, trained for twenty epochs by Adam [20]. All three consumed an identical standardised feature matrix and were evaluated at a decision threshold of one half, so that differences between them are attributable to the model rather than to the conditions of assessment.

4.3 Metrics and stratification

Five primary metrics were computed together with specificity, the false positive rate, the false negative rate, balanced accuracy and the Matthews correlation coefficient. Each was computed twice: once over the whole held-out partition and once within each protocol subset, using a boolean mask applied simultaneously to true labels, predicted labels and anomaly scores so that the three arrays stay aligned. Where a subset holds records of only one class the area under the curve is undefined, and where it holds no positive records precision and recall are undefined. Such cells are reported as undefined rather than as zero, because a precision of zero asserts that every alarm raised was spurious whereas an undefined precision asserts that no positive record existed.

5. EVIDENCE OF PROTOCOL-LEVEL CONCEALMENT

Table III. Aggregate results for all six experiments

Corpus	Detector	Acc.	Prec.	Rec.	F1	AUC
NSL-KDD	Dec. Tree	0.9020	0.9234	0.8678	0.8947	0.9240
NSL-KDD	Rand. Forest	0.8784	0.9795	0.7626	0.8575	0.9649
NSL-KDD	Linformer	0.9823	0.9793	0.9838	0.9816	0.9987
UNSW	Dec. Tree	0.8481	0.7939	1.0000	0.8851	0.9225
UNSW	Rand. Forest	0.8849	0.8988	0.9051	0.9020	0.9652
UNSW	Linformer	0.9231	0.9326	0.9362	0.9344	0.9832

Table III is the sort of table a reader of this literature expects. Read on its own it supports a mild conclusion: the Linformer is best, the two classical models are broadly comparable, and nothing on NSL-KDD falls below an accuracy of 0.87. Table IV disaggregates the same six experiments.

Table IV. Protocol-stratified results on NSL-KDD

Protocol	Records	Detector	Accuracy	Precision	Recall	F1	AUC
TCP	24,314	Decision Tree	0.9201	0.9194	0.9188	0.9191	0.9507
TCP	24,314	Random Forest	0.9272	0.9825	0.8680	0.9217	0.9724
TCP	24,314	Linformer	0.9854	0.9832	0.9873	0.9852	0.9991
UDP	3,518	Decision Tree	0.9358	0.8956	0.7519	0.8174	0.8723
UDP	3,518	Random Forest	0.9105	0.9163	0.5854	0.7144	0.8876
UDP	3,518	Linformer	0.9582	0.8845	0.8990	0.8917	0.9900
ICMP	1,872	Decision Tree	0.6031	1.0000	0.5289	0.6918	0.8906
ICMP	1,872	Random Forest	0.1848	0.9048	0.0361	0.0695	0.6842
ICMP	1,872	Linformer	0.9866	0.9911	0.9930	0.9921	0.9984

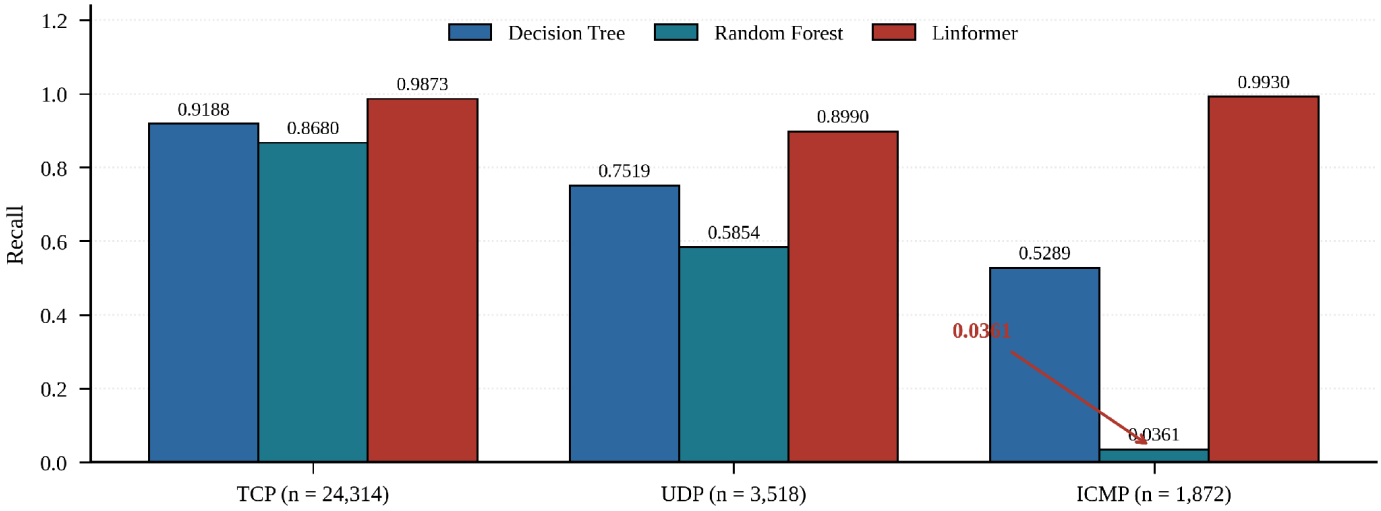


Fig. 1. Recall by protocol on NSL-KDD for the three detectors. The Random Forest reports an aggregate recall of 0.7626 while detecting fewer than four ICMP attacks in every hundred.

The Random Forest row for ICMP in Table IV is the finding this paper was written to report. Aggregate recall for that model is 0.7626. Within the ICMP subset it is 0.0361. The model detects fewer than four ICMP attacks in every hundred present, and its accuracy of 0.1848 on that subset falls below the 0.8424 obtainable by flagging every ICMP packet without inspection. Aggregate recall exceeds subset recall by a factor of 21.1.

Nothing in Table III hints at this. The aggregate accuracy of 0.8784 is unremarkable, the aggregate precision of 0.9795 is the highest in that column, and a reader comparing this model against published work would place it in the middle of the field. The disaggregation in Table IV shows that the model is a competent detector of TCP and UDP attacks and close to useless on ICMP.

The mechanism is the one set out in Section III-A. The ICMP subset is attack dominant at 84.24 percent while the corpus as a whole is near balanced, so a shallow model fitted against the pooled prior places its boundary where most ICMP attacks fall on the normal side. The Decision Tree, which received inverse-frequency class weighting, degrades less

severely but still falls to a recall of 0.5289 on the same subset. The Linformer attains 0.9930, which shows that the failure is a property of the models rather than of the subset.

TABLE V. Protocol-stratified results on UNSW-NB15

Proto.	Records	Detector	Acc.	Prec.	Rec.	F1
TCP	24,596	Dec. Tree	0.7859	0.6785	1.0000	0.8085
TCP	24,596	Rand. Forest	0.8530	0.8680	0.7957	0.8303
TCP	24,596	Linformer	0.9001	0.8923	0.8858	0.8891
UDP	18,550	Dec. Tree	0.9309	0.9168	1.0000	0.9566
UDP	18,550	Rand. Forest	0.9274	0.9198	0.9912	0.9541
UDP	18,550	Linformer	0.9535	0.9637	0.9757	0.9697
ICMP	6	All three	undefined	undefined	undefined	undefined

Table V shows a second variety of concealment on the other corpus. The Decision Tree attains a recall of exactly 1.0000 in aggregate and again within both assessable subsets. Read alone this looks like a detector that misses nothing. The accompanying precision of 0.7939 in aggregate and 0.6785 on TCP shows what has happened: the model assigns the positive class to nearly every record, so it captures every attack by flagging everything. On a corpus with an attack proportion of 63.91 percent, a model flagging every record without exception would attain a recall of 1.0000 and a precision of 0.6391. The observed values sit just above that floor.

A. Derived metrics within subsets

The primary metrics are not the only quantities that vary by subset. Table VI reports specificity, the two error rates and the Matthews correlation coefficient for the strongest detector within each subset, and these are the quantities on which requirement R4 of Section VII depends.

Table VI. Derived metrics within each subset, Linformer

Corpus	Proto.	Spec.	FPR	FNR	MCC
NSL-KDD	TCP	0.9835	1.65%	1.27%	0.9708
NSL-KDD	UDP	0.9722	2.78%	10.10%	0.8658
NSL-KDD	ICMP	0.9525	4.75%	0.70%	0.9495
NSL-KDD	Aggr.	0.9808	1.92%	1.62%	0.9645
UNSW	TCP	0.9119	8.81%	11.42%	0.7982
UNSW	UDP	0.8824	11.76%	2.43%	0.8704
UNSW	Aggr.	0.9046	9.54%	6.38%	0.8415

Two features of Table VI bear on the argument. The false negative rate on the NSL-KDD UDP subset is 10.10 percent against an aggregate figure of 1.62 percent, so the aggregate understates by a factor of six the rate at which attacks escape detection on the protocol where they are scarcest. The Matthews coefficient, which incorporates all four cells of the

confusion matrix and is therefore insensitive to the class prior, ranges from 0.8658 to 0.9708 within NSL-KDD alone. A single aggregate value of 0.9645 conveys none of that spread.

The aggregate rows of Table VI also show why requirement R4 asks for the false positive rate rather than accuracy. The two corpora differ in aggregate accuracy by 5.92 percentage points, which reads as a small gap, while their false positive rates differ by a factor of 4.97. Section VI shows that it is the second difference, not the first, that determines how the two behave in deployment.

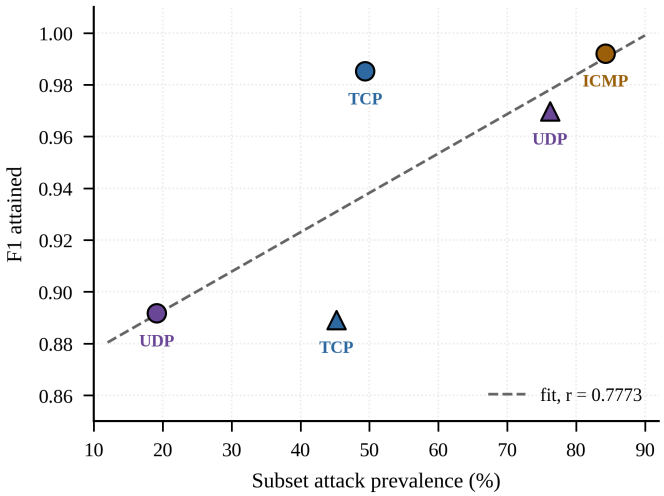


Fig. 2. Detection quality against subset attack prevalence across the five assessable subsets.

Table VII. Detection quality against subset attack prevalence

Corpus	Proto.	Records	Prevalence	F1
NSL-KDD	TCP	24,314	49.37%	0.9852
NSL-KDD	UDP	3,518	19.13%	0.8917
NSL-KDD	ICMP	1,872	84.24%	0.9921
UNSW	TCP	24,596	45.19%	0.8891
UNSW	UDP	18,550	76.20%	0.9697

Table VII and Fig. 2 record the second mechanism from Section III-A operating on the strongest of the three detectors. Across the five assessable subsets, F1 rises with subset attack prevalence, with a Pearson correlation of 0.7773 and a fitted slope near 0.0015 per percentage point. Five observations cannot support an inferential claim, and none is made. The pattern is consistent with the account given: a threshold of one half suits a subset whose prior is near one half, and the false negative rate of 10.10 percent on the attack-scarce NSL-KDD UDP subset, against 1.27 percent on TCP, is what that mismatch costs. Because every subset area under the curve stays above 0.99, the ordering the model produces within each subset is sound, and the deficit lies in where the threshold sits rather than in what the model has learned.

6. EVIDENCE OF PREVALENCE DISTORTION

The projection below holds the measured sensitivity and specificity of each Linformer fixed and varies only the assumed attack prevalence of the deployed traffic. Sensitivity and specificity are read from the held-out confusion matrices: 0.9838 and 0.9808 for the NSL-KDD model, 0.9362 and 0.9046 for the UNSW-NB15 model.

Table VIII. Projected statistics per million flows, NSL-KDD model

Prev.	TP	FP	Alerts	Prec.	False
0.5%	4,919	19,063	23,982	0.2051	79.5%
1.0%	9,838	18,967	28,805	0.3415	65.8%
2.0%	19,676	18,775	38,451	0.5117	48.8%
5.0%	49,190	18,201	67,390	0.7299	27.0%
10.0%	98,379	17,243	115,622	0.8509	14.9%
25.0%	245,949	14,369	260,317	0.9448	5.5%
50.0%	491,897	9,579	501,476	0.9809	1.9%

Table IX. Projected statistics per million flows, UNSW-NB15 model

Prev.	TP	FP	Alerts	Prec.	False
0.5%	4,681	94,876	99,557	0.0470	95.3%
1.0%	9,362	94,399	103,761	0.0902	91.0%
2.0%	18,723	93,445	112,169	0.1669	83.3%
5.0%	46,808	90,585	137,393	0.3407	65.9%
10.0%	93,616	85,817	179,433	0.5217	47.8%
25.0%	234,040	71,514	305,554	0.7660	23.4%
50.0%	468,079	47,676	515,755	0.9076	9.2%

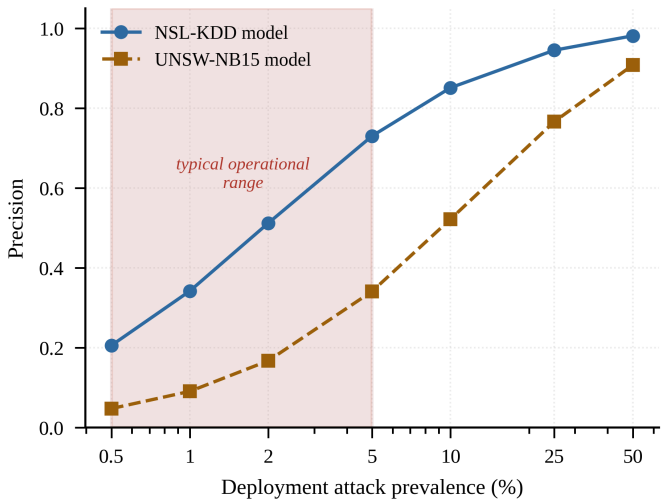


Fig. 3. Precision against deployment attack prevalence for the two models.

At the benchmark prevalence of roughly one half the projection returns 0.9809 for the NSL-KDD model against a measured 0.9793, a discrepancy of 0.0016 that validates the arithmetic. Below that point precision falls steadily. At five percent it is 0.7299, at one percent 0.3415 and at half a percent 0.2051. The model has not changed. Its sensitivity is 0.9838 and its specificity 0.9808 at every row of Table VIII.

The UNSW-NB15 model deteriorates faster because its false positive rate is 9.54 percent against 1.92 percent, a difference of a factor of 4.97. At one percent prevalence its precision is 0.0902, so fewer than one alert in ten would be genuine, and at half a percent 0.0470. The two models differ in aggregate accuracy by 5.92 percentage points, which reads as a modest gap. In deployment at one percent prevalence they differ in precision by a factor of 3.8.

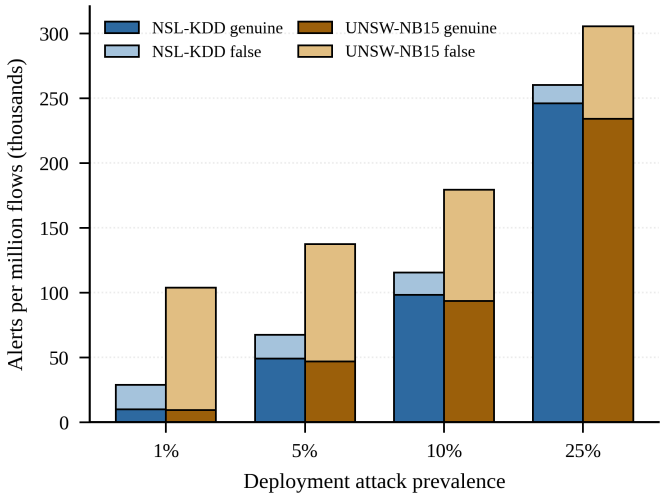


Fig. 4. Projected alert composition per million flows at four deployment prevalences.

Fig. 4 converts the projection into the quantity that determines whether a system is usable, which is the number of alerts an analyst team must process. On a network carrying one million flows a day at one percent attack prevalence, the NSL-KDD model would raise 28,805 alerts of which 65.85 percent are spurious, and the UNSW-NB15 model 103,761 of which 90.98 percent are spurious. At a review rate of one hundred alerts per analyst per day these correspond to 288 and 1,038 analysts. Neither figure is absorbable by manual triage, and neither is visible in any reported metric.

7. A CORRECTED REPORTING PROTOCOL

Both defects can be removed by additions to what is reported rather than by changes to what is built. The protocol below requires no additional data, no retraining and no additional computation beyond recomputing existing metrics over subsets.

Table X. Proposed reporting requirements

No.	Requirement	Defect
R1	Report record count and attack prevalence of every protocol subset	Concealment
R2	Report every primary metric within each subset alongside the aggregate	Concealment
R3	Report undefined, not zero, where a subset holds one class	Concealment
R4	Report specificity and the false positive rate, not accuracy alone	Distortion
R5	State the attack prevalence of the evaluation population	Distortion
R6	Report projected precision at stated deployment prevalences	Distortion

R7	Report projected alert volume per unit of traffic	Distortion

The first three requirements make the composition of the evaluation population visible and prevent a subset failure from being averaged away. Requirement R3 matters more than its brevity suggests: the metric libraries in common use return zero for undefined precision and recall under their zero-division policy, and a table of zeros invites the reading that the model failed rather than that no evidence existed. The ICMP row of Table V is such a case.

The last four requirements make the deployment gap explicit. R4 shifts attention to the metric that governs the alert burden. R5 makes the population behind a precision figure inspectable. R6 and R7 convert the figure into what a practitioner needs, which is how many alerts arrive and how many of them are worth opening. The projection requires only the sensitivity, the specificity and an assumed prevalence, all of which an author already has. The cost of the protocol is a table and two additional rows.

8. DISCUSSION

These two defects are not related as far as their origin is concerned, however, similar in terms of effect. They both occur since a single number is requested to characterize a heterogeneous population, and they are both eliminated by characterizing a population instead. Neither is a critique to any of the models discussed in this paper. The Random Forest which fails on ICMP is not a model that is poorly constructed. It is a model of which the reporting convention conceals failure.

The consequence of the protocol level is an implication on the manner of selection of detectors. When comparing two models based on aggregate metrics, and one of them fails with respect to a minority protocol, the comparison can choose the poor system to be deployed in a network where the minority protocol is important. The fact here does not determine the frequency with which it is the case in the published literature, since this literature is not reporting the figures to check. It proves that the preconditions to it can be found in two of the most popular benchmark corpora.

The outcome of prevalence has a ramification of the deployment of detectors. The measured characteristics of the NSL-KDD model in a system running on a network and achieving a detection rate of one percent prevalence of attacks generate two spurious alerts to each true alert. It is a triage issue and not a detection issue, and suggests the implementation of such a system as an initial step to feed a second filter and not as an independent blocking system. It also proposes the choice of candidate models on the false positive rate. The two models here compared on that criterion differ by a factor of five, which is not reflected by the accuracies of 0.9823 and 0.9231 of the models.

Another observation has to do with the direction of the prevalence effect. Since the prevalence decreases as precision decreases and no changes occur in recall, the weighted average of the two changes in deployment, although the model remains constant. The detector that is desensitized such that it can set a balanced precision and recall on a benchmark will be poor in precision and rich in recall in the field. Assuming that the tuning goal was the F1 score, the operating point they selected on the benchmark is not that operating point that their execution could have selected had they used the deployment prevalence.

9. LIMITATIONS

Four limitations qualify these findings. The baselines were constrained to a depth of two and five estimators, which makes them weak models; the concealment effect is demonstrated on weak models and its magnitude on stronger ones is not established here, though the mechanism of Section III-A does not depend on model capacity. The held-out ICMP subset of UNSW-NB15 holds six records of one class and supports no inference, so the protocol-level evidence rests on one corpus. The prevalence-quality association of Table VII rests on five observations and is offered as a pattern with a stated

mechanism rather than as an estimated relationship. And the deployment projection assumes that sensitivity and specificity measured on benchmark traffic transfer unchanged to operational traffic; this assumption is standard but has not been validated on live data here.

10. CONCLUSION

Aggregate evaluation of intrusion detection systems has two measurable defects. Metrics computed across a whole partition can conceal complete failure on a minority protocol, and precision measured at benchmark prevalence does not describe precision at deployment prevalence. Both were demonstrated across six experiments on 72,856 held-out records.

A Random Forest reporting an aggregate recall of 0.7626 attained 0.0361 within the ICMP subset, a difference of a factor of 21.1. Detection quality across five protocol subsets tracked subset attack prevalence at $r = 0.7773$. A precision of 0.9793 measured at benchmark prevalence corresponds to 0.3415 at one percent operational prevalence, and to an alert stream two thirds of which is spurious.

The seven reporting requirements of Section VII remove both defects at the cost of one table and two rows. They do not require better models, larger corpora or additional experiments. They require only that what is already computed be reported in a form that reflects the population it was computed over.

REFERENCES

[1] S. Axelsson, "The base-rate fallacy and the difficulty of intrusion detection," ACM Trans. Inf. Syst. Secur., vol. 3, no. 3, pp. 186-205, 2000.

[2] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep learning approach for network intrusion detection in software defined networking," in Proc. Int. Conf. Wireless Netw. Mobile Commun. (WINCOM), 2016, pp. 258-263.

[3] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," IEEE Access, vol. 5, pp. 21954-21961, 2017.

[4] R. C. Aygun and A. G. Yavuz, "Network anomaly detection with stochastically improved autoencoder based models," in Proc. IEEE 4th Int. Conf. Cyber Secur. Cloud Comput. (CSCloud), 2017, pp. 193-198.

[5] S. Naseer et al., "Enhanced network anomaly detection based on deep neural networks," IEEE Access, vol. 6, pp. 48231-48246, 2018.

[6] F. A. Khan and A. Gumaei, "A comparative study of machine learning classifiers for network intrusion detection," in Proc. 5th Int. Conf. Artif. Intell. Secur. (ICAIS), 2019, pp. 75-86.

[7] R. K. Malaiya, D. Kwon, S. C. Suh, H. Kim, I. Kim, and J. Kim, "An empirical evaluation of deep learning for network anomaly detection," IEEE Access, vol. 7, pp. 140806-140817, 2019.

[8] X. Zhou, Y. Hu, W. Liang, J. Ma, and Q. Jin, "Variational LSTM enhanced anomaly detection for industrial big data," IEEE Trans. Ind. Informat., vol. 17, pp. 3469-3477, 2020.

[9] B. Min, J. Yoo, S. Kim, D. Shin, and D. Shin, "Network anomaly detection using memory-augmented deep autoencoder," IEEE Access, vol. 9, pp. 104695-104706, 2021.

[10] J. Jeong, S. Park, J. Lim, J. Kang, D. Shin, and D. Shin, "A study on network anomaly detection using fast persistent contrastive divergence," Symmetry, vol. 16, no. 9, art. 1220, 2024.

[11] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in Proc. IEEE Symp. Secur. Privacy, 2010, pp. 305-316.

[12] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," Comput. Secur., vol. 86, pp. 147-167, 2019.

[13] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl. (CISDA), 2009, pp. 1-6.

[14] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS), 2015, pp. 1-6.

[15] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5-32, 2001.

[16] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825-2830, 2011.

[17] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, "Linformer: Self-attention with linear complexity," arXiv:2006.04768, 2020.

[18] A. Vaswani et al., "Attention is all you need," in Adv. Neural Inf. Process. Syst., vol. 30, 2017, pp. 5998-6008.

[19] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in Adv. Neural Inf. Process. Syst., vol. 32, 2019, pp. 8024-8035.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. Int. Conf. Learn. Represent. (ICLR), 2015.