Comparison of Machine Learning Algorithms trained under Differential Privacy for Intrusion Detection Systems

Ioannis Siachos
Eight Bells Ltd, Nicosia, Cyprus
ioannis.siachos@8bellsresearch.com

Konstantinos Kaltakis
Eight Bells Ltd, Nicosia, Cyprus
konstantinos.kaltakis@8bellsresearch.com

Konstantina Papachristopoulou
Eight Bells Ltd, Nicosia, Cyprus
konstantina.papachristopoulou@8bellsresearch.com

Ioannis Giannoulakis
Eight Bells Ltd, Nicosia, Cyprus
giannoul@8bellsresearch.com

Emmanouil Kafetzakis
Eight Bells Ltd, Nicosia, Cyprus
mkafetz@8bellsresearch.com

***Abstract*—Intrusion Detection Systems (IDS) are valuable tools for the proper identification and the timely response to potential security threats in a network, using traffic analysis and anomalous activities detection. Traditional IDS rely on rule-based or signature-based methods to detect known cyber attacks, but these methods often fail to detect novel ones. There has been a growing interest recently, in using Machine Learning (ML) algorithms to enhance the detection capabilities of IDS. As a downturn, the datasets used by ML algorithms for IDS applications refers to network logs which may contain sensitive information, resulting in privacy threats. To address this issue, Differential Privacy (DP) can be used to preserve the privacy of network logs, while still allowing the ML algorithm to extract useful information from the data. In this work we test the performance of four popular ML classifiers (Gaussian Naive Bayes, Logistic Regression, Support Vector Machines, Random Forest Classifier) in the CIC-IDS2017 dataset when a DP mechanism is added to each algorithm in comparison with the classical non-DP setting.**

***Index Terms*—Intrusion Detection System, Differential Privacy, Machine Learning**

I. Introduction

Intrusion Detection Systems (IDS) [1] represent a critical component of network security. IDS are designed to identify and respond to potential security threats by analyzing network traffic and identifying anomalous activities. With the increasing complexity of network environments and the rise of sophisticated cyber attacks, IDS are used for protection against data breaches and cyber threats. Unauthorized access, malware, phishing attacks, and other security threats that can possibly compromise the confidentiality, integrity, and availability of network resources are exposed by IDS. IDS are the first line of defense in network security and are essential in maintaining the confidentiality and integrity of sensitive data.

In recent years, there has been a growing interest in using Machine Learning (ML) algorithms to enhance the detection capabilities of IDS. Large volumes of network data can be analyzed by ML algorithms to find patterns and anomalies as indications of malicious activities [2]. Moreover, ML algorithms can adapt to new attack patterns and learn from new data, making them well-suited for the dynamic and ever-changing nature of network security. The use of ML in IDS has shown promising results, with studies reporting high detection rates and low false positives [3], [4].

However, the demonstrated efficiency in detection often comes at the cost of privacy, since ML algorithms for IDS use datasets such as network logs which may contain sensitive information. In a corporate network, the logs generated by the network infrastructure contain sensitive information about connected devices, IP addresses, port numbers, operating systems and network topology [3]. This presents a serious threat, when considering the deployment or sharing of ML-based IDS due to the numerous privacy attacks that are possible. Membership Inference, Reconstruction and Property Inference Attacks among others, can cause the disclosure of sensitive information about a network [5].

To address this issue, various privacy-preserving mechanisms such as Differential Privacy (DP) can be used. DP [6] is a formal framework for ensuring privacy in queries and calculations on a dataset. In the context of ML, by adding random noise to model training, the privacy of network logs can be preserved, while still allowing the ML algorithm to extract useful information from the data. DP can be applied to various ML algorithms, such as logistic regression or decision trees, to enhance the privacy and security of the IDS. The main advantage of DP over other privacy-preserving techniques is the fact that it provides provable guarantees about individual's privacy.

II. Background

A. Differential Privacy

Differential privacy [6] is a mathematical concept that provides a rigorous framework for measuring the privacy of data analysis. Formally, a randomized algorithm A satisfies ϵ-DP if, for any two datasets D and D', differing in the presence or absence of a single individual's data, and for any subset of outputs S of the algorithm:

$$P(f(D) \in S) \leq e^{\epsilon} P(f(D') \in S), \tag{1}$$

where ϵ is a small positive constant. Intuitively, this means that the output of the algorithm is roughly the same whether or not a particular individual's data is included in the dataset. Lower values of ϵ correspond to stronger privacy.

DP can be used in ML algorithms to protect the privacy of individuals whose data is used for training the algorithm. This is achieved by the addition of artificial noise during the training of the algorithm, such as Laplacian or Gaussian noise.

B. Gaussian Naive Bayes

The Gaussian Naive Bayes [7] algorithm is based on Bayes' theorem, and assumes that the features of a dataset are independent and identically distributed Gaussian random variables. The algorithm calculates the conditional probability of a class given a set of features using Bayes' theorem and then predicts the class with the highest probability: Due to its simplicity and efficiency, the Gaussian Naive Bayes algorithm is widely used in a variety of applications, such as text classification, spam filtering, and image recognition. Despite its naive assumptions, the algorithm often performs surprisingly well in practice and can serve as a strong baseline for more complex ML models.

In order for the Gaussian Naive Bayes algorithm to be DP, the authors in [8] first calculate the sensitivity of each feature. The sensitivity is the maximum value of difference between any pair of features. Then, according to the sensitivity values that were calculated, Laplacian noise is added to the parameters of the model. Thus, the algorithm achieves DP.

C. Logistic Regression

Logistic regression [9] is widely used in statistics and in classification tasks. It consists of the fitting of a linear model that assumes a logistic function of the input features in order to derive the binary outcome. The logistic function maps real-valued inputs to $(0, 1)$, which can then be interpreted as the probability that the input sample belongs to the positive class. The model is trained using maximum likelihood estimation. The coefficients can be interpreted as the weights assigned to each feature in the model. Logistic regression is widely used in a variety of applications, such as medical diagnosis, credit scoring, risk prediction, etc.

The authors in [10] show that the addition of Laplacian noise to the output of the minimization of the objective function of the Logistic Regression algorithm outputs a DP version of the original algorithm.

D. Support Vector Machines

Support Vector Machines (SVM) [11] is a supervised learning model based on statistical learning theory. The intended result is to find the farthest border in the classes that it separates. SVM combine an optimisation theory algorithm with statistical learning theory for training, and use linear functions for data classification, after transforming the data into a higher-dimensional space where it is more easily separable. SVM maximize predictive accuracy while automatically avoiding over-fit to the data [12]. There are a number of known advantages of SVM, including their high performance when there is a clear margin of separation between classes. SVM are also more effective in high dimensional spaces and with small datasets. They perform well on out-of-sample data, which makes them fast and efficient.

In [10], it is highlighted that the output hyperplane of the original SVM algorithm can cause leakage of information about the training data. In order to alleviate this liability, the objection function is perturbed with the addition of a term that contains a carefully calculated random vector and a different normalization term that depends on the choice of ϵ. This creates a DP-SVM algorithm.

E. Random Forest Classifier

Random forest classifier [13] is a powerful and widely used ML algorithm for classification tasks. It is an ensemble method that combines multiple decision trees to improve the accuracy and stability of the model. The algorithm constructs a set of decision trees by randomly selecting a subset of features and data points, and then averaging the predictions of the individual trees to obtain the final prediction. This approach reduces overfitting and improves the generalization performance of the model. Random forests are capable of handling both categorical and continuous features, and can capture complex nonlinear relationships between the features and the target variable. They are also robust to noise and outliers, making them suitable for dealing with noisy or incomplete data. Random forests are widely used in various applications, such as credit scoring, customer churn prediction, and image classification. However, they can be computationally expensive and require a large amount of memory for training and prediction. To address these issues, various optimization techniques, such as parallelization or feature selection, can be applied to improve the efficiency of the algorithm.

In [14], a DP Random Forest Classifier is constructed by using queries that return class labels by the Exponential Mechanism instead of a count query. The Exponential Mechanism works by sampling a distribution centered around the true value for the output of a query.

III. Data and Methods

A. Dataset

The CIC-IDS2017 dataset [15] is a recent and widely used benchmark dataset for evaluating IDS. It contains network traffic data produced by a variety of network attacks and benign network activities. The dataset is labeled with different types of attacks, including brute force, denial of service, and heartbleed, among others. The dataset is composed of both raw packet data and flow data, and it includes over 2.5 million records. It includes a wide range of network traffic features that can be used to develop and evaluate IDS. These features

Average Packet Size	Flow Bytes/s	Max Packet Length
Subflow Fwd Bytes	Fwd IAT Min	Avg Fwd Segment Size
Fwd Packet Length Max	Flow IAT Mean	Fwd Header Length
Fwd IAT Mean	Fwd IAT Total	Fwd Packets/s
Packet Length Mean	Total Length of Fwd Packets	Fwd Packet Length Mean
Flow Duration	Flow IAT Std	Fwd IAT Std
Flow Packets/s	Fwd IAT Max	

TABLE I
DATASET FEATURES

are extracted from raw network packet data and flow data, and include both basic and advanced features. Some of the basic features include packet and byte counts, inter-arrival times, and protocol type. Advanced features include statistical features such as mean, variance, and standard deviation of packet and byte counts, as well as more complex features such as entropy, total number of packets and bytes, and number of unique IP addresses. Additionally, the dataset includes features related to network port numbers, network flow direction, and network flow duration. These features provide a comprehensive view of network traffic and allow for the development of effective IDS. Nevertheless, they include valuable private information about the network and its structure, so we will apply DP in our effort to preserve their privacy and protect them. In our work we follow a data processing methodology found in [16]. As such we used a version of the dataset that contains 170366 records and has the features found in table I. It consists of a subset of the original dataset with samples labeled as Benign or Malicious.

B. Evaluation Framework

To evaluate the impact of DP on the performance of ML algorithms for IDS, we conducted an experiment in which we trained and tested Gaussian Naive Bayes, Logistic Regression, Support Vector Machines and Random Forest Classifier algorithms. For each algorithm, we trained both a DP version and a non-private version as a baseline, and evaluated their performance using the F1 score metric with 5-fold cross-validation on each run. The choice of F1 score metric is motivated by the need to balance precision and recall in the detection of security threats. By comparing the F1 scores of DP and non-private versions of each algorithm, we aim to determine the impact of DP on the accuracy and generalization performance of IDS. We chose to test 50 values of ϵ between 0.01 to 1 and train the classifiers with baseline hyperparameters that can be found in the scikit-learn [17] python library implementations of the algorithms. The DP versions of the algorithms were implemented from scratch. Due to the imbalanced nature of the dataset, a random undersampling were conducted in each of the train sets of the 5-fold cross validation procedure.

IV. RESULTS

By comparing the F1 scores of each classifier with and without DP, it is expected that the scores increase as the ϵ value increases. Intuitively, as a larger ϵ value represents a weaker privacy guarantee, leading to a higher degree of noise added, which in turn affects the classifier's performance. The baseline scores can be seen in Table II. The charts with ϵ on the x-axis and F1 score on the y-axis for each classifier can be found in Figures 1 to 5.

For Gaussian Naive Bayes, the highest F1 score of 0.825 was achieved when ϵ was set to 0.072. This score is 0.017 lower than the baseline score of 0.842, indicating that DP has a low negative impact on the performance of Gaussian Naive Bayes for a relatively low value of ϵ.

For Logistic Regression, the highest F1 score of 0.908 was achieved when ϵ was set to 0.829. Again, this score is slightly lower than the baseline score of 0.921 (with a gap of 0.013), indicating that DP has a negative impact on the performance of Logistic Regression as well. Also, the best score was achieved with a relatively high ϵ value.

For Random Forest, the highest F1 score of 0.907 was achieved when ϵ was set to 0.0282. This score is 0.045 lower than the baseline score of 0.952. Nevertheless, we can observe that a good F1-score was achieved for a relatively low ϵ value.

For SVM, the highest F1 score of 0.896 was achieved when ϵ was set to 0.0543. This score is still lower than the baseline score of 0.92, but a also high score of 0.895 was achieved when ϵ was set to 0.01.

Among the four classifiers, Random Forest has the best overall performance without DP, but SVM has the lowest gaps as ϵ varies, indicating that behaves better under DP. For lower values of ϵ, SVM clearly performs best. The Gaussian Naive Bayes and Logistic Regression Classifiers follow the expected behavior of higher-ϵ/better-performance while the Random Forest and SVM dont. For the second case, this is not an expected behavior. For instance, the scores of Random Forest on the experiments conducted in [14] follow the general rule but ours don't. Also, as a general observation, all classifiers give fluctuating results. This gives us an insight on the complex nature of the DP noise addition mechanisms. The author's take on this is that the CIC-IDS2017 dataset presents a challenging feature space that is very sensitive to noise and algorithm choice. However, it is worth noting that the performance of all classifiers is impacted negatively to some degree by the addition of DP, but we can observe acceptable F1-scores for low ϵ values in all the classifiers.

As a final note, since (to our knowledge) there is no prior work that compares the performance of ML algorithms with and without DP in a IDS dataset, we have no baseline for the comparison of our results.

Algorithm	F1 score
Gaussian NB	0.842
Logistic Regression	0.921
Random Forest	0.952
Support Vector Machines	0.92

TABLE II
BASELINE ALGORITHMS AVERAGE F1 SCORE

V. CONCLUSION

This work presents a comparison between some ML classifiers when trained under a DP setting in the CIC-IDS2017

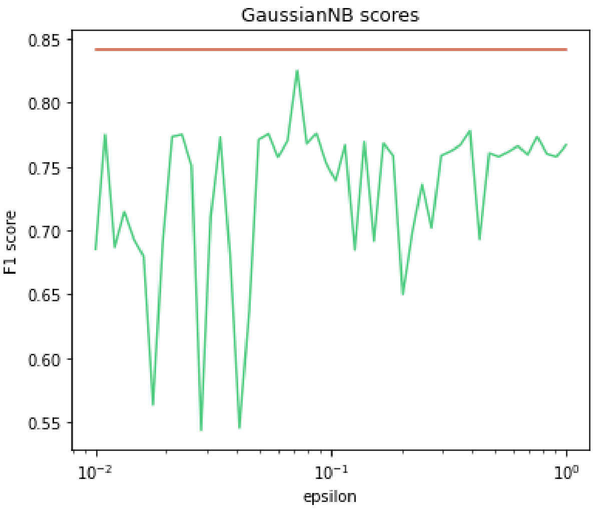

Fig. 1. Gaussian NB Results

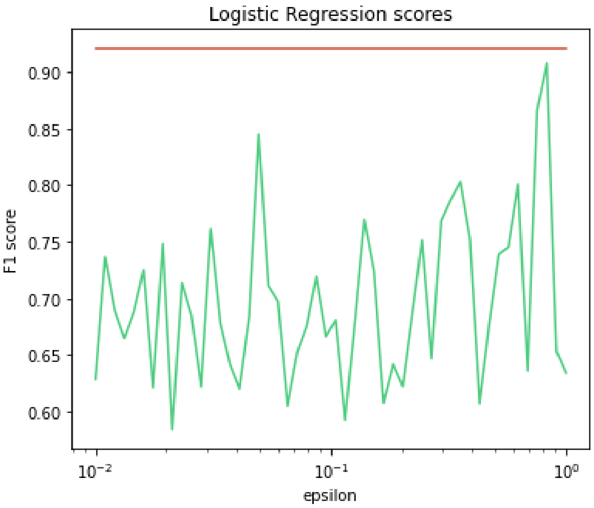

Fig. 2. Logistic Regression Results

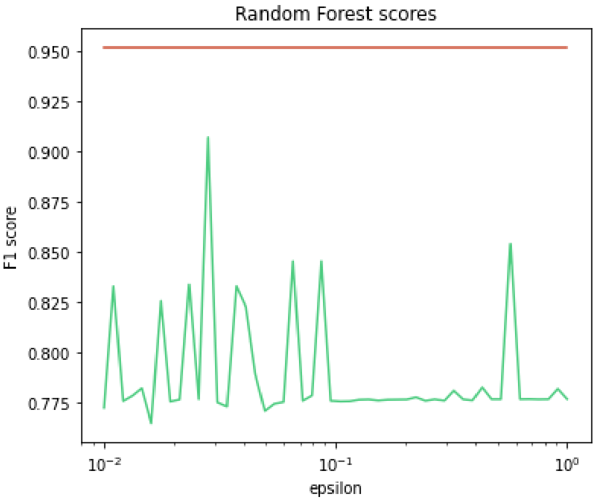

Fig. 3. Random Forest Results

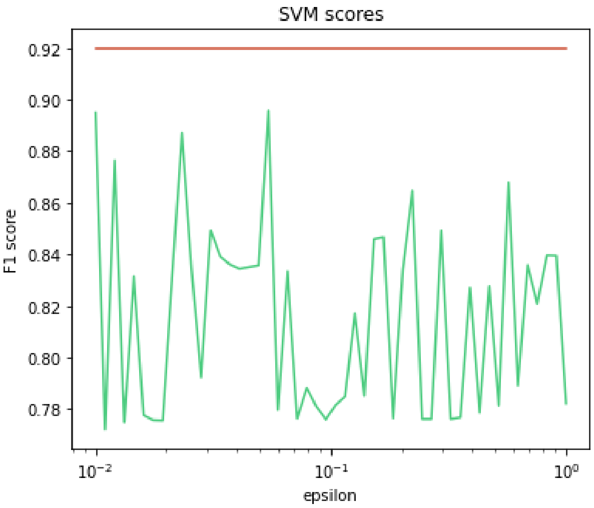

Fig. 4. Support Vector Machines Results

dataset. The classifiers under investigation were Gaussian Naive Bayes, Logistic Regression, Support Vector Machines and Random Forest. Our experiments show that we can find ϵ values that provide good privacy guarantees while not burdening the overall classifier performance in a substantial degree. As a future work, we plan to extend our experiments to more popular ML algorithms such as k-Nearest Neighbours, Multilayer Perceptrons, Graph Neural Networks, etc. Also, we want to perform extensive hyperparameter search to the aforementioned algorithms and to the ones used in our work. Finally we ought to include more IDS datasets in order to have a better picture about the use of DP in the context of ML-based IDS. We hope that this will help us have a clearer view on some of the problematic results from the experiments, as discussed in Section IV.

Acknowledgment

The research for this paper is supported by the Horizon Europe ENCRYPT project (Grant Agreement no 101070670).

References

[1] H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, Jan. 2013. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1084804512001944

[2] I. H. Sarker, "Machine learning for intelligent data analysis and automation in cybersecurity: current and future prospects," *Annals of Data Science*, pp. 1–26, 2022.

[3] J. O. Mebawondu, O. D. Alowolodu, J. O. Mebawondu, and A. O. Adetunmbi, "Network intrusion detection system using supervised learning paradigm," *Scientific African*, vol. 9, p. e00497, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2468227620302350

[4] P. Vanin, T. Newe, L. L. Dhirani, E. O'Connell, D. O'Shea, B. Lee, and M. Rao, "A study of network intrusion detection systems using artificial intelligence/machine learning," *Applied Sciences*, vol. 12, no. 22, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/22/11752

[5] M. Rigaki and S. Garcia, "A survey of privacy attacks in machine learning," *arXiv preprint arXiv:2007.07646*, 2020.

[6] C. Dwork, “Differential privacy,” in *Automata, Languages and Programming*, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–12.

[7] H. Kamel, D. Abdulah, and J. M. Al-Tuwaijari, “Cancer Classification Using Gaussian Naive Bayes Algorithm,” in *2019 International Engineering Conference (IEC)*. Erbil, Iraq: IEEE, Jun. 2019, pp. 165–170. [Online]. Available: https://ieeexplore.ieee.org/document/8950650/

[8] J. Vaidya, B. Shafiq, A. Basu, and Y. Hong, “Differentially private naive bayes classification,” in *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, vol. 1. IEEE, 2013, pp. 571–576.

[9] K. Kirasich, T. Smith, and B. Sadler, “Random Forest vs Logistic Regression: Binary Classification for Heterogeneous Datasets,” vol. 1, no. 3, p. 25, 2018.

[10] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, “Differentially private empirical risk minimization.” *Journal of Machine Learning Research*, vol. 12, no. 3, 2011.

[11] V. Kecman, *Support Vector Machines – An Introduction*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 1–47. [Online]. Available: https://doi.org/10.1007/10984697_1

[12] G. Steidl, *Supervised Learning by Support Vector Machines*. New York, NY: Springer New York, 2015, pp. 1393–1453. [Online]. Available: https://doi.org/10.1007/978-1-4939-0790-8_22

[13] L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324

[14] S. Fletcher and M. Z. Islam, “Differentially private random decision forests using smooth sensitivity,” *Expert Systems with Applications*, vol. 78, pp. 16–31, jul 2017. [Online]. Available: https://doi.org/10.1016%2Fj.eswa.2017.01.034

[15] “IDS 2017 | Datasets | Research | Canadian Institute for Cybersecurity | UNB.” [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html

[16] K. Kostas, “Anomaly detection in networks using machine learning,” *Research Proposal*, vol. 23, p. 343, 2018.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.