SANTA: Semi-supervised Adversarial Network Threat and Anomaly Detection System

Muhammad Fahad Zia*[1][0009-0002-8159-3831], Sri Harish Kalidass*[1][0009-0003-8109-8651], Jonathan Francis Roscoe[1][0009-0003-7017-2449]

[1] Future Cyber Defence, BT plc, Adastral Park, IP5 3RE, UK
muhammadfahad.zia@bt.com, sriharish.kalidass@bt.com, jonathan.roscoe@bt.com

Abstract. With the exponential increase in devices connected to the Internet, the risk of security breaches has in turn led to an increase in traction for machine learning based intrusion detection systems. These systems involve either supervised classifiers to detect known threats or unsupervised techniques to separate anomalies from normal data. Supervised learning enables accurate detection of known attack behaviours but requiring quality ground-truth data, it is ineffective against new emerging threats. Unsupervised learning-based systems address this issue due to their generalizable approach; however, they can result in a high false detection rate and are generally unable to detect specific types of each threat. We propose an ensemble technique that addresses the shortcomings of both approaches through a semi-supervised approach which detects both known and unknown threats in the network by analysing traffic metadata. The robust approach integrates A) an adversarial regularisation based autoencoder for unsupervised representation learning and B) supervised gradient boosted trees to detect the type of detected threats. The adversarial regularisation enables a reduced false positive rate and the combination of the autoencoder with the supervised stage enables resiliency against class imbalance and caters to the ever-evolving threat landscape by detecting previously unseen threats and anomalies. SANTA's ability to detect never-before-seen threats also indicates its potential to address the concept drift, a phenomenon where the known threat changes its behaviour/attack sequence over time. The system is evaluated on the CSE-CIC-IDS2018 dataset, and the results confirm the resilience and adaptability of the SANTA system against known shortcomings of both supervised and unsupervised approaches.

Keywords: Anomaly detection, Semi-supervised learning, Adversarial regularization, concept drift.

1 Introduction

Forecasts suggest that by 2025, there would be more than 75 billion devices connected to the internet – an approximate of 300% increase from the 2019 baseline [1]. This sharp increase in devices connected to the network has increased the network intrusion and cyberattack incidents across the globe. According to the report published by

*Equal Contribution

AAG [2] in 2023, there has been 125% increase in cyber-attacks in 2021. Undetected threats on a network can have severe impact on essential services and facilities for businesses; this includes loss of data, revenue, and reputation and threat to national security in the case of governments. The development of efficient intrusion detection systems has, therefore, become more important than ever in the face of evolving threat techniques.

The challenges in the domain of intrusion detection can be broadly classified into two categories; sufficient accuracy in detection of known attack patterns and the generalization ability to cater to the evolution of attacks. Signature-based intrusion detection systems address the first challenge through the use of supervised learning-based models. These models are trained on large historical datasets [3] to establish signatures or patterns of these attacks thereby enabling detection of future attacks conforming to these patterns. The problem with these models, however, is there inability to cope with new attack patterns and types as there are no available signatures to match to for these attacks. This second challenge is addressed by anomaly-based intrusion detection systems which cater to this using unsupervised learning. These systems use models that are trained to cluster data based on different criteria including similarity measures. This enables the system to separate normal and anomalous data by assigning different clusters to each. These systems are subsequently able to deal with new attacks, which would still be tagged as abnormal or anomalous since they differ from the normal behaviour or pattern. Further classification to detect specific type of attacks, however, is limited in these unsupervised learning-based systems as is their accuracy in comparison to supervised approaches. In addition to this, these systems also suffer from the known generalization problem where models misclassify malicious attacks as normal if their pattern deviates only slightly from normal behaviour.

Machine Learning (ML)-based Network threat detection systems have proven to perform better than traditional intelligence tool to protect networks against cyberattacks. The supervised tree-based classifiers and results on publicly available re-search Network dataset is discussed here (Thaseen, S.; Kumar) [5]. The unsupervised network threat and anomaly detection results are not reliable as the accuracy seems to vary from 57% to 80% and with very high false positive rate of 20% and over. (Syarif, I.; Prugel-Bennett) [6]. The promising unsupervised work identified is ARCADE (Adversarially Regularized Convolutional Autoencoder for Anomaly Detection) (Lunardi, W.T., Lopez, 2022) [4] approach. The ARCADE uses the raw packets instead of aggregated NetFlow features to train the network. In the detection stage, ARCADE uses both the encoder and decoder networks. A semi supervised approach (J. Ran, Y. Ji and B. Tang, 2019) [7] carried on Aegean Wi-Fi Intrusion Dataset (AWID) public dataset, the results outperformed other ML approaches. One of the interesting semi-supervised algorithms that was identified is XGBOD (Zhao, Y. & Hryniewicki, M. K., 2018) [8] which is an ensemble semi-supervised algorithm that was experimented on non-security datasets.

In this paper, we propose a unique combination of the aforementioned intrusion detection approaches that deals with the known challenges in intrusion detection. SANTA is a robust semi-supervised threat and anomaly detection system that enables the

classification of both known and unknown attack patterns using limited labelled data and adversarial training to reduce the false detections or misses during inference.

The contributions of the paper are: (a) the combination of adversarially regularised autoencoder to enrich data for supervised learning. (b) Evaluation of the pro-posed model in terms of accuracy on known and unknown attack types. (c) Evaluation of the proposed model in terms of resiliency when less labelled data is available for training. (d) Comparison against other known methods of anomaly detection.

The remainder of the paper is organized as follows; section 2 outlines related work; section 3 depicts the architecture of the proposed model and the methodology behind each component; section 4 outlines the dataset used in experimentation and its specifications; section 5 details the experimentation and model evaluation results; conclusions are presented in section 6.

2 SANTA

Our semi-supervised adversarial network threat and anomaly detection (SANTA) system comprises of two modules in a meta-learning pipeline where the output of first is used to enrich the input of the second. The simplified flowchart is presented in Figure 1 showing the processing pipeline and individual components. The NetFlow data is passed through the autoencoder to output the embeddings, also referred to as the newly learnt enriched features (through unsupervised representation learning). The original NetFlow is then concatenated with the enriched features to produce the enriched data. The supervised classifier is provided with the enriched data and learns to detect and identify threats and anomalies. Each of these steps are detailed in the corresponding sections below – unsupervised learning module and training strategy, supervised learning module, data enrichment and finally the inference strategy.

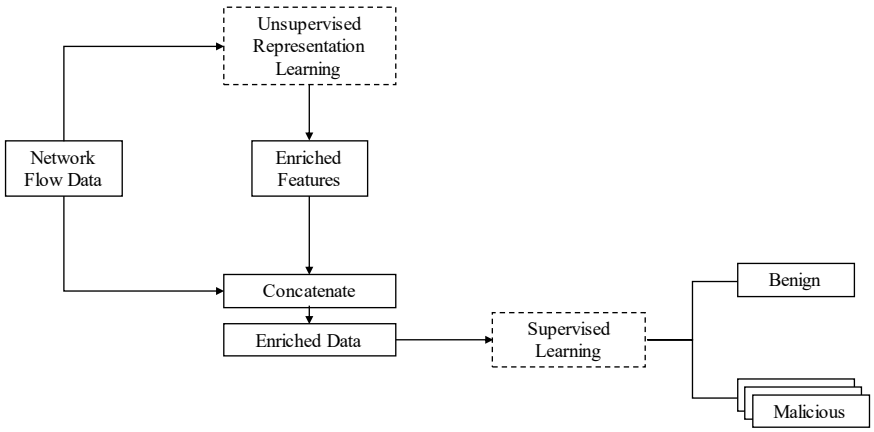

Fig. 1. Simplified flowchart showing the key components of SANTA

2.1 Unsupervised Representation Learning

We implement unsupervised representation learning using an adversarially trained autoencoder based on work published by Lunardi [4].

The autoencoder involves the use of 3 deep neural networks – **Encoder**, **Decoder** and **Critic**. The Encoder is 4-layer deep convolutional autoencoder that "encodes" the input data by learning latent features at each layer to output an encoding of the original data. This encoding captures a rich summary of the data and is used to reproduce the original data by the Decoder. The Decoder has an architecture similar to the encoder network but uses transpose-convolutions to expand the encoding in each step to accurately reproduce the original data.

These two networks are trained in tandem on solely normal (benign) traffic flows with the objective of minimizing the reconstruction error, which is the difference between the original and reconstructed data. This ensures that the network only learns to reconstruct normal traffic and not anomalies thus ensuring that the reconstruction error for anomalous flows will be high and can be used to distinguish between real and anomalous flows.

A known problem is that of generalisation; the network can be generic enough to be able to reconstruct anomalous data to sufficient quality, despite being trained on solely normal data flows thus reducing the ability of the algorithm to distinguish between anomalous and normal input data. **Adversarial regularisation-based training** using the Critic network in SANTA addresses this issue. The Critic network has a similar architecture to the encoder network with a difference in the output layer to output a single value as a score (instead of an encoding). It is trained to discriminate between reconstructed and original data by output high scores for original data and low for reconstructed data. The objective for this network is thus to maximize the difference between the scores it gives to original and reconstructed data. This strategy of training is called adversarial regularisation owing to the Encoder-Decoder and Critic network being trained with opposing objectives. This ensures that the trained network is more tightly bound to the data it is trained on (normal traffic in this case) and reduces the generic nature of the network thereby addressing the generalization problem to an extent.

The complete architecture of the unsupervised learning module is shown in Figure 2 below with further details into the precise architecture of each separate network summarized in Table 2 in the Appendix.

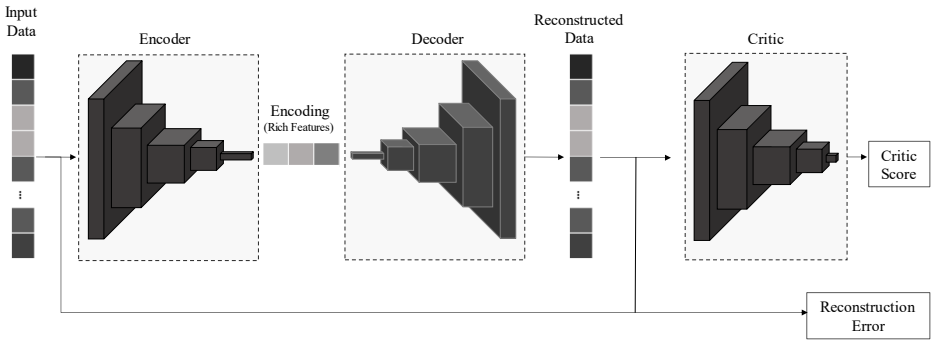

Fig 2-Architecture diagram of SANTA's unsupervised module.

Once the autoencoder is trained using the adversarial regularisation strategy, the encoder is used to encode the input data and output rich encodings that are concatenated to the original data to provide a more enriched feature space for the next stage of the SANTA pipeline. The details of the training strategy are presented below.

2.2 Training

During training, the unsupervised module is trained based on two objectives.

The first objective ($\mathcal{L}_A$) is to reduce the reconstruction error which is the $\mathcal{L}_2$ loss between the reconstructed ($\bar{x}$) and original data (x). The $\mathcal{L}_2$ loss can be expressed as:

$$\mathcal{L}_2(x, \bar{x}) = \sum_{i=0}^{F}(x_i - \bar{x}_i)^2 \quad (1)$$

where F is the total number of features in the dataset. The first objective is additionally regularized to reduce the critic score on the reconstructed data as is expressed below:

$$\mathcal{L}_A = \mathbb{E}_{x \backsim \mathbb{P}_r}[\mathcal{L}_2(x, \bar{x}) + \lambda_A C(\bar{x})] \quad (2)$$

where $\mathbb{P}_r$ is the data distribution and λ_A is a regularization coefficient. The second objective ($\mathcal{L}_B$) is to increase the squared difference between the critic score (C) on original and reconstructed data which is the adversarial regularization previously discussed. This is shown below:

$$\mathcal{L}_B = \mathbb{E}_{x \backsim \mathbb{P}_r}[(C(x) - C(\bar{x}))^2] \quad (3)$$

The three networks are thus trained in tandem with the decoder and critic essentially training the encoder which is then extracted apart and used to generate encodings on the original data to concatenate to the same and pass to the supervised module for training.

Training in the supervised module is encompassed by trees being trained to optimize the multinomial deviance and the module learning to output the correct classification for each data instance.

2.3 Data Enrichment

As mentioned above, the output from the unsupervised module, in the form of embeddings generated by the adversarially trained encoder, is used to enrich the original data; the details of which are as follows.

The data enrichment process involves exploiting the input data to a great extent to harness its potential to unprecedented levels to benefit the model development. This section describes the parameters, factors, and the process of the data enrichment stage.

The enrichment process produces newly learned representations of the original raw input data. Two forms of data that are concatenated to produce the enriched data, A) the encoded data from unsupervised stage and B) raw input data. The data enrichment is an output of meta learning process where the encoded data is produced by the en-

coding component of the autoencoder which was trained using a critic network through the aforementioned adversarial training strategy. Since the autoencoder is trained using solely benign data, the model's learning is limited to only effectively encode benign data, this limits the model's ability to encode malicious or anomalous flows. Since the encodings represents the raw data in a new latent space the encodings of malicious or anomalous data have a unique signature which distinguishes them from normal or benign data and thus helps the subsequent supervised model to detect and identify known threats, unknown threats and to some extent address the concept drift as is evident in the results from experiments documented in section 4; the specific evaluation of the data enrichment process is discussed in the section 4.2.

Delving deeper into the enrichment process it involves generating the embeddings (e) for input data (d_{input}) and transforming the embeddings by normalizing and adding weights to form the transformed embeddings (e_T) as represented in equation 4.

$$e_T = N(e) * U \tag{4}$$

where N(e) represents the normalized embeddings and U represents the weights; is a real number chosen through an empirical process. There is ongoing research to find an optimal way to produce fine-tuned weights for the embeddings.

This is followed by normalising the input data, and finally concatenating the transformed embeddings with the normalised input data to form the final enriched data (d_E) which is represented in equation 5 below.

$$d_E = e_T + N(d_{input}) \tag{5}$$

where $N(d_{input})$ represents the normalized input data.

In the IDS-2018-V2 dataset with originally 43 features, the label and the description of labels was removed, and the total number of features used was down to 41 dependent variables. The length of embeddings extracted from the unsupervised stage is 6 based on the architecture of the autoencoder finalised by an empirical process of hyperparameter tuning. The 6 real-valued embeddings, multiplying weight value (U) to the normalised embeddings and further concatenating with normalised input data containing 41 dependent variable produces 47 real-valued features, which is the size of the enriched data (d_E). This enriched data is used to train the supervised gradient boosted tree classifier.

2.4 Supervised Learning

The supervised learning algorithm used is a gradient boosted tree which offers generalisation of boosting to arbitrary differentiable loss functions, based on work by Friedman [15]. The encodings from the previous stage are concatenated to the original data to form the input for this stage of processing which takes in labelled training data and learns to output a classification on each flow of data.

The complete architecture for this supervised stage of processing is shown in Figure 3 below.

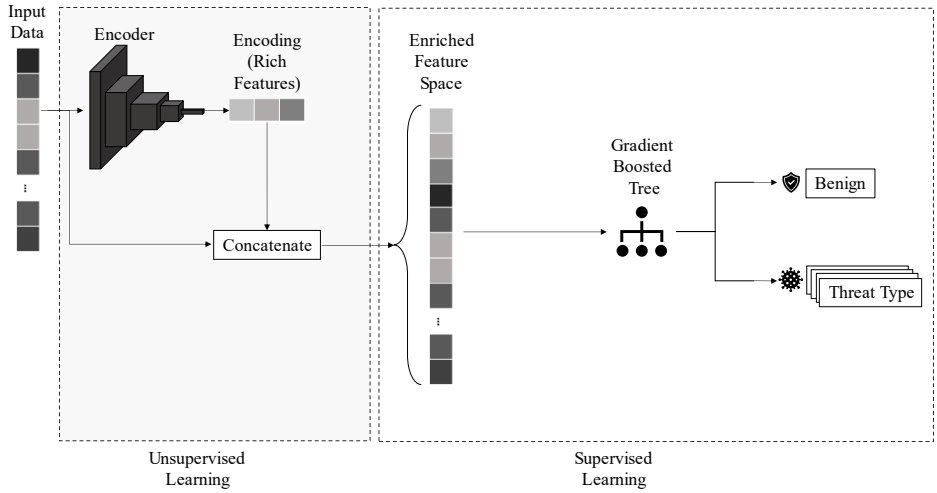

Fig 3- SANTA complete architecture (Inference)

2.5 Inference

In the inference stage, the trained encoder is used to generate encodings to enrich the data before passing it on to the supervised module for threat classification. SANTA is thus able to identify each individual class that it was previously trained on.

In the case of previously unseen data, we introduce a post processing step where the confidence scores on each class are used to determine how confident the model is in its predictions. Ideally, for previously unseen classes the model confidence scores on each known class will be low. Hence thresholding measures on the scores for each class are used to ascertain whether the input data belongs to the known classes or should be classified as a new but generic anomaly class.

For example, a data instance with a confidence score of less than 50% for all the known classes (including the *normal* class) can be considered as a new anomaly or attack pattern. The threshold is currently intuitively set at 50% using manual inspection of the results. The inference on new data is explained in further detail in section 4.2.

3 Dataset

The SANTA model was experimented using two datasets, A) an internally generated synthetic carrier data from the preconfigured network infrastructure and B) CSE-CIC-IDS2018-V2 dataset [10]. Both the datasets comprise of a form of network metadata known as NetFlow data. NetFlow Is a Network Protocol Developed by Cisco for Col-

lecting IP Traffic Information and Monitoring Network Flow. The details of the two datasets used for experimentation are as follows.

3.1 Synthetic Carrier dataset

The initial analysis and testing of the SANTA model was carried using the synthetic carrier data. The synthetic carrier data refers to the aggregated NetFlow generated from a custom experimental setup. The data itself is confidential and therefore we have limited the discussions and the results to appropriate levels and focusing more on the CSC-IDS-2018-V2 dataset-based results.

The synthetic carrier data is a small dataset that comprises of benign and port-scan activity type flows. Port scan activity consists of a variety of techniques that aim to discover information about networks and hosts. The discovered vulnerabilities could be exploited in a future attack which could have severe consequences. Port scans may be indicative of reconnaissance by threat actors, but they are a common activity for security teams to assess and monitor networks.

The dataset consists of raw NetFlow containing predefined set of features and a time windowed pre-processing technique is applied to extraction 17 dependent variables. Depending on the value set for time (t, (mins)) in time windowing the volume of extracted flows are generated. The lower the time(t) value, higher number of extracted flows and vice versa, however at any instance time(t) can only be positive and the total number of extracted flows can't exceed the volume of original raw NetFlow. In our experiment, a time(t) value of 30(mins) is applied on the raw NetFlow data. The timestamp is used for each observation in the dataset to partition group the data by specified time intervals, which can be assumed as time frequency-based aggregation of the data. This aggregated data is used to extract some features based on pre-defined empirical relations, which results in the final processed dataset. The table below illustrates the pre-processing description.

Table 1. Synthetic carrier dataset description

Data Type	Raw NetFlow	Time windowed (Mins)	Extracted Flows	Data Ratio
Benign	430,437	30	9962	89%
Port Scan attack	313,770	30	1167	11%

A snapshot of the list of extracted flow variables are furnished below.
{'ip_addresses,'syn_flag_count','dst_port_count','dst_srv_port_count','dst_ip_count',
'proto_count','tcp_proto_count','udp_proto_count','icmp_proto_count',
'tcp_proto_ratio','udp_proto_ratio','icmp_proto_ratio','reply_count','reply_ratio',
'mean_packets', 'max_packets', 'mean_bytes', 'max_bytes','n_flows', 'duration'}

3.2 CSE-IDS-2018-V2 dataset

Further experimentation was carried out with the CSE-CIC-IDS2018 dataset [10]. The creators evaluated the shortcomings of the eleven publicly available datasets since 1998 and came up with a dataset to address those. It conforms to each of the eleven criteria of the last intrusion detection dataset evaluation framework [12] which none of the other datasets could completely meet. More details on the dataset creation are available in [13]. The version used for evaluation takes the original .pcap files from this dataset to generate NetFlow-based data and is called NF-CSE-CIC-IDS2018.

The NF-CSE-CIC-IDS2018 consists of, in addition to a benign or normal class, six different common update-to-date attacks which conform to real world criteria; we trained and validated our model on a subset of four commonly occurring attack scenarios out of the six – botnet, brute force attacks, DDoS (Distributed Denial of Service) and infiltration attacks. We included a fifth web attacks category in the test set to evaluate model performance on previously unseen data. Each row of data contains aggregated statistics on each flow of packets in the network in the form of 41 features. The number of flows or occurrences for each type of attack used in our experiments is given in Table 2.

Table 2. Dataset classes and corresponding number of occurrences

	Class	# of occurrences
1	Normal	998,135 (62.63%)
2	Botnet	143,097 (8.98%)
3	Brute Force	120,912 (7.59%)
4	DDoS	211,607 (13.28%)
5	Infiltration	116,361 (7.30%)
6	Web Attacks	3,502 (0.22%)

4 Evaluation

The NF-CSE-CIC-IDS2018 dataset was split into two sets with 70% of data used for cross-fold validation and 30% used as a separate evaluation set containing an additional class of web attacks that is not included in the first set in order to evaluate the model on new and evolving threats.

For evaluation purposes, we use the three common information retrieval metrics:

- **Precision (Pr)** – ratio of correctly classified attack flows (true positives - TP) and total classifications (sum of true positives and false positives - FP).
- **Recall (Re)** – ratio of correctly classified attack flows (TP) and all flow instances (sum of true positives and false negatives - FN).
- **F-Measure (F1)** – the harmonic combination of precision and recall values.

The three measures are calculated as shown in equation 4.

$$Pr = \frac{TP}{TP+FP}, Re = \frac{TP}{TP+FN}, F1 = \frac{2}{\frac{1}{Pr}+\frac{1}{Re}} \quad (4)$$

4.1 Comparison with other models

SANTA was then trained and evaluated using the first set of data using 5-fold cross validation technique. This technique involves splitting of the dataset into 5 folds with a different combination of 4 sets to train and 1 to validate at each run. The results are then averaged across each combination to ensure a report with less bias. The technique is visualized in the Figure 4 below showing the data split into 5 folds and each of the 5 runs with different combination of train and test set.

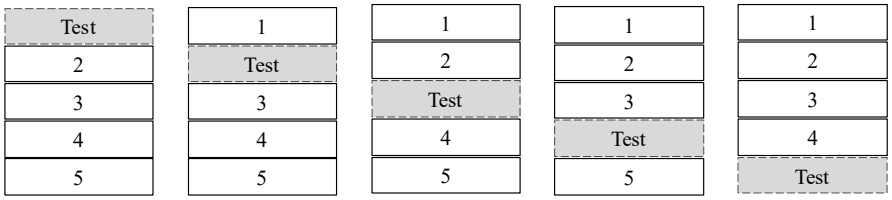

Fig 4-Cross validation technique with 5 folds.

The results are compared to those from known supervised models and the results are shown in Table 3 below.

Table 3. Results on NF-CIC-IDS2018 dataset

Model	Precision	Recall	F1-Score
SANTA	**0.98**	**0.84**	**0.86**
Gradient Boosting Classifier	**0.98**	**0.84**	**0.86**
Random Forest Classifier	**0.95**	**0.87**	**0.89**
Linear SVC	0.73	0.44	0.43
Logistic Regression	0.68	0.43	0.43

The initial results demonstrate that the SANTA model performs competitively with Random Forest and Gradient Boosted Trees on the dataset in terms of classifying known attacks.

4.2 New attack scenarios

The separated test set with the additional unseen class of web attacks was then used to evaluate the SANTA model against the top contenders from the cross-validation stage – Gradient Boosting Classifier and Random Forest Classifier. Table 2 also shows the attack classes present in the set and the results in Table 4 show the precision and recall values on each class.

Table 4. Comparison with other models on seen and unseen test data

	Precision						**Recall**					
Model	1	2	3	4	5	6	1	2	3	4	5	6
SANTA	**0.91**	1.00	0.69	1.00	0.84	0.10	0.94	0.94	1.00	0.40	0.00	**0.85**
Random Forest Classifier	**0.93**	1.00	1.00	0.96	0.92	0.00	1.00	1.00	1.00	1.00	0.33	**0.00**
Gradient Boosting Classifier	**0.92**	1.00	1.00	0.95	0.98	0.01	1.00	1.00	0.98	1.00	0.22	**0.00**

The results are calculated based on confidence thresholds on the output of each model. If a predicted class has a confidence (or probability) of less than 0.5, it is classified as an anomaly or a previously unseen attack pattern. The threshold of 0.5 as mentioned previously is currently set intuitively after manual inspection of results. All the predicted web attacks also form part of this category. In the cyber security scenario and with the detection logic that is in place for this experiment, the two values of note for each model are the precision on the 1st class (normal instances) and the recall on the 6th class (anomalies or unseen attacks). This is because of our two objectives. Firstly, we want to reduce the instances that are falsely classified as normal as this can lead to threats going through to a system undetected; these false positives are captured by the precision value on the first class. Secondly, we want to ensure that there are no unseen threats that are missed by the model; these 6th class false negatives are reflected in the corresponding recall value. The results indicate SANTA is competitive in reducing undetected threats and the ability to detect new threats with significant consistency.

This experiment evaluates the effectiveness of the data enrichment using autoencoder which sets the SANTA model apart. The web attacks data are a new type of attack which was never used for training or validation process. When the web attack data was used for testing the ability of the models to detect an unknown threat, SANTA had a remarkable 0.85 recall whereas other models had a recall value of 0.00 (refer to Table 4). This is because the enriched data used in SANTA model contains the encodings generated by the pruned encoder component of the unsupervised model.

It must be noted that the supervised training process utilises the enriched data, which forces the model to use encodings to detect and identify threats. However, in the case of a new threat (unknown threat) faced by the SANTA model, the unsupervised component containing the encoder produces ineffective encodings which subsequently set this data apart from previously seen (normal) data. These are passed on to the supervised classifier, concatenated with the raw data, to classify the threat. Since

the encodings are significantly different, the deviant signature is detected by the supervised classifier which classifies the input data as an anomaly.

4.3 Synthetic Carrier dataset-based evaluation

This section discusses the results produced using the synthetic carrier dataset, where a few selections of supervised and semi-supervised models were used for the benchmarking experiment. The graph below shows the widely opted machine learning model evaluation metrics precision, recall and F1-score on a range of models. The XGBOD-41[8] is a semi-supervised extreme gradient boosted outlier detection model comparing of multiple outlier detection algorithms (KNN, HBOS etc.) stacked in various combination of hyper parameters to include 41 outlier models. Similarly, for XGBOD-25 and XGBOD-15, where the number of outlier models are reduced to 25 and 15 respectively using manual selection process. The XGBOD-41 has the highest precision score reaching 88%, followed by SANTA model reaching 87%. The SANTA models top the recall with 83% and F-1 Score of 85%.

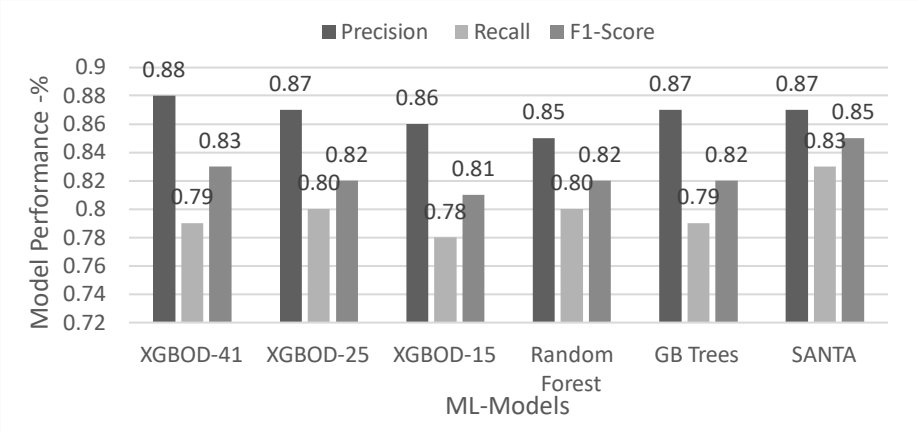

Fig 5-Synthetic Carrier NetFlow data -Classification results

It must be noted that the XGBOD model is computationally expensive for training and inference. For example, the SANTA is 18 times faster compared to XGBOD during the inference and this makes the SANTA an optimal solution for real-time high-volume deployments with minimal computational requirements, even edge deployments.

The initial results also indicate the SANTA's resiliency to increase in quantities of training data. The graph below compares the performance of SANTA model and random forest on 30% of training data and 100% of training data. The results indicates that the SANTA models trained on just 30% of total training data outperforms the

random forest model trained on 100% of training data. This also indicates the low quantity ground truth training data requirement for SANTA model.

5 Conclusion

This paper introduced a new approach to network-based threat detection utilising semi-supervised learning with a combination of unsupervised representation learning and supervised learning.

The use of adversarial regularisation in the unsupervised module to train the auto-encoder allows the model to cater to the known generalization problem with increased number of false positives in unsupervised approaches. The results on the NF-CIC-

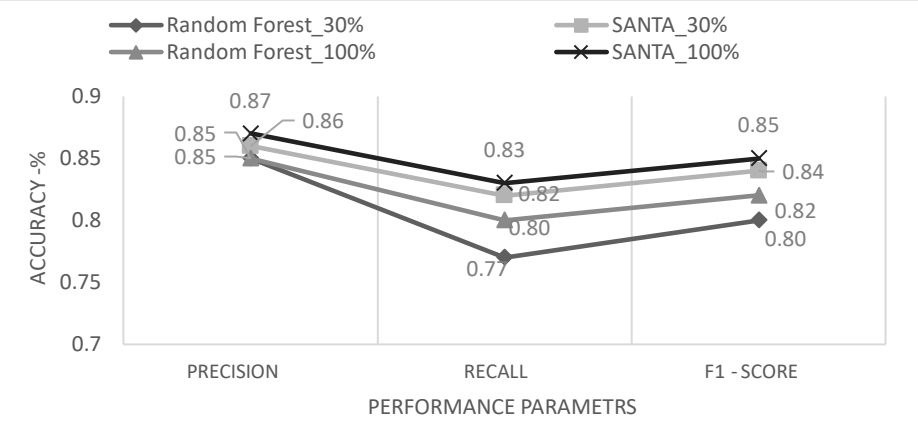

Figure 6-30% labelled data Vs 100% labelled data model performance

IDS2018 dataset corroborate the hypothesis with a high precision value on *normal* class indicating small number of false positives.

The combination of the enriched feature space from unsupervised module with original data allows the model to perform extremely well on unseen data of new attack patterns. The results on a test set including a new attack type show the ineffectiveness of supervised techniques such as the Random Forest on new and evolving attacks, while the SANTA model showed a remarkable recall rate on new data. This makes the SANTA very relevant in the cybersecurity domain especially where new threats are constantly cropping up and existing threats are evolving daily. Most importantly, the known threats tend to evolve over time (concept drift) to fool the security systems with new patterns of attack, SANTA model's initial analysis indicates it potential to detect such change in behaviours over time.

Future work is needed research into improving the ability of the unsupervised module in enriching the dataset and further development of the supervised algorithm's detection logic to improve accuracy and robustness on new and existing attack pat-

terns. There is also scope for empirical investigation into the threshold value that is set for interpretation of classification results.

Appendix

Table 6. Encoder, Decoder and Critic Architecture.

Encoder			
Layer	Kernel, Stride	Output	Parameters
Input	–	1 × 41	
Convolution	4, 2	16 × 21	80
Leaky ReLU	–	–	–
Batch Normalization	–	–	64
Convolution	4, 2	32 × 11	2,080
Leaky ReLU	–	–	–
Batch Normalization	–	–	128
Convolution	4, 2	64 × 6	8,265
Leaky ReLU	–	–	–
Batch Normalization	–	–	256
Linear	–	6	2,310
Total			**13,174**

Decoder			
Layer	Kernel, Stride	Output	Parameters
Input	–	6	
Linear	–	64 × 6	2,688
Transpose Convolution	4, 2	32 × 11	8,224
ReLU	–	–	–
Batch Normalization	–	–	128
Transpose Convolution	4, 2	16 × 21	2,064
ReLU	–	–	–
Batch Normalization	–	–	64
Transpose Convolution	4, 2	1 × 41	65
Sigmoid	–	–	–
Total			**13,233**

Critic			
Layer	Kernel, Stride	Output	Parameters
Input	–	1×41	
Convolution	4, 2	16×21	80
Leaky ReLU	–	–	–
Batch Normalization	–	–	64
Convolution	4, 2	32×11	2,080
Leaky ReLU	–	–	–
Batch Normalization	–	–	128
Convolution	4, 2	64×6	8,265
Leaky ReLU	–	–	–
Batch Normalization	–	–	256
Linear	–	6	4,230
Leaky ReLU	–	–	–
Layer Normalization	–	–	12
Linear	–	1	7
Total			**15,113**

References

1. Statistica Research Dept.: Internet of things - Number of Connected Devices Worldwide 2015–2025, https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide (2022).
2. Charles G., AAG IT: The Latest Cybercrime Statistics, https://aag-it.com/the-latest-cyber-crime-statistics (2023).
3. Bingdong Li, Jeff S., George B., Mehmet H. G.: A survey of network flow applications. Journal of Network and Computer Applications 36(2), 567–581 (2013).
4. Lunardi, Willian T., Martin A. L., Jean-Pierre G.: Arcade: Adversarially regularized convolutional autoencoder for network anomaly detection. IEEE Transactions on Network and Service Management (2022).
5. Thaseen, S.; Kumar, C.A. An Analysis of Supervised Tree Based Classifiers for Intrusion Detection System. In Proceedings of the 2013 International Conference on Pattern Recognition, Informatics and MOBILE Engineering, Salem, India, 21–22 February 2013; pp. 294–299
6. Syarif, I.; Prugel-Bennett, A.; Wills, G. Unsupervised Clustering Approach for Network Anomaly Detection. In Proceedings of the International Conference on Networked Digital Technologies, Dubai, United Arab Emirates, 24–26 April 2012; Springer: Berlin/Heidelberg, Germany, 2012;
7. J. Ran, Y. Ji and B. Tang, "A Semi-Supervised Learning Approach to IEEE 802.11 Network Anomaly Detection," 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring), Kuala Lumpur, Malaysia, 2019, pp. 1-5, doi: 10.1109/VTCSpring.2019.8746576.

8. Zhao, Y. & Hryniewicki, M. K. (2018), XGBOD: Improving Supervised Outlier Detection with Unsupervised Representation Learning., in 'IJCNN' ,IEEE, (https://arxiv.org/abs/1912.00290).
9. Jerome H. Friedman: Greedy function approximation: A gradient boosting machine. The Annals of Statistics, 29(5) 1189-1232 (2001).
10. A Realistic Cyber Defense Dataset (CSE-CIC-IDS2018), https://registry.opendata.aws/cse-cic-ids2018, last accessed on 2023/06/01.
11. Shiravi A, Shiravi H, Tavallaee M, Ghorbani A. A.: Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Computers Security 31(3):357–74 (2012).
12. Gharib A., Sharafaldin I., Habibi Lashkari A., Ghorbani A. A.: An evaluation framework for intrusion detection dataset. International Conference on Information Science and Security (ICISS), 1–6. (2016).
13. Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSp, 1, 108-116 (2018).