Ensemble-learning Approaches for Network Security and Anomaly Detection

Juan Vanerio
AIT Austrian Institute of Technology & Universidad de la República
jvanerio@fing.edu.uy

Pedro Casas
AIT Austrian Institute of Technology
pedro.casas@ait.ac.at

ABSTRACT

The application of machine learning models to network security and anomaly detection problems has largely increased in the last decade; however, there is still no clear best-practice or silver bullet approach to address these problems in a general context. While deep-learning is today a major breakthrough in other domains, it is difficult to say which is the best model or category of models to address the detection of anomalous events in operational networks. We present a potential solution to fill this gap, exploring the application of ensemble learning models to network security and anomaly detection. We investigate different ensemble-learning approaches to enhance the detection of attacks and anomalies in network measurements, following a particularly promising model known as the Super Learner. The Super Learner performs asymptotically as well as the best possible weighted combination of the base learners, providing a very powerful approach to tackle multiple problems with the same technique. We test the proposed solution for two different problems, using the well-known MAWILab dataset for detection of network attacks, and a semi-synthetic dataset for detection of traffic anomalies in operational cellular networks. Results confirm that the Super Learner provides better results than any of the single models, opening the door for a generalization of a best-practice technique for these specific domains.

The research leading to these results has been partially funded by the Vienna Science and Technology Fund (WWTF) through project ICT15-129, "BigDAMA".

1 INTRODUCTION

Network security and anomaly detection represent both a keystone to ISPs, who need to cope with an increasing number of unexpected events that put the network's performance and integrity at risk. The high-dimensionality of network data provided by current network monitoring systems opens the door to the massive application of machine learning approaches to improve the detection and classification of anomalous events. However, selecting the best machine learning model for a specific problem is a complex task - it is commonly accepted that there is no silver bullet for addressing different problems simultaneously. Indeed, even if multiple models could be very well suited for a particular problem, it may be very difficult to find one which performs optimally for different data distributions and statistical mixes. The ensemble learning theory permits to combine multiple models to form a (hopefully) better one. Ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone. In principle, if no single model covers the true prediction behind the data, an ensemble can give a better approximation of that oracle, true prediction model. In addition, an ensemble of models exhibits higher robustness with respect to uncertainties in training data, which is highly beneficial.

Ensemble learning has in principle a higher computational cost and complexity than single based learning approaches. Nevertheless, current big data platforms and off-the-shelf data processing technology is mature enough to allow a fast and parallel operation of multiple algorithms [2], easing this constraint.

In this paper we devise a novel detection technique for network security and anomaly detection using the Super Learner ensemble learning model [1]. The Super Learner is a supervised learning method that finds the optimal combination of a collection of base prediction algorithms. The Super Learner performs asymptotically as well as the best possible weighted combination of the base learners, providing a very powerful approach to tackle multiple problems with the same technique. In addition, it defines an approach to minimize over-fitting likelihood during training, using a variant of cross-validation.

The proposed solution is evaluated on two different scenarios and using five different ensemble combination algorithms for the Super Learner, using the well-known MAWILab dataset for detection of network attacks, and a semi-synthetic dataset for detection of traffic anomalies in operational cellular networks. Evaluations confirm that ensemble techniques have the ability to perform as well as the

best available base learning model, achieving even better results in most scenarios and using different combination approaches. To the best of our knowledge, this paper is the first attempt to apply the Super Learner approach to network anomaly detection problems, aiming at discovering a new category of machine learning models which could be applied in a more systematic fashion to networking problems. We believe that this study would enable a broader application of ensemble learning approaches to network security and anomaly detection, with very promising results.

The remainder of this paper is organized as follows: Sec. 2 briefly reviews the related work. Sec. 3 describes the main concepts behind the Super Learner ensemble approach, and presents the different learning models used in the study, both at the individual level and at the ensemble level. Sec. 4 reports benchmarking results for the proposed ensemble learning approaches, comparing their performance to that achieved by the individual models in the detection of both network attacks and network traffic anomalies in two distinct datasets. Finally, Sec. 5 concludes this work.

2 RELATED WORK

There are a couple of extensive surveys on general domain anomaly detection techniques [11] as well as network anomaly detection [12, 13], including machine learning-based approaches. The application of learning techniques to the problems of network security and anomaly detection is largely extended in the literature. There is a particularly extensive literature in the application of learning-based approaches for automatic traffic analysis and classification. We refer the interested reader to [10] for a detailed survey on the different ML techniques applied to automatic traffic classification. The specific application of ensemble learning approaches to network security and anomaly detection is by far more limited, and even if it is generally observed in the practice that ensembles tend to yield better results when there is a significant diversity among the models, only few papers have applied them to network security [15] and network anomaly detection [14].

3 ENSEMBLE LEARNING

In the context of supervised learning there are several methods to train algorithms with available data to use them for prediction purposes. The performance of a particular algorithm or predictor depends on how well it can assimilate the existing information to approximate the oracle predictor, i.e. the ideal optimal predictor defined by the true data distribution. However, knowing a priori which algorithm will be the best suited for a given problem is almost impossible in practice. One could say that each algorithm learns a different set of aspects of reality from the training datasets, and then their respective prediction capability also differs between problems.

According to [7], single hypothesis algorithms or simple learning models may suffer from three different bottlenecks: **statistical problem** - arises when the space of hypothesis is too large for the amount of available training data, resulting in several algorithms with similar accuracy and risk of choosing one that will not predict future data points well; **computational problem** - several algorithms are not guaranteed to find the global optimum; **representation problem** - results from the hypothesis space not containing any model that is a good approximations of the true distribution.

Rather than finding the best model to explain the data, ensemble methods construct a set of models and then decide between them with some combinatorial approach, seeking complementarity in the sense that the learning limitations of each predictor compensates for the others. Thus, the execution of several of these algorithms in parallel provides diversity of predictions. Several papers have studied methods exploiting this diversity to enhance the overall prediction capability by combining the outputs of multiple algorithms [5, 7]. Essentially, this is made by using a scheme known as *Ensemble Learning* that uses the output of the predictors as inputs for a new algorithm that receives the name of second level or *meta learner*. A notable example of this approach is the well known Random Forest algorithm. Classical ensemble learning approaches include bagging, boosting, and stacking [7].

General ensemble learning approaches might be prone to over-fitting the data. In [1] a simple ensemble learning algorithm named *Super Learner* is proposed as a possible solution for this over-fitting limitation. It proposes a method to minimize the over-fitting likelihood using a variant of cross-validation. In addition, the Super Learner provides performance bounds, as it performs asymptotically as good as the best available single hypothesis predictor.

The Super Learner algorithm makes aggressive use of cross validation: the available labeled dataset consisting of n samples is split in K approximately equal sets. As usual, each of these sets is used as a *validation set*, while its complement, $K-1$ sets are used as the *training set*. For each split, the J first level learners are fitted with the training dataset and then do predictions for the samples of the validation set. By merging the predictions done for every fold we obtain a new dataset Z of size $n \times J$, containing the predictions done by each first level learner for every sample in the disjoint validation sets. This new dataset Z is used as design input matrix to train the meta learner algorithm, which will then be used to perform the final predictions. In the original paper [1], the meta learner can be arbitrarily complex, yet a simple linear regression model is used for the presented regression scenario. The paper presents a formal proof showing that this Super Learner is optimal in the sense that it can perform at least asymptotically as well as the best first level learner available.

3.1 Binary Classification

The logic expressed in [1] can be adapted for use on binary classification problems such as the one we tackle in this paper. Essentially, suppose there are n i.i.d. observations $(X_i, y_i) \sim P_0$ with $i = 1, \ldots, n$ that generate empirical probability distributions P_n, and that the goal is to estimate the classification function ψ_0 such that:

$$\psi_0(X) = \arg\min_{\psi \in \Psi} E\left[L\left(y, \psi(X)\right)\right] \tag{1}$$

where $L\left(y, \psi(X)\right)$ is a given loss function that measures the discrepancy between prediction and real value - e.g., square loss in [1], for all possible feature vector $X \in \mathcal{X}$ and its corresponding label $y \in \mathcal{Y}$. ψ_0 is then a mapping function from the feature space into the label space and Ψ the parameter space of all possible functions such that $\mathcal{X} \rightarrow \mathcal{Y}$. Now let $\{\hat{\psi}_j\}$ $j = 1, \ldots, J$ be the collection of first level learners, which represent mappings from the empirical distribution P_n into parameter space Ψ.

When using K-fold cross-validation let $k \in \{1, \ldots, K\}$ be the index of a split of the data into a validation set $V(k)$ and its complement, the training set $T(k)$. Let then $k(i)$ be the split index in which sample i belongs to the validation set, i.e. $i \in V(k(i))$ and $f_{j,T(k)}$ the realization of the j^{th}-first level learner $\hat{\psi}_j$ after being trained in $T(k)$ - assuming the training has as target the minimization of the expected risk $E[L(y, \hat{\psi}_j(X))]$. Then, a new observations dataset $Z = \{(Z_i, y_i)\}$ is constructed such that the i^{th}-sample $z_i = \left\{f_{j,T(k(i))} : j = 1, \ldots, J\right\}$ is the vector of the predictions of the J first level learners for sample i when sample i is not in the training dataset.

The last input for the Super Learner algorithm is another user defined algorithm $\phi : \{\mathcal{Y}\}^J \rightarrow \mathcal{Y}$, that shall be used as a predictor for labels $y \in \mathcal{Y}$ from data points $z \in \{\mathcal{Y}\}^J$. This algorithm must also be trained to minimize the expected risk in a similar fashion to the first level learners, that is to become similar to the optimal mapping:

$$\phi^*(Z) = \underset{\phi \in \Phi}{\arg\min} E\left[L\left(Y, \phi(Z)\right)\right] \tag{2}$$

over the set Φ of functions $\{\mathcal{Y}\}^J \rightarrow \mathcal{Y}$. Although not the case presented in [1], this fitting can be made using penalization or cross-validation to further avoid over-fitting. Let then $g : \{\mathcal{Y}\}^J \rightarrow \mathcal{Y}$ be the function obtained from fitting algorithm ϕ with training dataset $\{Z_i\}$ and label set $\{y_i\}$.

Once g has been determined, the first level learners are re-trained on the whole available training dataset to obtain the fitted predictors $\{f_j : j = 1, \ldots, J\}$. Thus, the Super Learner algorithm becomes a new algorithm S such that:

$$S(X_i) = g\left(f_1(X_i), \ldots, f_J(X_i)\right) \tag{3}$$

As a final note, the outputs of the first level learners and the Super Learner can be categorical in case of a *hard decision* or a score in a *soft decision* case. The latter is more expressive, as it provides an extra degree of freedom in the selection of the decision threshold and allows for performance descriptions such as *Receiver Operation Characteristic (ROC)* curves. Thus, we decided to use as output from each algorithm the probability of the evaluated sample belonging to the "positive" class (i.e., detection of an anomaly); as such, the elements of matrix Z represent probabilities and, similar to the most generic case for X, are also continuous values.

3.2 First Level Learners

Ensembles of machine learning models tend to yield better results when there is a significant diversity among the individual base models. Therefore, we select an assorted group of base learning models with very different underlying data assumptions. In particular, we select the following five standard, fully-supervised models [10]: (i) SVM with linear kernel, (ii) decision trees (CART); K-NN with direct majority voting, using $K = 10$, (iv) multi layer perceptron neural network, and (v) naive Bayes. Most of these models have already shown good performance in previous work on anomaly detection and classification [3, 4]. The hyper-parameter configuration values for each model are selected on a manual basis, both by trial and error as well as by following default recommended settings for the python *scikit-learn* library used in all the implementations. These models are trained according to the previously described procedure, first to create the matrix Z, and then re-trained on the whole available training dataset to make predictions on the testing dataset. In the evaluations, we compare the individual performance of each of these base models to the performance achieved by the devised super learners algorithms, described next.

3.3 Super Learner Algorithms

The original work [1] uses a simple minimum square linear regression as the example Super Learner. Following the Super Learner logic for binary classification described in Sec. 3.1, we conceived five different Super Learner algorithms. As we are dealing with binary classification problems, a first natural choice is the usage of logistic regression, which shall be the first evaluated Super Learner.

In [7], a linear weighted algorithm is suggested as meta-learner for ensemble learning, by taking predictions from each first-level learner and weighting them to get a weighted-majority-voting-like classifier; more concrete, let $H(X) = \sum_{j=1}^{J} w_j h_j(X)$ be the weighted sum of the individual first-level learner predictions $h_h(X)$, the algorithm decides for the positive class if $H(X) > \beta$, being β the decision threshold, or the negative class otherwise. The weights w_j can be defined in different ways; in this work we use three different types of weights:

MVuniform: gives the same weight $(1/J)$ to each learner, implementing simple majority voting.

MVaccuracy: assigns weights $w_j = \dfrac{\alpha_j}{\sum_{i=1}^{J} \alpha_i}$ to the prediction of learner j, being α_i the *accuracy* of the learner - i.e., the fraction of true classifications achieved on the whole available training dataset.

MVexp: computes weights with an exponential classification accuracy, $w_j = \dfrac{e^{\lambda \alpha_j}}{\sum_{i=1}^{J} e^{\lambda \alpha_i}}$, where λ is selected to reduce the influence of low accuracy predictors - we take $\lambda = 10$ for such an effect.

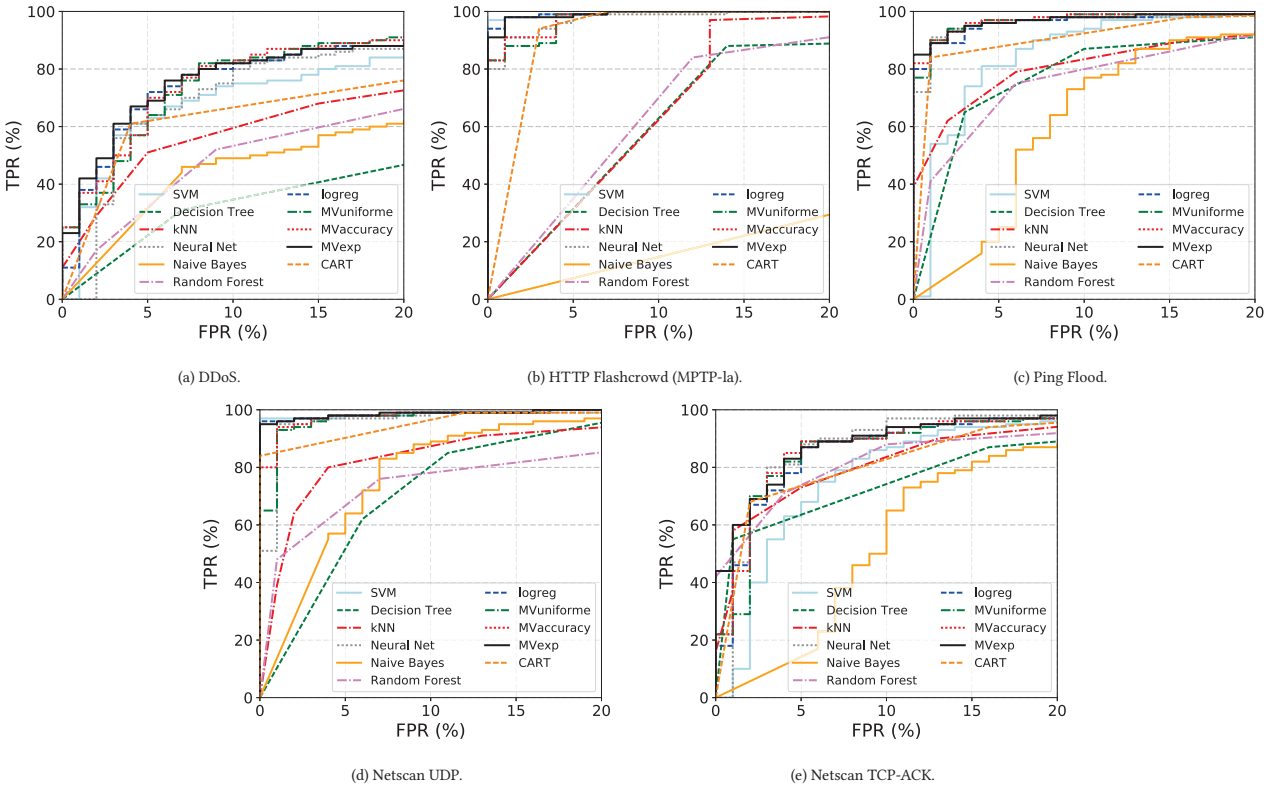
Finally, [1] mentions that there is no need to restrict the Super Learner algorithm to parametric regression or classification fits. For example, one could define it in terms of a particular machine learning algorithm. To also test this direction, we devise another Super Learner based on a simple decision tree model, using the well known CART decision tree algorithm.

4 EVALUATION AND DISCUSSION

In this section we show that the Super Learner approach can enhance the results obtained on the binary classification problems studied in [3] and [4] for detection of network attacks and anomalies respectively. In those studies, different standard first-level learning models are used. An exception to this is the usage of Random Forests (RF) [10], which actually represent an ensemble learning approach, by combining the output of multiple decision trees through plain majority voting. In a nutshell, each decision tree of a RF is built on different subsets of input features, randomly selected. The RF algorithm is more sophisticated than the Super Learner itself, in the sense that it already performs feature selection during training.

Table 1: ROC AUC on MAWI dataset.

	DDoS	mptp-la	netscan-ACK	netscan-UDP	ping-flood
Decision Tree	0.731	0.865	0.920	0.930	0.929
Naive Bayes	0.760	0.657	0.881	0.938	0.901
Neural Net	0.911	0.994	0.967	0.986	0.988
SVM	0.898	**0.997**	0.943	0.995	0.968
kNN	0.840	0.919	0.951	0.954	0.951
Random Forest	0.821	0.914	0.945	0.918	0.930
logreg	0.924	**0.998**	0.965	**0.996**	0.991
MVaccuracy	**0.928**	0.994	0.967	0.993	**0.992**
MVexp	**0.928**	**0.997**	**0.970**	**0.996**	**0.992**
MVuniforme	0.927	0.994	0.966	0.992	0.991
CART	0.879	0.984	0.946	0.983	0.977



(a) DDoS. (b) HTTP Flashcrowd (MPTP-la). (c) Ping Flood.

(d) Netscan UDP. (e) Netscan TCP-ACK.

Figure 1: Detection performance per type of attack on the MAWI dataset. Except for the CART-based Super Learner, all Super Learners generally outperform base predictors.

We therefore compare the performance achieved by each single, first-level detector against that achieved by the proposed Super Learners. To have comparable results to [3] and [4], we additionally add a RF-based detector, trained on the same training set as the rest of the models, and having as many internal decision trees as first level learners has the Super Learner; i.e., 5 in this paper. Comparison is performed on the basis of true positive vs false alarm rates (TPR/FPR respectively) through ROC curves, as well as by computing the area under these ROC curves (AUC).

4.1 Data Description

Two different datasets are used to test the performance of the proposed approaches: the MAWI dataset for network security [9], and a semi-synthetic dataset for traffic anomalies in cellular networks, conceived in [4]. Each dataset is split in two sets: a training set, with approximately 20% of the samples, and a testing set with the remainder 80%. We perform the learning procedures on the training set and then evaluate the performance of the predictors on the testing set using ROC curves and AUC values.

4.1.1 MAWI Dataset. MAWI is a public collection of 15-minute real network traffic traces captured every day on a backbone link between Japan and the US since 2001. Building on this repository, the MAWILab project uses a combination of four traditional anomaly detectors to partially label the collected traffic [9]. From the labeled anomalies and attacks, we focus on a specific group which are detected simultaneously as "anomalous" by the four MAWILab detectors to achieve a high quality on the obtained labels. We consider five types of attacks/anomalies in particular: (i) DDoS attacks (DDoS), (ii) HTTP flashcrowds (mptp-la), (iii) Flooding Attacks (Ping Flood), (iv) UDP and (v) TCP probing traffic. The considered algorithms were trained to detect each of these attack types independently and in parallel, in the same fashion and using the same features as [3]. As a result, each detection approach can detect the occurrence of an attack and also classify its nature. The dataset spans a full week of MAWILab traffic traces collected in late 2015; traces are split in consecutive time slots of one second each, and a high-dimensional set of 245 features describing the traffic in each of these slots is used, see [3] for more details.

4.1.2 Semi-Synthetic Anomalies Dataset. We also use a semi-synthetic dataset of network anomalies observed in cellular networks, conceived in [4] by using real DNS traffic measurements. After collecting DNS traces for longer than six months in 2014 at a cellular network of a large-scale European operator, the authors used a technique to generate new traffic traces by carefully recombining real traffic traces. Basically, they considered samples of manually labeled one-minute intervals from the original data, characterized by a vector of features containing the distribution of DNS query counts by device Manufacturer, device OS, APN, FQDN and DNS transaction flag. With the anomaly-free intervals they generate new synthetic background traffic, simply by shuffling the data samples of the same time of the day and same day class (working or festivity). Then, three different types of anomalies are introduced into the synthetic data, derived from real anomalies observed in this operational network. These anomalies mimic different types of service outages, and are represented by impacting a different number of end-users requesting particular services on specific domain names. The different anomalies considered are E1: short lived (hours) high intensity anomalies (e.g., 10% of devices repeating a request every few seconds), where the involved devices share the same manufacturer and OS; E2: several days lasting low intensity anomalies (e.g., 2% of devices repeating requests every few minutes) and E3: short-lived variable intensity anomalies affecting all devices of a specific APN. The used dataset consists of a full month of synthetically generated measurements, reported with a time granularity of 10 minutes time bins. Each time bin is assigned a class, either normal (label 0) or anomalous (label 1, 2 or 3 for the three anomaly types respectively). The dataset includes 16 different variations of E1, E2 and E3 anomalies, impacting a different fraction of end-users - going from 0.5% to 20%. Full details on the synthetic traffic generation are available in [4].

4.2 Detection of Network Attacks

Fig. 1 depicts the ROC curves obtained by each detector and for each attack type, and Tab. 1 reports the corresponding AUC values.

Table 2: ROC AUC on semi-synthetic dataset.

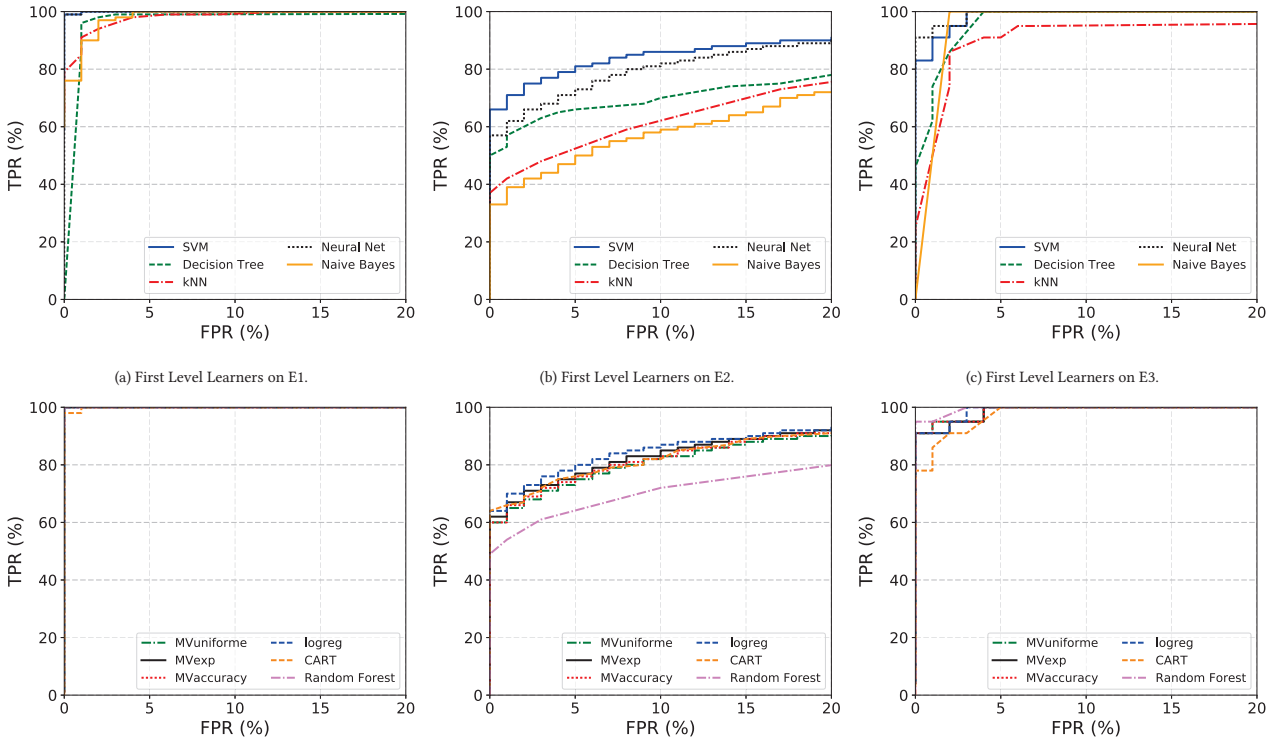
	E1	E2	E3
Decision Tree	0.993	0.873	0.991
Naive Bayes	0.996	0.861	0.989
Neural Net	**0.999**	0.944	0.996
SVM	**0.999**	0.944	0.995
kNN	0.995	0.859	0.963
Random Forest	**0.999**	0.876	**0.998**
logreg	**0.999**	**0.956**	0.996
MVaccuracy	**0.999**	0.948	0.996
MVexp	**0.999**	**0.954**	0.996
MVuniforme	**0.999**	0.945	0.996
CART	0.997	0.924	0.994

Besides few exceptions, most Super Learner strategies achieve better TPRs at the same FPR than both most first level learners and the Random Forest algorithm. Indeed, except for the CART-based Super Learner, all Super Learners generally outperform base predictors. Majority Voting with both exponential or direct accuracy based weights and the Logistic regression Super Learner strategies actually achieve the best results in all attack types; in particular, the MVexp Super Learner performs the best for all attack types, calling for a plausible generalizable and particularly fit model for this testing scenario. Still, it is worth mentioning that, similar to [3], performance is rather poor on the detection of DDoS attacks; the best algorithms detect about 80% of the attacks with a FPR of 10%. Exactly the opposite happens with the HTTP flashcrowds, where all Super Learners and some of the base predictors are above 95% detection rate with a FPR below 5%. The benefits of the Super Learner approach are less evident in this case, as there is not much room left for improvement, It is worth mentioning that the Random Forest detector never achieves an AUC score higher than any of the Super Learner strategies.

4.3 Detection of Network Anomalies

Fig. 2 depicts the ROC curves obtained by each detector for each anomaly type, and Tab. 2 reports the corresponding AUC values. Results are split by First Level Learners and Super Learners in different figures. Similar to [4], achieved results for anomalies of type E1 and E3 differs completely from E2. Every predictor achieves an AUC over 99% and detection rates above 97% at a FPR of 5% for E1 anomalies. Thus, there is little room for improvement, which leads to only very subtle differences between the performances of Super Learners and base learners. Similar observations can be drawn from the detection of E3 anomalies.

The attempt to detect E2 anomalies shows quite different results. Not only all predictors performed relatively poor - the best ones achieve almost 80% TPR at a FPR of 5%, but also many of them achieve very low performance; e.g. the ensemble algorithm Random Forest gets a TPR just above 60% for a 5% FPR and only 80% TPR for a high FPR of 20%, clearly worse than any other ensemble technique. Still, this scenario is the one that better highlights the advantages of the Super Learner approach for anomaly detection, as all the Super Learners - expect from CART, outperform the first level learners.



(a) First Level Learners on E1.

(b) First Level Learners on E2.

(c) First Level Learners on E3.

(d) Super Learners on E1.

(e) Super Learners on E2.

(f) Super Learners on E3.

Figure 2: Detection Performance per type of anomaly on the semi-synthetic dataset

5 CONCLUDING REMARKS

The advantages of ensemble learning techniques for detection of attacks and anomalies, and particularly of the Super Learner approach, could be confirmed through evaluation. Not only we found that Super Learner based predictors have the ability to perform as well as the best available first level learner, but often achieve better results. The performance improvements are higher in scenarios where the performance of the first level predictors were relatively low; when first learners performance is already high, there is not enough room for improvement and the additional computational cost of the Super Learner may not be justified. The different evaluated Super Learner schemes achieved very similar performances. However, the MVexp Super Learner performs the best for all attack and anomaly types in both datasets, suggesting a potentially good approach to go for by default in similar binary classification problems. The simplicity and very low computational costs of MVExp majority voting makes also a very nice case for such type of models. We believe that this study would enable a broader application of ensemble learning approaches to network security and anomaly detection, with very promising results.

REFERENCES

[1] M. Van der Laan, E. C. Polley and A. E. Hubbard, "Super learner", in *Statistical applications in genetics and molecular biology*, vol. 6 (1), pp. 1-21, 2007.

[2] P. Casas, A. D'Alconzo, T. Zseby and M. Mellia, "Big-DAMA: Big Data Analytics for Network Traffic Monitoring and Analysis", in *ACM SIGCOMM LANCOMM Workshop*, 2016.

[3] P. Casas, A. D'Alconzo, G. Settanni, P. Fiadino and F. Skopik, "POSTER:(Semi)-Supervised Machine Learning Approaches for Network Security in High-Dimensional Network Data", in *ACM CCS*, 2016.

[4] P. Casas, P. Fiadino and A. D'Alconzo, "Machine-learning based approaches for anomaly detection and classification in cellular networks", in *TMA*, 2016.

[5] Y. Freund, R. E. Schapire, Y. Singer and M. K. Warmuth, "Using and combining predictors that specialize", in *ACM STOC*, 1997.

[6] J. Hansen, "Combining predictors: Some old methods and a new method", available online at Citeseer, 1998.

[7] T. Dietterich, "Ensemble learning", in *The handbook of brain theory and neural networks*, vol. 2, pp. 110-125, MIT Press, 2002.

[8] P. Sollich and A. Krogh, "Learning with ensembles: How overfitting can be useful", in *Advances in neural information processing systems*, pp. 190-196, 1996.

[9] R. Fontugne, P. Borgnat, P. Abry and K. Fukuda, "MAWILab: Combining Diverse Anomaly Detectors for Automated Anomaly Labeling and Performance Benchmarking", in *ACM CoNEXT*, 2010

[10] T. T. T. Nguyen and G. Armitage, "A Survey of Techniques for Internet Traffic Classification using Machine Learning", in *IEEE Comm. Surv. & Tut.*, vol. 10 (4), pp. 56-76, 2008.

[11] V. Chandola, A. Banerjee and V. Kumar, "Anomaly detection: A survey", in *ACM Comput. Surv.*, vol. 41 (3), pp. 1-58, 2009.

[12] M. Ahmed, A. Naser Mahmood and J. Hu, "A Survey of Network Anomaly Detection Techniques", in *J. Netw. Comput. Appl.*, vol. 60, pp. 19-31, 2016.

[13] W. Zhang, Q. Yang and Y. Geng, "A Survey of Anomaly Detection Methods in Networks", in *CNMT Symposium*, 2009.

[14] R. Ravinder Reddy, Y. Ramadevi and K. V. N. Sunitha, "Real Time Anomaly Detection Using Ensembles", in *ICISA International Conference*, 2014.

[15] M. Ozdemir and I. Sogukpinar, "An Android Malware Detection Architecture based on Ensemble Learning", in *Trans. on Machine Learning and Artificial Intelligence*, vol. 2 (3), pp. 90-106, 2014.