

Beating Gradient Boosting: Target-Guided Binning for Massively Scalable Classification in Real-Time

Dymitr Ruta, Ming Liu, Ling Cen
 EBTIC, Khalifa University, UAE
 {dymitr.ruta,liu.ming,cen.ling}@ku.ac.ae

Abstract—Gradient Boosting (GB) consistently outperforms other ML predictors especially in the context of binary classification based on multi-modal data of different forms and types. Its newest efficient implementations, including XGBoost, LGBM and CATBoost, push GB even further ahead with fast GPU-accelerated compute engine and optimized handling of categorical features. In an attempt to beat GB in both the performance and processing speed we propose a new simple yet fast and robust classification model based on predictive binning. At first all features undergo massively parallelized binning into a unified originally compressed risk representation, independently optimized to maximize the AUC score against the target. The resultant array of summarized micro-predictors, resembling 0-depth decision trees, directly expressing originally represented target risk, are then passed through the greedy feature selection to compose a robust wide-margin voting classifier, whose performance can beat GB while the extreme build and execution speed along with highly compressed representation welcomes extreme data sizes and real-time applicability. The model has been applied to detect cyber-security attacks on IoT devices within FedCSIS'2023 Challenge and scored 2nd place with the $AUC \approx 1$, leaving behind all the latest GB variants in performance and speed.

I. INTRODUCTION

THE RISE of AI to prominence in access and control of just about everything is upon us and so is the expanding infrastructure network of fixed and mobile sensing and computing devices, capable of sending and receiving data, known as the Internet of Things (IoT). In such digitized environment security and integrity of every individual node as well as the whole network is critical and hence the cyber-security against the internal and external threats of different nature and scale is of crucial importance. A survey presented in [1] offers a thorough overview of different security threats IoT devices are exposed to, reviews the current security mechanisms, trying to address them, and identifies a continuous challenge in this very fast evolving ecosystem, in which the evidence based designed security solutions are always playing a catch-up game and leave a lagged gap, in which new threats or attacks may inflict a lot of damage before they are detected, analysed and neutralized. Machine Learning (ML) has been growing in parallel to these revolutionary changes and since the outset offered methods for automated detection of security threats based on data, both by learning from historically labelled attack examples and by discovering and flagging anomalies from normal operation of the IoT devices. Several reviews of ML deployment in the IoT cyber-security environment have been presented recently like in [2] assessing various ML

models while focusing further on the SVM application to smart city traffic flows prediction, or in [3] where similar classification, utility and suitability analysis of the most common ML methods, applied in various aspects of IoT cyber-security, is carried out with perhaps a deeper focus on deep learning.

To our surprise, however, gradient boosting methods developed around the start of our century by the pioneering work in [4], [5], that have ever since consistently been winning big data and ML competitions ([6]-[11]), have been rather scarcely covered in the literature dedicated to cyber-security. We can argue that in more practical realistic ML applications to cyber-security like detecting threads based on complex log-extracted data, the old favourite models like SVM are simply not scalable enough [2], [11], while high-performing deep learning networks cannot easily encode the multi-modal unstructured data coming in a variety of forms and types, i.e. in quite different form than regular images or time-series, deep learning are performing well for. For such data GB models appear much more suitable and easy to be applied.

In this paper, however, we attempt to go a step further than a standard application of the optimized gradient boosting models. Inspired by the essence of what makes GB work well we propose the target guided binning (TGB) process that transforms all input features into an array of independent AUC-optimized and robust micro-predictors of the binary target with which a simple voting can outperform the latest GB variants both in terms of performance, transparency, data handling overhead and the processing speed. While binned features resemble somewhat 0-depth decision trees, they leave TGB process already AUC-pre-optimized to maximally suppress unnecessary complexities within the original feature domains while maximally exposing and summarizing its predictive power against the known binary target.

Neither optimising with respect to AUC [16], [17] nor binning [17], [18] is new in the context of classification. Since the advantages of AUC-optimized classifier design have been widely exposed [16], there was a fast-paced evolution towards AUC-optimized classification that eventually culminates with the advent of CATBoost [17], and the ingenious way it handles high cardinality categorical data utilising target statistic as a robust form of feature re-engineering. We build up on this direction by more explicit target guidance in a form of predictive binning uniformly applied to both categorical and numerical features, yet to guard against performance damaging *target-leakage*, extensively investigated in [18], we simply

build the cross-validation process into the binning process.

We intend to comparatively demonstrate the advantages of the proposed TGB over GB in an objectively evaluated FedCSIS'2023 challenge dedicated to predicting cyber attacks on IoT devices based on logs data files that the latest GB variants should perform very well for. We specifically focus on and optimise TGB for scalability utilizing massive parallelization of the processing pipeline that can be deployed along multiple dimensions to approach real-time readiness even for such large scale problems as the cyber-security attacks prediction. For that reason we deliberately leave, typically critical, feature engineering aspects unexplored to great depths, instead focusing on TGB algorithmics and comparative experimentation applied to only one family of features, that nevertheless produced excellent results that scored the 2nd place in the competition after leading throughout the whole preliminary phase.

The remainder of the paper is organized as follows. The FedCSIS'2023 Challenge is briefly described in Section II. The critical element of the proposed target guided binning process along with its fast, massively parallelized implementation are discussed in Section III. The composition of wide-margin voting classifier from TGB-binned features that includes important aspects of feature selection and the evaluation criteria they use, are covered in Section IV, followed with the description, presentation and discussion of the experimental results in Section V and the conclusions drawn in Section VI.

II. FEDCSIS'2023 CHALLENGE

The FedCSIS'2023¹ competition focused on detecting cyber-security threats based on the behavior of IoT devices captured in their detailed logs data. Over 20k of log files have been provided to the competitors, each of which representing a single data sample of timestamped variable-length sequence of events capturing specific IoT device interaction/operation over a fixed period of time. System calls, system and user processes' details, lists of open files and libraries, counts of various events, errors, integrity checks are just some of 24 raw features included in the log files both in numerical and categorical (text) format. Out of the total 20044 samples the correct binary target label (*ATTACK*) was provided for 15027 training samples, leaving the remaining unlabeled 5017 examples for AUC testing on the KnowledgePit.ml platform, hosting the competition. Before the final evaluation of the submitted full solutions, i.e. throughout the competition, KnowledgePit.ml operated a leader-board of competitors' solutions evaluated based on the preliminary set of unknown 10% of the full testing set. FedCSIS'2023 Challenge is sponsored by the Łukasiewicz Research Network - Institute of Innovative Technologies, EMAG and EFIGO sp. z o.o. companies.

III. TARGET-GUIDED BINNING (TGB)

Given the data examples are provided in a composite form a variable-length table of time-ordered event features it was imminent that any kind of feature engineering strategy would

involve some form of aggregation over the whole log table of typically thousands of records. Moreover given relatively large number of unique values observed for several categorical features it is expected that potential number of possible derived features could be large. In an attempt to extract possibly fullest predictive value from such evidence we decided to reduce feature engineering to measuring the per-log-frequency of all observed unique feature values and simultaneously transform these frequencies into summarized ordinal target-risk levels monotonically increasing with the target likelihood conditioned on the intervals or subsets contained within each risk level. Such predictive TGB transforms all feature space irrespective of their form or data type into unified, numerically stable and additive micro-predictors of the target.

Target-guided binning focuses on a single, very simple goal: how to exploit the guidance of the binary target to bin the input feature in a way that maximally improves its generalized predictive power over the target. An objective, scale- and threshold- invariant measurement of feature predictive power in binary classification is the area under the receiver-operator curve (AUC). Denoting by x and $y(x) = y_x$ our input variable and the binary target, respectively, and by $AUC(y_x, x)$ the empirical AUC between y_x and x , the target-guided binning process can be formally defined by the transformation function T that maps all values of x into $x_T \in \{1, 2, \dots, k\}$ such that the $AUC(y_x, x_T)$ is maximized:

$$T : x_T = T(x, y_x), \quad x_T = \arg \max_{x_T} AUC(y_x, x_T) \quad (1)$$

At the first glance, this task seems trivial given the relation of AUC and the Wilcoxon-Mann-Whitney statistic [14], [15], that gives an AUC a simple interpretation of the probability of a random *positive* $x_+ = \{x_i : y(x_i) = 1\}$ being larger than the random *negative* $x_- = \{x_j : y(x_j) = 0\}$:

$$AUC(y, x) = P(x_+ > x_-) \quad (2)$$

It is trivial to show that to maximize such probability it is sufficient to simply transform x to the ranking of target rates along all unique values of x . The recipe for (y) target-guided binning of x that maximizes $AUC(y, x)$ seems, therefore, to be just finding unique values of x : x_u , computing target rates y_{xu} for all x_u , and replacing x with the positions they appear in x_u sorted by y_{xu} . Assuming Matlab coding syntax, finding x_T becomes straightforward:

$$[\sim, j] = \text{sort}(y_{xu}); \quad [\sim, x_T] = \text{ismember}(x, x_u(j));$$

Although such logic is in principle correct it ignores a fundamental property of a good predictor: the generalization ability and would likely fail on two accounts. First, the binning for numerical variable has to provide the mapping for the entire domain, not just unique values observed in the training set, otherwise the binning is unable to allocate previously unseen values of x into any bin. Second, a feature binned as described above essentially over-fits the observed data with the degree dependent on the the number of unique inputs x_u . In the extreme case of (almost) all unique values, that is typical for continuous floating point features, the target rates will be

¹<https://knowledgepit.ml/fedcsis-2023-challenge>

extremely unreliable as computed on the basis of just a single or a few examples and would wildly differ in the unseen testing set, on which the binning should be designed to perform well.

To address these two cases our target guided binning would operate on the intervals (subsets) that at all times span the entire domain (universal set) of numerical (categorical) feature and attempt to efficiently and optimally define their numbers, edges (set members) and label-permutation that maximises AUC, but not over the training data (x_u, y_{xu}) , on which the bins are built and merged, but on the previously unseen validation set (v_u, y_{vu}) . The bottom-up approach is proposed for our TGB algorithm, which starts from singleton intervals (subsets) exclusively covering all unique values x_u and then greedily merge to maximise $AUC(y_x, x_T)$ but only until monitored validation set performance $AUC(y_v, v_T)$ starts falling.

A. Massively parallelized TGB implementation

Target guided binning has been developed with extreme efficiency and scalability in mind. Each feature is binned independently and in parallel using the same binary target as a guidance in such a way that the AUC score of its binned representation against the target is maximized in a generalization sense, i.e. AUC is measured via cross-validation on different data partitions than those on which the bin definitions were built.

TGB starts from building singleton intervals (or subsets for categorical features) containing all unique values and reordering them along the rising likelihood of positive target. Then the intervals/subsets proceed to greedy merger process which continues until no further gain in validation AUC can be achieved through further mergers. Figure 1 illustrates a sample TGB process for numerical feature which starts from individual singleton intervals and then proceeds through the greedy neighboring interval merger until no more validation AUC improvement is possible, which in the depicted sample scenario converges to 6 intervals. The AUC-optimized intervals are then mapped to ordinal bin labels calibrated or scaled according to user preferences yet always monotonic with the conditional positive target rate. Feature 1 also illustrates readiness for massively parallelized implementation of the TGB process which could be effortlessly applied along the feature level, cross-validation partitions or testing for optimal merger along running pairs of neighboring intervals.

The missing data (NaNs) are mapped to a bin that has the closest posterior target probability as the one observed for missing data. Similarly previously unseen data are provisioned to receive bin that has the target posterior probability the closest to the target prior. Greedy interval merger follows very fast vectorized test of the impact of the decomposed AUC score implemented on matrix formulation that allows to compute all simulated pairs merger in a single step per round resulting in tabular formatted bin definitions mapping all original feature domain into optimised incrementally summarized intervals/subsets labelled with target-monotonic ordinal risk levels. Given the final bin definitions, transforming any new data into bins is equally lightning fast and, importantly,

represented by *uint8* data type reducing the binned data complexity to just 1 byte per value. Compared to the original data typically coming in double precision or text format, TGB typically reduces the size of the memory required to hold the data around 100-fold, while obfuscating the original values behind ordinal risk mask.

IV. WIDE-MARGIN VOTING CLASSIFIER COMPOSITION

Constructing a robust classifier based on TGB-transformed (binned) data is straightforward and in its simplest form can be executed by a simple voting i.e. by adding up all feature bin values (risk votes). Further AUC-measured performance gains can be achieved by more or less sophisticated feature selection strategies, which for the voting classifier simply translates into finding a sum of binned features that maximises AUC against the binary target. Two highly scalable heuristic optimisation methods have been developed to execute such robust additive selection of binned features and both can deal with hundreds of thousands of features in seconds if supported by multi-core parallel processing and/or dedicated capable GPU.

A. Greedy forward selection (GFS)

Greedy forward selection of binned features starts from the strongest feature and keeps adding features that maximally improve the appended sum's AUC against the target in each round until this is no longer possible. The process of testing for optimal addition is fast since the current best subset is constantly retained in the form of collapsed running sum and stored indices of selected members, while testing the AUC improvement when adding another binned feature is vectorized and massively parallelized with additional speedups possible when executed on the GPU. In practical applications, when faced with tens to hundreds of thousands of features, each round of finding a binned feature that maximally improves the pool's AUC usually takes around 1s. In the latest implementation this greedy search was additionally improved by reducing the data type of vectors holding the sums to *uint16* and allowing each feature to be added multiple times - thereby equipping the method with a fast feature weighting capability.

B. Fast probability based incremental learning (FPBIL)

Probability based incremental learning (PBIL) is a simple population based heuristic optimization that is perfectly suited for simple evaluation functions based on binary encoded feature selection. Beyond that fit, PBIL has been chosen to help with feature selection also for two other reasons. Its critical operation is constant sampling from the probability vector that involves generation of random number matrices of enormous sizes that can be massively accelerated on the GPU. Moreover, evaluation of the population of solutions at each generation involves preparation of the intermediate voting sums corresponding to binarized selection vectors sampled from the evolving probability vector, all of which has been very efficiently vectorized and passed on to equally optimized and parallelized evaluation of the AUC. Operating such PBIL on the GPU with the Philox based random number generator

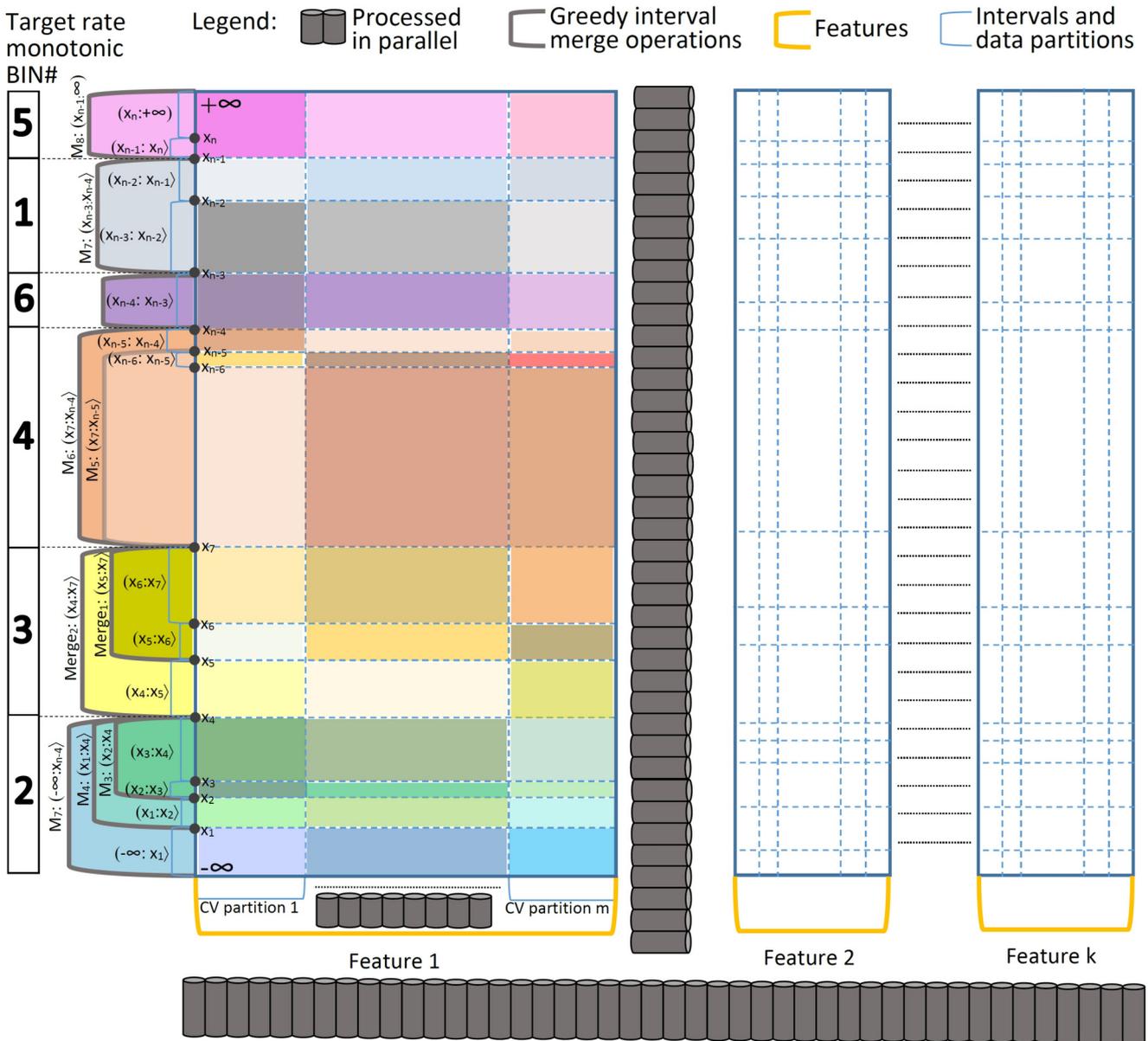


Fig. 1. An annotated example of the target guided binning (TGB) for numerical feature x . In the original domain $(-\infty, +\infty)$ first all unique values: $\{x_1 : x_n\}$ are found and wrapped within singleton intervals: $(-\infty : x_1], (x_1, x_2], \dots, (x_n : +\infty)$. Then neighboring intervals are greedily merged along multiple cross-validation partitions until the validation sets AUC against the target no longer improves leaving the final optimised intervals ordinarily labelled to represent target-monotonic risk levels. The process is ready for massive parallelization along multiple dimensions: independent features, cross-validation partitions and the neighboring pairs of intervals examination for optimal merger. Distinct colors are representing the conditional target rate heatmap and the effect of its aggregation after mergers.

on a population of 1000 100k-elements solutions with a learning rate of 0.5-1 typically converges after a couple of hundreds of generations at the pace of multiple generations per second. Compared to the greedy forward selection which normally converges with up to 200 out of 100k features, accelerated PBIL-based selection converges with thousands of features and typically better AUC-score

C. Criteria for feature selection evaluation

Ideal evaluation criterion for feature selection is the actual classifier performance for the selected features. The only reason why much simpler proxy measures are normally used is that evaluating the classifier with different set of features is expensive and normally requires a rebuilt of the whole model from scratch to extract new classifier output. For our case, however, the voting classifier only needs to add the newly selected feature values to the cumulative sum from features

already in the pool to update the classifier output, hence it follows an online update process and is therefore very fast. For this reason TGB with voting enables us to use directly the powerful threshold independent classifier performance measures like AUC as a feature selection criterion which supports classifier robustness while keeping it simple and fast.

1) *Area under the ROC curve (AUC)*: AUC measure has already been discussed above as the prime threshold-less indicator of the overall predictive power of a feature x against the binary target y . In the context of target guided binning for the reason of being AUC-optimized and also due to the fact that computing AUC for binned features likely involves fewer unique bin-label ordinal values, we have introduced a dedicated summarized representation of the relationship between x and y for the purpose of AUC computation called feature predictive structure $P = [u, c, v]$ that contains a sorted list u of unique values of x , and the corresponding lists of their counts c as well as the counts of the positive targets $v = \sum(y|x)$. For simplicity we will replace the original x and y with their summarized representation in P such that $P = [x, c, y]$. Note that such summarized representation significantly reduces the sizes of both the feature x with respect to the target y down to the essential statistics sufficient to evaluate its full predictive power. Given P , the cumulative true and false positive vectors can be readily computed for multiple features or targets by this vectorized Matlab code:

```
tp=[0;cumsum(flipud(y))];
fp=bsxfun(@minus,[0;cumsum(flipud(c))],tp);
tp=bsxfun(@rdivide,tp,sum(y));
fp=bsxfun(@rdivide,fp,sum(y));
```

such that the AUC can be accurately and rapidly computed for multiple features or targets using a 1-liner:

```
auc=sum(diff(fp).* (tp(1:end-1,:)+tp(2:end,:))/2);
```

2) *Kolmogorov-Smirnov Distance (KSD)*: Kolmogorov-Smirnov distance, test or statistic in our context expresses simply the maximum absolute difference between the cumulative rate of positive targets and the cumulative rate of negative examples along the sorted unique inputs x . Given our compact predictive structure $P = [x, c, y]$ KSD can be rapidly computed for multiple inputs/targets using:

```
y=cumsum(y); s=y(end); c=cumsum(c)-y; n=c(end)-s;
ksd=bsxfun(@rdivide,y,s)-bsxfun(@rdivide,c,n);
ksd=max(abs(ksd));
```

3) *Classification Impurity Score (CIS)*: This new measure utilizes the specificity of working with binned feature votes and is designed to stimulate stable wide margin classifier especially for very high performance close to AUC=1. The measure works on the sorted predictor outputs (sums of bin votes) and focuses on the interval, within which samples are not classified 100% correctly. For every sample falling within this interval it then simply adds up distances between the prediction (sum) for these samples and the interval boundary that if reached would eliminate the misclassification for any threshold. Since our voting classifier simply holds the sum of selected binned features, the logic of this measure is to evaluate how many

votes need to be added (for false negative) to or subtracted (for false positive) from the current sum of votes such that the sample would be correctly classified irrespective of the applied threshold. Formally, assuming sorted classifier outputs x_i , $i = 1, \dots, n$, the corresponding binary targets y_i and the interval of indices $j = k, \dots, l$ such that $1 \leq k < l \leq n$ and

$$\begin{aligned} \forall_{s:x_s \leq x_k} y_s &= 0 \\ \forall_{s:x_s \geq x_l} y_s &= 1 \end{aligned} \quad (3)$$

then the CIS can be defined by:

$$CSI = \sum_j (x_l - x_j) y_j + \sum_j (x_j - x_k) \bar{y}_j \quad (4)$$

Using Matlab code the above definition can be readily captured as follows:

```
i=find(x<x(find(y,1,'first')),1,'last');
if isempty(i) i=1; end
j=find(x>x(find(~y,1,'last')),1,'first');
if isempty(j) j=numel(x); end
l=i:j;
cis=sum(x(j)-x(l(y(l)))) + sum(x(l(~y(l)))-x(i));
```

Note that in case of 100% accurate classification the impurity measure could be adjusted to receive negative values proportional to the gaps or margins in votes that needed to be bridged to observe classification impurity, hence such measure can be very effective for very high performance wide-margin classification with AUC scores very close to 1, which happens to be the case of the FedCSIS' 2023 Challenge.

V. EXPERIMENTAL RESULTS

All features have been generated in the exactly same form capturing the frequency (counts) of unique values observed within the log files. For the datetime and other continuous numerical variables the domain has been split into 100 equi-percentile intervals and the derived features measured defacto frequencies of observed percentile values. For features listing all open filenames and libraries with paths the two variants of unique elements were applied: the whole unique paths separated by commas and, in the second variant, all the unique path sub-strings separated by \ character. Such feature engineering process resulted in over 300k 1-hot-encoded style features that after reduction by eliminating duplicates and constant features shrunk to a set of about 40k of unique raw features. These features have then been passed on to the TGB process allowing up to 20 (and later 100) unique bins applied only on the training set of 15027 labelled samples and resulted with bin definitions re-applied on both the training and testing sets to achieve the final transformed training and testing sets taking ordinal values from 1 to 20 (100).

Feature selection process followed on the binned training set in all combinations of the presented feature selection methods and evaluation metrics, however, we only show the results for AUC and CIS since KSD produced the results similar to AUC.

The feature subset sums obtained as a result of all the selection-evaluation combinations along with outputs from many other variants of restricted feature subsets and gradient

boosting models applied for comparison have been normalized within (0,1) interval and submitted for evaluation on the preliminary testing set containing only 10% of all testing samples. The feedback received was in line with the results received from the validation sets with the top scored model variants, notably achieving leaderboard's top AUC=1.0 submitted as final solutions for the evaluation on the full testing set.

Comparative AUC performance results of our target guided binning (TGB) with gradient boosting (GB) variants classifiers and various combinations of feature-selection and evaluation criteria are presented in Table I. For both the training and validation sets we have observed GFS performing slightly better with AUC than CIS criterion however the opposite was observed for FPBIL selection method. The FPBIL applied with CIS metric typically returned solutions of a couple of thousands of features with the final converged impurities of just about 50-500, while starting from impurities in the order of millions. On the other hand the GFS typically converged with about only 100-200 features, for which the added score produced the AUC reaching extremely close to 1. What produced the best results, however, was sequentially applied GFS interchangeably with AUC and CIS criteria, until no further improvement in the validation AUC could be achieved. Although results for 100-bin TGB appear to show slightly better validation results than for TGB with up to 20 bins, final testing revealed later that 20-bin TGB could have performed better, i.e. 100-bin TGB appeared to be slightly over-fitted with too fine granularity and the best results could be expected somewhere in between for example 50-bin TGB. Although in our validation TGB on its own i.e. with simple voting outperforms all GB model variants, final testing revealed that CATBoost could climb to similar performance levels if applied on top of the 20-binned rather than raw features and could most likely have improved our final combined testing score thanks to a significant diversity with TGB-generated results.

TABLE I
COMPARATIVE PERFORMANCE OF GB/TGB VARIANTS COMBINED WITH DIFFERENT FEATURE SELECTION/EVALUATION CRITERIA.

Classifier	FSelection	Criterion	AUC Score	
			VAL	TST
XGBoost	ALL-LogLoss		0.9992	0.9980
LGBM	ALL-LogLoss		0.9967	0.9954
CATBoost	ALL-LogLoss		0.9994	0.9983
TGB20-SUM	GFS	AUC	0.9996	0.9985
	GFS	CIS	0.9994	0.9982
	FPBIL	AUC	0.9992	0.9981
	FPBIL	CIS	0.9994	0.9983
	GFS	AUC-CIS	0.9997	0.9991
TGB100-SUM	GFS	AUC-CIS	0.9997	0.9989
	FPBIL	CIS	0.9998	0.9979
XGB-BIN20	ALL-LogLoss		0.9992	0.9985
CATBoost-BIN20	ALL-LogLoss		0.9994	0.9993
MEAN(TOP5(TGB))	NA		1	0.9997

VI. CONCLUSION

Presented target guided binning rapidly transforms any input evidence into a robust array of 1-feature micro-predictors of

the binary target and offers readily available, high quality classification by voting with ordinal-risk represented binned feature outputs in near-real time. Further performance gains are available through fast parallelized greedy feature selection and gpu-optimized FPBIL features selection methods utilizing both AUC and newly introduced CIS as evaluation criterion to achieve stable high margin performance. In the competitive setup of detecting cyber-security attacks on the IoT devices based on log files data the presented methodology appears to consistently beat gradient boosting models in all aspects: the speed of building the model, the classification performance, simplicity, transparency and added security layer, topping the preliminary evaluation on the leader-board of the FedC-SIS'2023 Challenge with the score of AUC=1 and eventually scoring the 2nd place with AUC=0.9997 in the final testing.

REFERENCES

- [1] F. Alaba, M. Othman, I. Hashem, F. Alotaibi. Internet of Things security: A survey. *Journal of Network and Computer Applications* 88:10-28, 2017.
- [2] M. Mahdavinjad, M. Rezvan, M. Barekatin, P. Adibi, P. Barnaghi, A. Sheth, Machine learning for internet of things data analysis: a survey, *Digital Communications and Networks*, 2018.
- [3] M. Garadi, A. Mohamed, A. Ali, X. Du, I. Ali and M. Guizani. A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security, *IEEE Communications Surveys & Tutorials*, 2020.
- [4] L. Mason, J. Baxter, P.L. Bartlett, and M. Frean. Boosting Algorithms as Gradient Descent In S.A. Solla, T.K. Leen and K. Müller (eds): *Advances in Neural Information Processing Systems 12*: 512–518, MIT Press, 1999.
- [5] J.H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* 29(5): 1189-1232, 2001.
- [6] D. Ruta, M. Liu, L. Cen. Feature Engineering for Prediction of Frags in Tactical Games. *Proc. Int. Conf. 2023 IEEE International Conference on Multimedia and Expo*, 2023.
- [7] D. Ruta, M. Liu, L. Cen and Q. Hieu Vu. Diversified gradient boosting ensembles for prediction of the cost of forwarding contracts. *Proc. 17th Int. Conf. on Computer Science and Intelligence Sys.*, pp 431-436, 2022.
- [8] D. Ruta, L. Cen, M. Liu and Q. Hieu Vu. Automated feature engineering for prediction of victories in online computer games. *Proc. Int. Conf on Big Data*, pp 5672-5678, 2021.
- [9] Q. Hieu Vu, D. Ruta, L. Cen and M. Liu. A combination of general and specific models to predict victories in video games. *Proc. Int. Conf. on Big Data*, pp 5683-5690, 2021.
- [10] D. Ruta, L. Cen and Q. Hieu Vu. Deep Bi-Directional LSTM Networks for Device Workload Forecasting. *Proc. 15th Int. Conf. Comp. Science and Inf. Sys.*, pp 115-118, 2020.
- [11] D. Ruta, L. Cen, Q. Hieu Vu. Greedy Incremental Support Vector Regression. *Proc. Fed. Conf. on Comp. Sci. and Inf. Sys.*, pp 7-9, 2019.
- [12] Q. Hieu Vu, D. Ruta and L. Cen. Gradient boosting decision trees for cyber-security threats detection based on network events logs. *Proc. IEEE Int. Conf. Big Data*, pp 5921-5928, 2019.
- [13] Q. Hieu Vu, D. Ruta, A. Ruta and L. Cen. Predicting Win-rates of Hearthstone Decks: Models and Features that Won AIAA'2018 Data Mining Challenge. *Int. Symp. Advances in Artificial Intelligence and Apps (AIAA)*, pp 197-200, 2018.
- [14] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics* 1:80–83, 1945.
- [15] H.B. Mann, D.R. Whitney. On a test whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics* 18: 50–60, 1947.
- [16] C.X. Ling, J. Huang, and H. Zhang. AUC: a statistically consistent and more discriminating measure than accuracy. In *Int. Joint Conf. on Artificial Intelligence*, pp 519–526, 2003.
- [17] Z. Yang, Q. Xu, S. Bao, Y. He, X. Cao, Q. Huang. Optimizing Two-way Partial AUC with an End-to-end Framework. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 45:10228-10246, 2023
- [18] L. Prokhorenkova, G. Gusev, A. Vorobev, A.V. Dorogush and A. Gulin. Catboost: unbiased boosting with categorical features. In S. Bengio et al (eds.): *Advances in Neural Information Processing Systems* 31:6638–6648, Curran Associates, Inc, 2018.