applied sciences

MDPI

Article

Time Aware F-Score for Cybersecurity Early Detection Evaluation

Manuel López-Vizcaíno *,†, Francisco J. Nóvoa †, Diego Fernández † and Fidel Cacheda †

Center for Information and Communications Technologies Research (CITIC), Department of Computer Science and Information Technologies, University of A Coruña, 15071 A Coruña, Spain; fidel.cacheda@udc.es (F.C.)

* Correspondence: manuel.fernandezl@udc.es

† These authors contributed equally to this work.

Abstract: With the increase in the use of Internet interconnected systems, security has become of utmost importance. One key element to guarantee an adequate level of security is being able to detect the threat as soon as possible, decreasing the risk of consequences derived from those actions. In this paper, a new metric for early detection system evaluation that takes into account the delay in detection is defined. Time aware F-score (TaF) takes into account the number of items or individual elements processed to determine if an element is an anomaly or if it is not relevant to be detected. These results are validated by means of a dual approach to cybersecurity, Operative System (OS) scan attack as part of systems and network security and the detection of depression in social media networks as part of the protection of users. Also, different approaches, oriented towards studying the impact of single item selection, are applied to final decisions. This study allows to establish that nitems selection method is usually the best option for early detection systems. TaF metric provides, as well, an adequate alternative for time sensitive detection evaluation.

Keywords: early detection; machine learning; classification algorithms; network security; social networks; time-aware metrics

check for updates

Citation: López-Vizcaíno, M.; Novoa, F.J.; Fernández, D.; Cacheda, F. Time Aware F-Score for Cybersecurity Early Detection Evaluation. *Appl. Sci.* **2024**, *14*, 574. https://doi.org/10.3390/app14020574

Academic Editors: Peter R. J. Trim and Yang-Im Lee

Received: 19 November 2023
Revised: 4 January 2024
Accepted: 5 January 2024
Published: 9 January 2024

1. Introduction

The relevance of the early detection problem has been explored in many fields. Although a thorough study was proposed there was limited research on its evaluation metrics and methodologies. Mostly there is little work related to non-dataset-dependent metrics and lack of interpretability and better discrimination among results. In a real world environment, where a true streaming evaluation is applied, no data from the full dataset is available, so parameters for the metric must be extracted from the problem.

Many fields could benefit from an early detection approach as stopping any kind of anomaly or problem as soon as possible would minimise the risks of unwanted results.

Particularly, in the field of network and systems security, the longer the time passes and the more phases of the attack are achieved, the higher the probabilities of significant damage.

In the early detection in social media field, the detention of fake news or rumours as well as cyberbullying would decrease their consequences. Those consequences include depression, self-esteem decrease, suicide or suicide ideations, for example.

The lack of a proper dataset independent early detection metric with good interpretability was solved with the definition of TaF. This is a non-dataset-dependent metric for the evaluation of early detection systems, presented as an alternative to the issues found in the state-of-the-art metrics. Additionally, the evaluation of multiple alternatives for detection systems functions could improve the results, thanks to the variety of options available for the different problems.

To summarise, the fundamental issue found in this topic was related to the proper time aware evaluation and thus the main contribution of this paper is the definition of a non-dataset-dependent time-aware metric. Also, another contribution is the study of different approaches for taking final decisions based on the elements that are being processed.

The remaining sections of this paper is organised as follows: After the Introduction section, a Related Works containing relevant references to analyse the state-of-the-art. Then, Section 3 present the formal representation of the methods used in the experiments, a brief analysis of the datasets used, the models applied and the metrics used for the evaluation of performance. Next, on the Results section values from the experiments are presented in terms of figures, tables and a study of the results. Lastly, Discussion and Conclusions outline the outcome of the experiments, displaying also the implications and limitations of this research.

2. Related Works

As stated in the previous section, the relevance of the early detection problem has been explored in many fields. Although, a thorough study was proposed there was limited research on its evaluation metrics and methodologies. Best efforts as evaluation metric definition for the early detection problem was presented by *Early Risk Detection Error* (*ERDE*) [1], *F-latency* [2] an also by [3] where an alternative, *Time aware Precision* (*TaP*) non-dataset-dependent metric was proposed.

At the 2017 CLEF (Conference an Labs for the Evaluation Forum), *ERDE* was presented in the workshop for early prediction (*eRisk*) as the metric for the evaluation of the tasks. Then, by using the time-aware proposed metric, participants detection systems for early detection of situations such as depression or other behaviours and disorders using social media network data [1,4].

F-latency, a latency-weighted $F1$, was created to avoid heuristically defined parameters, as it was one of the problems detected by the authors in the definition of *ERDE*. Those parameters were replaced by dataset defined values for the evaluation of depression detection in social media [5].

Time aware Precision (*TaP*), was presented in [3] based on the study of *ERDE* and *F-latency* as well as other non time aware metrics to overcome problems with the definition of both. Such as, interpretability, better discrimination among results and to obtain a dataset agnostic approach. In a real world environment, where a true streaming evaluation is applied, no data from the full dataset is available, so parameters for the metric must be extracted from the problem.

In terms of metrics those three, *ERDE* [1] *F-latency* [5] and *TaP* [3], are the best effort presented for early detection evaluation with specific latency dependent metrics.

Also, it must be mentioned that disregarding specific latency aware metrics, some examples can be found by using traditional metrics like precision, recall or F-score, for the early detection problem. For example [6] or [7], where $F1$ is measured at specific points without time penalization.

Particularly, in the field of network and systems security, early detection of attacks could prevent an increase of the threat minimising the outcome, as presented in [8]. The longer the time passes and the more phases of the attack are achieved, the higher the probabilities of significant damage. As development in this area, several works had been published, such as [9,10] in order to detect attacks at their early stages, usually not taking into account the time in their evaluation. Even, if it is incorporated, usually is just measured by the amount of time until the detection, as shown in [11,12], where a distributed denial of service (DDOS) is targeted.

In the early detection in social media some works have been presented to target different aspects of those interactions and the problems that could derive from them. For example, in order to stop as soon as possible the diffusion of fake news and rumours, some solutions are presented as in [13,14] but relying only in the time required for the detection as latency evaluation. This, combined with the increase of cyberbullying [15] and the possible consequences of it and the diffusion and spread of fake news and rumours could lead to depression, self-esteem decrease, suicide ideations or even suicide [16].

Recently, a thorough comparison between different approaches for cyberbullying detection was presented in [17], where as part of machine learning evaluation, Logistic

Regression is shown as an alternative. As part of the cyberbullying detection problem, some session based studies had been presented, using specifically Large Language Models (LLM) for the early detection task [18] . Lately, also [19] applies machine learning models as Decision Tree (DT), Random Forest (RF) and AdaBoost (AB) for cybersecurity threats in form of Networks Attacks.

Summarising, in terms of metrics, several attempts have been made towards an early detection evaluation, such as *ERDE* [1], *F-latency* [5] and *TaP* [3]. Although, some present several problems for real world applications and their interpretation.

3. Materials and Methods

In this section we present a thorough description of the elements used in the tests for obtaining the results. First, an introduction to the early detection representation and its particularities is included in the Methods section. Then, a comprehensive explanation of the datasets is shown, followed by the models used for the decision making phase. Finally, the metric used for the evaluation is presented.

3.1. Methods

The early detection problem presents certain characteristics that must be taken into account, and it must be formally described in order to define proper methods. Although a thorough description of the formal definition of the early detection problem can be found in [3], a summary is presented next for ease of interpretation. To do so, we define $E = \{e_1, e_2, \ldots, e_{|E|}\}$ as the set of entities susceptible of being classified, in this case $|E|$ describes the amount of entities. Each one of them (e) is composed from a series of items (I_e), with a label (l_e) for the class of the entity. This could represent either that the entity is anomalous or not, $l_e = true$ and $l_e = false$ respectively. It must be added that although this represents a binary classification task, early detection systems could provide a third value showing that the decision has not been taken yet (i.e., a delay).

The set of items for a particular entity is expected to change over time and is represented by $I_e = (< I_1^e, t_1^e >, < I_2^e, t_2^e >, \ldots, < I_n^e, t_n^e >)$, being $< I_k^e, t_k^e >, k \in [1, n]$ the tuple that represents, for each entity e, its k-th item I_k^e, and the timestamp associated with the particular item I_k^e is denoted as t_k^e.

Also, it is important to notice that the following statement must be true, as items must be time ordered:

$$\forall < I_k^e, t_k^e > : \; t_k^e \; before \; t_{k+1}^e$$

An item, I_k^e, is characterized by a feature vector, and, particularly, it can be inferred that all items linked to an entity, $I_k^e, k \in [1, n]$, share the same feature vector. Those attribute values will be expected to change over time.

$$I_k^e = \left[f_{k_1}^e, f_{k_2}^e, \ldots, f_{k_m}^e \right], k \in [1, n]$$

Due to the independence of entities, the sequence of items I_e for each entity $e \in E$ may exhibit varying lengths, denoted as n. It is important to emphasize that the number of features, m, remains consistent across all items.

In this kind of problem, for a given entity e, the goal is to identify any anomalous behaviour while examining the fewest number of items from I_e possible.

The objective function will be defined as $f(l_e, I_e \times [1..n]) \rightarrow \{0, 1, 2\}$. If an entity e is deemed anomalous following the analysis of i_1 to I_k items, this function will output 1 (i.e., *positive*). If an entity e is determined to be normal (i.e., non-anomalous or *negative*) after processing an amount of k items and the preceding ones, then $f(l_e, I_e, k) = 0$. Lastly, $f(l_e, I_e, k) = 2$ represents that no definitive decision can be made regarding e entity after processing an amount k of items, indicating the need for further processing (i.e., *delay*). As a result, when $f(l_e, I_e, k)$ yields outputs 0 and 1, the processing of items $I_{k+1}, \ldots, I_n$ is unnecessary. Conversely, if the output is 2, it necessitates the processing of additional items $I_{k+1}, \ldots, I_n$ until a conclusive output is obtained or the sequence of items is exhausted.

Among the different existent evaluation methods (e.g., batch, streaming, time-based, etc.) we choose streaming evaluation as it provides the best representation for each individual entity.

In this case, I_e, which represents the sequence formed by individual items, is treated individually and following the original sequence. As a result, for each entity e, the function $f(l_e, I_e, k)$, $k \in [1, n]$, must be executed until a conclusive output is achieved or until the maximum number of *items* (n) is reached. Similar to previous scenarios, when item k, I_k, is being processed, the model can incorporate all preceding items, $I_1, \ldots, I_{k-1}$, as illustrated in Figure 1.

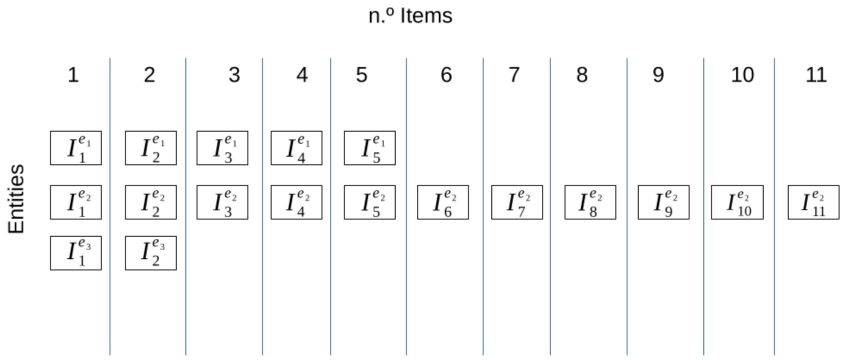

n.º Items
1
2
3
4
5
6
7
8
9
10
11
Entities
$I_1^{e_1}$ $I_2^{e_1}$ $I_3^{e_1}$ $I_4^{e_1}$ $I_5^{e_1}$
$I_1^{e_2}$ $I_2^{e_2}$ $I_3^{e_2}$ $I_4^{e_2}$ $I_5^{e_2}$ $I_6^{e_2}$ $I_7^{e_2}$ $I_8^{e_2}$ $I_9^{e_2}$ $I_{10}^{e_2}$ $I_{11}^{e_2}$
$I_1^{e_3}$ $I_2^{e_3}$

Figure 1. Items distribution for streaming evaluation.

3.2. Datasets

For the sake of variety in the experiments developed in this paper, we selected two different datasets. In order to study the outcome under particular circumstances, we choose two where both their nature and particular characteristics differ. The first one, obtained from Kitsune dataset [2], is the OS Scan Attack, which presents some particularities that will be shown later in this section. The second one was specially collected for the eRisk 2017 Workshop [1]. In this case, posts from the website Reddit were included after a selection process and being identified as written by users with depression or users without depression. As for Kitsune OS Scan Attack, an analysis is included next.

Kitsune dataset is composed by data traffic from a video monitoring network and includes different attacks performed over the network through several days. It was designed and utilized to evaluate the Intrusion Detection System—Network based (NIDS) described in [2]. In particular we use the OS Scan Attack, which examines the devices connected trying to detect hosts and which operating systems (OS) they are using, to find potential vulnerabilities, and it will be referred as “Network Attack” from now on. This dataset is composed by a set of network packets from which a group of engineered features are extracted and tagged as “Attack” or “Normal”. In order to further analyse the traffic, we divided it into bidirectional flows as described in [20,21], defined as the aggregation of packets with same pair source IP address—destination IP address, source port—destination port and protocol over a defined period of time. To account for the time division, we used timestamp of the packets as a split point using 0.1 s as threshold for the inter-packet time and 1 s as threshold for the flow span as described in [22]. For the experiments performed, we used the features described in [2]. As presented in the original paper only characteristics of the packet itself and its relation in time with the rest of the traffic are used in the creation of the dataset without including specific information of the flow they belong to.

Table 1 summarizes the main statistics for the datasets, among which OS Scan Attack from Kitsune dataset can be found. The amount of individual packets reaches nearly 1.7 million, distributed in 75,700 bidirectional flows. As the dataset represents an OS scan attack, most of anomalous flows are going to be around 2 packets in size, like it can be seen in the average of packets per flow for attack class, which accounts for the request and the reply from the device under attack. This, results on getting the majority of *Entities* of Anomalous type even if the greater part of *items* or packets belong to the Normal class.

The eRisk depression dataset, referred as "Depression Dataset" from now on was expressly collected for the 2017 edition of eRisk workshop on Early Detection [1]. It is composed by a set of publicly available Reddit posts published by users in about a year of use period. Those posts were later marked as "Depressed" or "Non-depressed" guided by self-reported diagnoses of depression. In this case, subjects correspond with the *Entities* (E) and each of the *items* (I_i^e) with the subject's posts. As for the features used in the experiments we will use the ones defined in [23,24]. Although, we will not include features created by the aggregation of a sequence of posts by only including individual post characteristics.

Also, we include in Table 1, along with the ones of the previous described dataset, the main statistics computed for the subjects and posts considered. This dataset is significantly smaller both in terms of number of *Entities* and *Items* but it has a higher average ratio of *items* per *entity*. With only 887 *Entities* and slightly over 500,000 items, it achieves nearly 600 items for each entity. Being the amount of "Anomalous" cases the 15.22% of the total subjects, it is relevant to display that this ratio reaches almost half the value for "Anomalous" than for "Normal" cases.

Table 1. Summary of the main characteristics of the datasets.

Dataset	Entities & Items	Normal	Anomalous	Total
Depression	Entities	752	135	887
	Items	481,837	49,557	531,394
	Items per entity	640.7	367.1	599.1
Network attacks	Entities	10,045	65,655	75,700
	Items	1,566,602	131,249	1,697,851
	Items per entity	155.96	1.99	22.43

3.3. Models

The set of models used in the experimental evaluation of the proposed selectors and metric is composed by some well known of-the-shelf state-of-the-art machine learning algorithms. The selection of this set was made based on the work presented in [25] for detection of cyberbullying in Vine social network. This methods were also tested on the same datasets for the early detection problem as shown in [3]. Particularly, the implementation by *scikit-learn* [26] was used, and the description of the models with the parameters used is listed below:

- *LinearSVC:* Support Vector Classification implementation with a 'linear' kernel that improves parameter selection and scalability.
  - Parameters: $C = 1$, $class\_weight = 'balanced'$, $dual = False$, $max\_iter = 1000$
- *ExtraTree:* Meta estimator that uses averaging of randomized decision trees for classification.
  - Parameters: $n\_estimators = 50$, $bootstrap = False$, $class\_weight = None$
- *AdaBoost:* Meta estimator that refits a classifier by updating weights of incorrectly classified instances.
  - Parameters: $n\_estimators = 1000$, $learning\_rate = 2.0$, $algorithm = 'SAMME.R'$
- *Random Forest:* Meta estimator which uses a determined number of decision trees and applies averaging to obtain the classifier.
  - Parameters: $n\_estimators = 500$, $class\_weight = None$, $max\_features = 'sqrt'$, $max\_depth = 7$, $bootstrap = False$

- *Logistic Regression:* Conformed by a *logit MaxEnt* classifier, where the maximum entropy classifier is combined with a logistic function.
  - Parameters: $C = 0.1$, $class\_weight = \text{'}balanced'$, $dual = False$, $penalty = \text{'}l2'$, $solver = \text{'}sag'$

After models for detection in individual items are trained, we apply a selector system to take the final decision based on individual items. Those decision functions are tested by using the results previously obtained with standard final deciders as baselines. In this case, four different functions are applied to this task, particularly:

- *mean*: Uses an aggregation of probabilities for each class, and decides one or the other based on those values.
- *bigger*: Uses the bigger class probability to provide the output of the classifier.
- *last*: Uses the last item processed to decide if the result is normal or anomalous
- *2last*: Uses the combination of the class probabilities of the last two values in order to provide the final output.

3.4. Metrics

For the early detection problem evaluation it is important to obtain models that not only make correct predictions but also that those are taken as soon in time as possible. In order to do so, a metric that is able to consider the time used for making the prediction is needed. Several metrics had been presented to fulfill that task, among which: Early Risk Detection Error *ERDE* [1], *F-latency* [5] and Time aware Precision *TaP* [3].

In this paper, we define the metric *Time aware F-score* or *TaF* for short, to overcome some limitations of the previous metrics and to ease the interpretation of the results. This metric presents a behaviour more similar to *F-latency*, than previous ones, but improving the configuration parameters. Reflecting the same idea as *TaP*, a penalization point and the degree of penalization is defined as problem-based instead of based on the particular values of the dataset. In this sense, it is more natural to do so that to set it based on the mean of those values. This last point is also crucial because in a real world streaming situation, it will not be possible to get the complete image or range of values before one specific item is processed.

The metric has been defined as follows:

$$TaF_{o,\lambda}(e_i, k) = \begin{cases} 1 & if\ TP \wedge k \leq o \\ 1 - pf_{o,\lambda}(k) & if\ TP \wedge k > o \\ 0 & if\ delay \end{cases}$$

$$TaF(E, k) = \frac{\sum_{e_i \in E} TaF(e_i, k)}{|E_{TP}| + \frac{1}{2}(|E_{FP}| + |E_{FN}|)}$$

The cases defined in $TaF_{o,\lambda}$ use the same principle as in the previous metrics (*ERDE* and *TaP*) but just for the correctly detected positive cases (true positives, *TP*). The maximum value is obtained if the item is identified before the point of measure o, otherwise a penalization function named $pf(k)_{o,\lambda}$ is applied.

For the final $TaF(E,k)$ value an aggregation of intermediate results is performed in order to use the final value as a penalized count of true positive cases. That is to use it instead of $|E_{TP}|$, applying, then, the F-score function. The number of real true positive cases (TP) is depicted as $|E_{TP}|$, whereas negative cases are shown as $|E_{FP}|$ and $|E_{FN}|$ for false positives and false negatives respectively.

For this metric, the penalty function is defined as follows, to give values in the range $[1, 0]$ which generates an output in the metric in the same range. This maintains the relation with the output values of F-score.

$$pf(k)_{o,\lambda} = -1 + \frac{2}{1 + e^{-\lambda(k-o)}}$$

The values of penalization shown in Figure 2 for different values of the parameter λ display how the results for individual entities e_i affects to the final value of TaF. In every instance a proper prediction is emitted although, the amount of *items* required to achieve that prediction varies as presented on X-axis. It must be observed that the output range of the function for TaF metric is $[0,1]$.

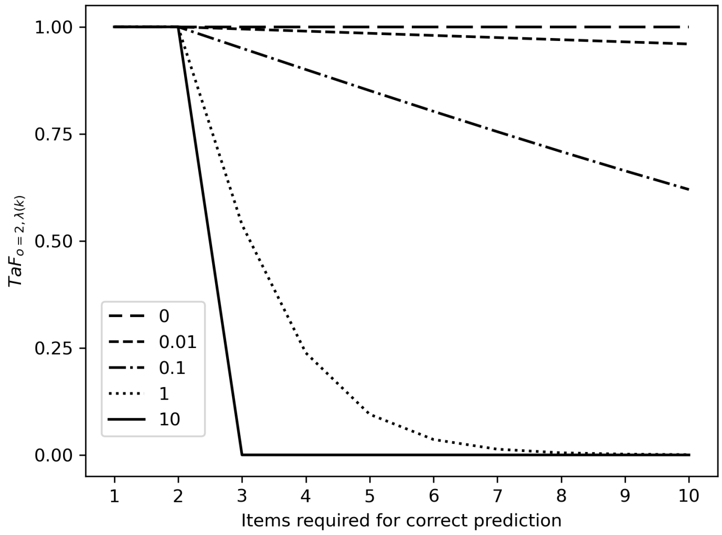


Figure 2. TaF configured with point of measure $o = 2$ and various values of λ. The number o items used to reach a correct prediction by the system are represented on the X-axis, supposing a delay was produced fro previous items. In this case, when $x = 5$ the system outputs the proper prediction on item $k = 5$ and, therefore, it generates a delay earlier on (i.e., $x < 5$).

4. Results

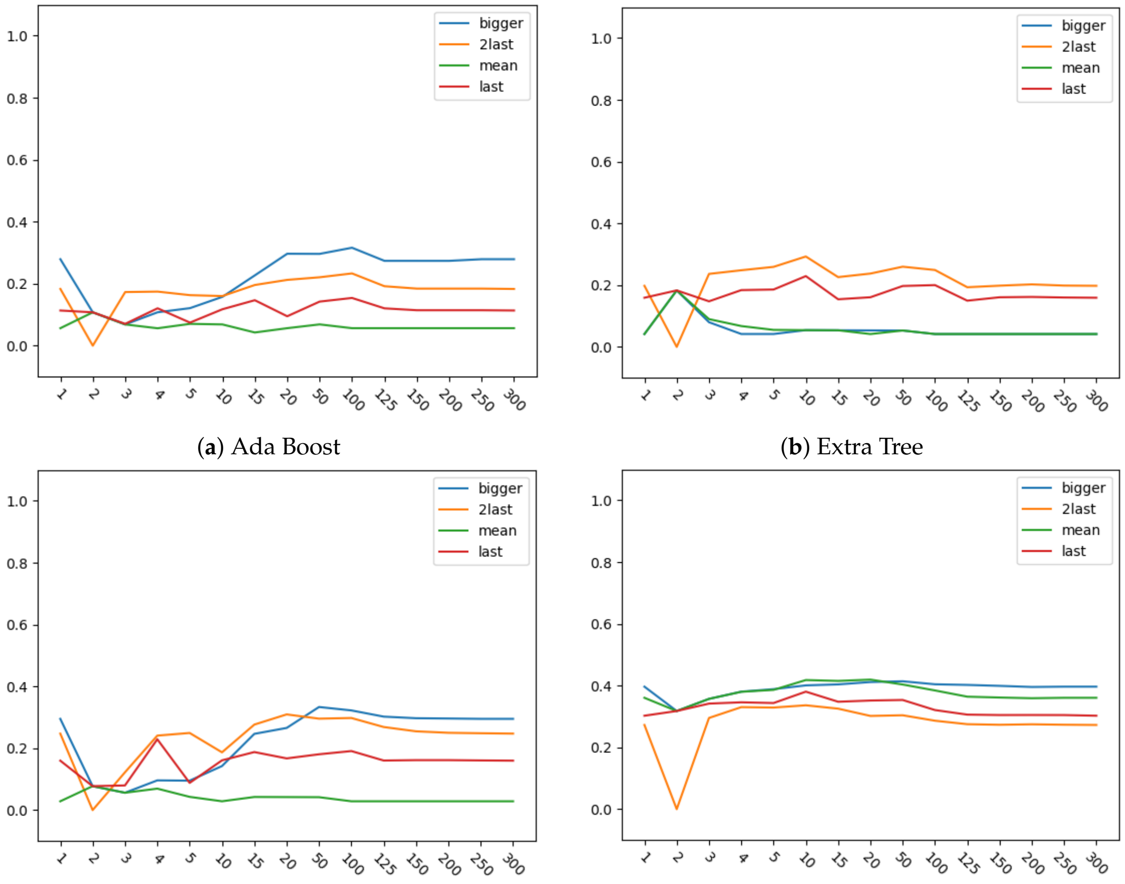
Results for eRisk dataset over Random Forest, Extra Tree, Ada Boost and Logistic Regression models are presented on Figures 3 and 4. Figure 3 shows the results obtained with Time aware F-score (TaF) metric, when compared with Figure 4, which shows the results when *F-latency* metric is used, some differences must be noticed.

First, TaF parameters are not dataset defined but problem defined, as presented in the previous section, in contrast to other metrics. Besides that, it can be seen that more differences between the four models, and mostly between different selectors for each model, can be spotted by using this metric. Particularly, it provides an improved representation of selectors' performances.

Also, when specific values are observed in Figures 3 and 4, some differences are shown between selectors. One particular observation can be made at 2 items for 2*last*, with a decrease in performance for both metrics. This can be explained mainly because of an absence of many predictions in the previous point combined with a poor performance when the first item is included for these decisions. The results could imply that the first element of the sequence is mostly incorrectly classified and it does not improve the general output of the algorithm at that point.

Finally regarding Figure 3, when the number of processed items increase, best behaviour for the majority of the models is obtained with bigger selection function, although for Extra Tree *2last* obtains the best results. This function is also the second better for Ada Boost and Random Forest, reaching even better values than bigger function for lower amount of items processed.

Furthermore, Table 2 presents results obtained with TaF with the same combination of models and selectors when applied on Kitsune dataset.



(**a**) Ada Boost

(**b**) Extra Tree

(**c**) Random Forest

(**d**) Logistic Regression

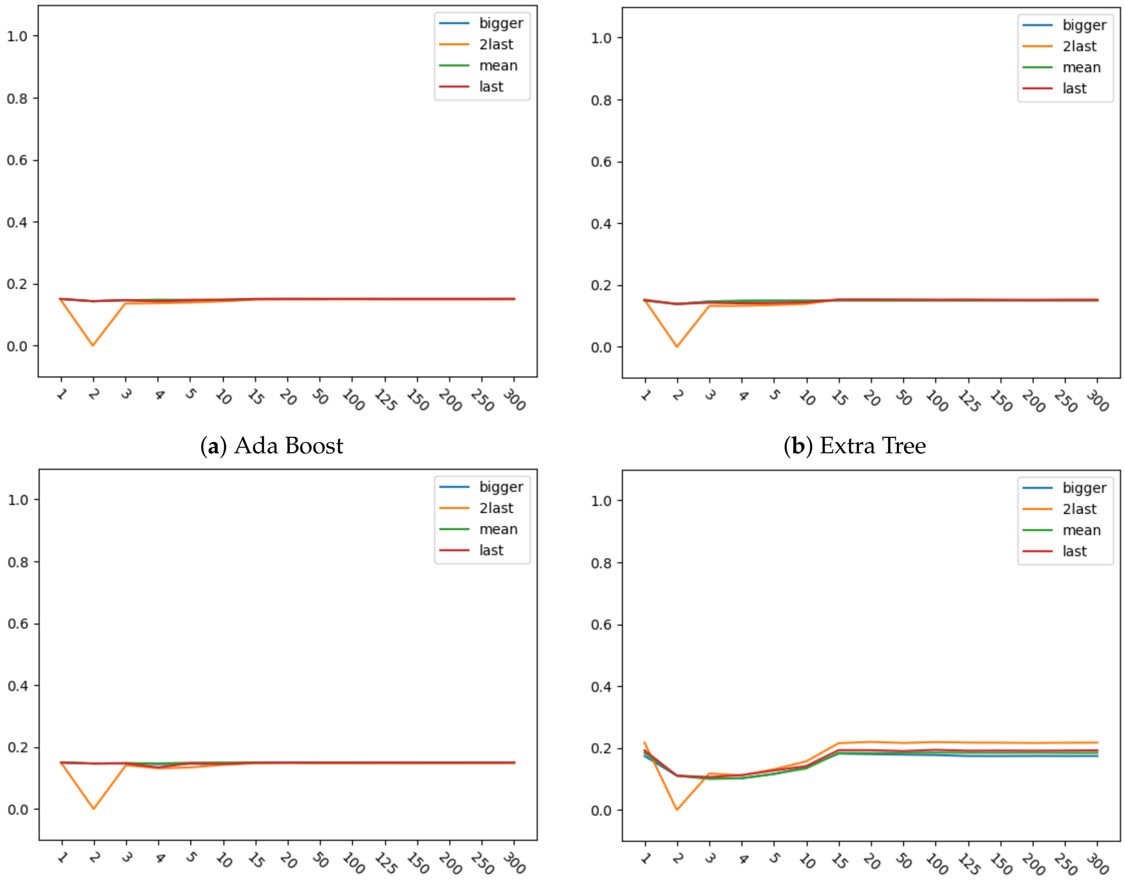
Figure 3. Results for combinations of selectors and models (Random Forest (RF), Extra Tree (ET), AdaBoost (AB) and Logistic Regression (LR)) for each number of items analyzed with eRisk dataset using Time aware F-score (TaF) as metric.

In this case, as the problem analysed is specific of an OS Scan Attack and defined flows present distinct characteristics, models trained display high values of TaF metric regardless the combination of models and selectors. Even though that is the case for this particular, it allows to observe how higher values of the metric seem less influenced by differences in selectors, and small differences can be observed in the model (Logistic Regression) where those values are lower.

Results on Kitsune dataset display also particularities when the amount of items processed is lower, and this situation is enhanced by the fact that this dataset presents a high number of entities with just two items. Also, it must be remembered that those small entities are in their vast majority attack entities.

Although no direct comparison with previous results referenced is possible due to the use of a new metric presented in this paper, the analysis of both *F-latency* and TaF results for eRisk dataset (Figures 3 and 4) provides a base to connect it with previous experiments. Besides, the use of one of the selection methods for the early detection systems that was already applied for the same Machine Learning models supplies the needed base for the required results interpretation.

In view of the conducted experiments and the results obtained, TaF metric is shown as an improved alternative to the early detection metrics. This means that this metric could be applied to evaluate any problem from the domain overcoming the limitations present on other time aware metrics. In terms of improvements of the early detection by machine learning models, it can be seen that a further exploration of selection methods could be applied. By expanding the ways the final decision is taken, performance might be improved.



(**a**) Ada Boost (**b**) Extra Tree

(**c**) Random Forest (**d**) Logistic Regression

Figure 4. Results for combinations of selectors and models (**a**) Forest (RF), Extra Tree (ET), Ada Boost (AB) and Logistic Regression (LR)) for each number of items analyzed with eRisk dataset using *F-latency* as metric.

Table 2. Results for combinations of selectors and models (Random Forest (RF), Extra Tree (ET), Ada Boost (AB) and Logistic Regression (LR)) for each number of items analyzed with Kitsune dataset.

Model	**Selector**	**1**	**2**	**3**	**4**	**5**	**25**	**50**	**75**	**100**	**125**	**150**	**175**	**200**	**225**	**300**
RF	bigger	0.9950	1.0000	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950
	2last	0.9950	0.0000	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950
	mean	0.9950	1.0000	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950
	last	0.9950	1.0000	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950
ET	bigger	0.9950	1.0000	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950
	2last	0.9950	0.0000	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950
	mean	0.9950	1.0000	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950
	last	0.9950	1.0000	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950
AB	bigger	0.9950	1.0000	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950
	2last	0.9950	0.0000	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950
	mean	0.9950	1.0000	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950
	last	0.9950	1.0000	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950	0.9950
LR	bigger	0.9896	0.9946	0.9896	0.9896	0.9896	0.9896	0.9896	0.9896	0.9896	0.9896	0.9896	0.9896	0.9896	0.9896	0.9896
	2last	0.9948	0.0000	0.9948	0.9948	0.9948	0.9948	0.9948	0.9948	0.9948	0.9948	0.9948	0.9948	0.9948	0.9948	0.9948
	mean	0.9896	0.9946	0.9896	0.9896	0.9896	0.9896	0.9896	0.9896	0.9896	0.9896	0.9896	0.9896	0.9896	0.9896	0.9896
	last	0.9896	0.9946	0.9896	0.9896	0.9896	0.9896	0.9896	0.9896	0.9896	0.9896	0.9896	0.9896	0.9896	0.9896	0.9896

5. Discussion

From the results obtained from this experiments we can extract two main conclusions. The first one is the strengths of *TaF* as metric for early detection problem evaluation, as it allows to better differentiate between outputs than *F-latency* metric. Being also problem dependent instead of datasets dependent, which was previously presented as a limitation for real world streaming environments evaluation.

Second conclusion extracted from this experiments is the ability of 2*last* as selection function to extract the best results when little information is known, being just overcome by bigger, when bigger amount of items were processed. As the objective of these systems is to detect as soon as possible this situation, the use of a *nlast* approach could improve general performance of systems. Problems detected when less than n items are being processed could be avoided by the combination of two different selection functions depending on the amount of items. Which could even be explored in future works with the analysis of the application of different selection functions to the *nlast* items processed.

6. Conclusions

Finally, from the research presented in this paper it can be extracted that due to the limitations of existent time aware metrics, the alternative provided by TaF contributes to a proper evaluation of detection systems in the early detection problem.

Also, and under the evaluation supplied by TaF metric, it is highlighted the importance of specific selection methods for early detection problem. A broader study could be performed with the inclusion of more selection methods and datasets with different particularities, but it is interesting to see how the number of items has an impact both on the penalisation and on the amount of information used by the system.

Author Contributions: Conceptualization, F.C. and M.L.-V.; Methodology, D.F. and F.J.N.; Software, M.L.-V.; Data curation, F.C.; Resources: D.F. and F.J.N.; Writing—original draft, M.L.-V.; Writing—review & editing, F.J.N., D.F. and F.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Ministry of Economy and Competitiveness of Spain and Fondo Europeo de Desarrollo Regional (FEDER) Funds of the European Union under Project PID2019-111388GB-I00; and in part by the Centro de Investigación de Galicia—Centro de Investigación en Tecnologías de la Información y las Comunicaciones (CITIC) Funded by Xunta de Galicia and the European Union (European Regional Development Fund–Galicia 2014-2020 Program), under Grant ED431G 2019/01.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Network data were obtained from dataset creators and are available at the DOI 10.24432/C5D90Q. Depression data were obtained from eRisk organization with their permission (https://erisk.irlab.org/2017/, accessed on 20 November 2023).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AB	AdaBoost
DDOS	Distributed Denial Of Service
ERDE	Early Risk Detection Error
ET	Extra Tree
FN	False Negative
FP	False Positive
LR	Logistic Regression
MDPI	Multidisciplinary Digital Publishing Institute
NIDS	Network Intrusion Detection Sytem

OS	Operative System
TaF	Time aware F-score
TaP	Time aware Precision
TP	True Positive
TN	True Negative
RF	Random Forest

References

1. Losada, D.E.; Crestani, F. A Test Collection for Research on Depression and Language Use. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction;* Springer: Cham, Switzerland, 2016; pp. 28–39. [CrossRef]
2. Mirsky, Y.; Doitshman, T.; Elovici, Y.; Shabtai, A. Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. In Proceedings of the Network and Distributed Systems Security (NDSS) Symposium 2018, San Diego, CA, USA, 18–21 February 2018.
3. Lopez-Vizcaino, M.F.; Novoa, F.J.; Fernandez, D.; Cacheda, F. Measuring Early Detection of Anomalies. *IEEE Access* **2022**, *10*, 127695–127707. [CrossRef]
4. Losada, D.E.; Crestani, F.; Parapar, J. eRisk 2020: Self-harm and Depression Challenges. In *Advances in Information Retrieval*; Springer International Publishing: Cham, Switzerland, 2020; pp. 557–563.
5. Sadeque, F.; Xu, D.; Bethard, S. Measuring the latency of depression detection in social media. In Proceedings of the WSDM 2018—11th ACM International Conference on Web Search and Data Mining, Marina Del Rey, CA, USA, 5–9 February 2018; pp. 495–503. [CrossRef]
6. Chinchor, N. MUC-4 Evaluation Metrics. In Proceedings of the 4th Conference on Message Understanding, McLean, VA, USA, 16–18 June 1992; pp. 22–29.
7. Samghabadi, N.S.; Monroy, A.P.L.; Solorio, T. Detecting Early Signs of Cyberbullying in Social Media. In Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, Marseille, France, 11–16 May 2020; European Language Resources Association (ELRA): Paris, France, 2020; pp. 144–149.
8. Hutchins, E.M.; Cloppert, M.J.; Amin, R.M. Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. *Lead. Issues Inf. Warf. Secur. Res.* **2011**, *1*, 113–125.
9. Narayanan, S.N.; Ganesan, A.; Joshi, K.; Oates, T.; Joshi, A.; Finin, T. Early detection of cybersecurity threats using collaborative cognition. In Proceedings of the 4th IEEE International Conference on Collaboration and Internet Computing, CIC 2018, Philadelphia, PA, USA, 18–20 October 2018; pp. 354–363. [CrossRef]
10. Pivarníková, M.; Sokol, P.; Bajtoš, T. Early-Stage Detection of Cyber Attacks. *Information* **2020**, *11*, 560. [CrossRef]
11. Xu, C.; Lin, H.; Wu, Y.; Guo, X.; Lin, W. An SDNFV-Based DDoS Defense Technology for Smart Cities. *IEEE Access* **2019**, *7*, 137856–137874. [CrossRef]
12. Privalov, A.; Lukicheva, V.; Kotenko, I.; Saenko, I. Method of Early Detection of Cyber-Attacks on Telecommunication Networks Based on Traffic Analysis by Extreme Filtering. *Energies* **2019**, *12*, 4768. [CrossRef]
13. Zhou, X.; Jain, A.; Phoha, V.V.; Zafarani, R. Fake News Early Detection: A Theory-driven Model. *Digit. Threat. Res. Pract.* **2020**, *1*, 12. [CrossRef]
14. Zhao, Z.; Resnick, P.; Mei, Q. Enquiring Minds: Early Detection of Rumors in Social Media from Enquiry Posts. In Proceedings of the 24th International World Wide Web Conference, Florence, Italy, 18–22 May 2015; International World Wide Web Conferences Steering Committee: Geneva, Switzerland, 2015; pp. 1395–1405. [CrossRef]
15. *Cyber Bullying: Common Types of Bullying 2019*; Statista: New York, NY, USA, 2019.
16. Royen, K.V.; Poels, K.; Daelemans, W.; Vandebosch, H. Automatic monitoring of cyberbullying on social networking sites: From technological feasibility to desirability. *Telemat. Inform.* **2015**, *32*, 89–97. [CrossRef]
17. Teng, T.H.; Varathan, K.D. Cyberbullying Detection in Social Networks: A Comparison Between Machine Learning and Transfer Learning Approaches. *IEEE Access* **2023**, *11*, 55533–55560. [CrossRef]
18. Yi, P.; Zubiaga, A. Session-based cyberbullying detection in social media: A survey. *Online Soc. Netw. Media* **2023**, *36*, 100250. [CrossRef]
19. Dhanya, K.A.; Vajipayajula, S.; Srinivasan, K.; Tibrewal, A.; Kumar, T.S.; Kumar, T.G. Detection of Network Attacks using Machine Learning and Deep Learning Models. *Procedia Comput. Sci.* **2023**, *218*, 57–66. [CrossRef]
20. Aitken, P.; Claise, B.; Trammell, B. *RFC 7011*; Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information; RFC Editor: Marina del Rey, CA, USA, 2013. [CrossRef]
21. Trammell, B.; Boschi, E. *RFC 5103*; Bidirectional Flow Export Using IP Flow Information Export (IPFIX); RFC Editor: Marina del Rey, CA, USA, 2008. [CrossRef]
22. Lopez-Vizcaino, M.; Novoa, F.J.; Fernandez, D.; Carneiro, V.; Cacheda, F. Early Intrusion Detection for OS Scan Attacks. In Proceedings of the 2019 IEEE 18th International Symposium on Network Computing and Applications, NCA 2019, Cambridge, MA, USA, 26–28 September 2019. [CrossRef]
23. Cacheda, F.; Fernández, D.; Novoa, F.J.; Carneiro, V. Analysis and Experiments on Early Detection of Depression. In Proceedings of the Conference and Labs of the Evaluation Forum, Avignon, France, 10–14 September 2018.

24. Cacheda, F.; Fernandez, D.; Novoa, F.J.; Carneiro, V. Early Detection of Depression: Social Network Analysis and Random Forest Techniques. *J. Med. Internet Res.* **2019**, *21*, e12554. [CrossRef] [PubMed]
25. Rafiq, R.I.; Hosseinmardi, H.; Mattson, S.A.; Han, R.; Lv, Q.; Mishra, S. Analysis and detection of labeled cyberbullying instances in Vine, a video-based social network. *Soc. Netw. Anal. Min.* **2016**, *6*, 88. [CrossRef]
26. Scikit-Learn: Machine Learning in Python—Scikit-Learn 1.1.2 Documentation. Available online: https://scikit-learn.org/stable/ (accessed on 20 December 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.