applied sciences

MDPI

Article

Improving Multi-Class Classification for Recognition of the Prioritized Classes Using the Analytic Hierarchy Process

Algimantas Venčkauskas, Jevgenijus Toldinas * and Nerijus Morkevičius

Department of Computer Science, Kaunas University of Technology, 44249 Kaunas, Lithuania; algimantas.venckauskas@ktu.lt (A.V.); nerijus.morkevicius@ktu.lt (N.M.)
* Correspondence: eugenijus.toldinas@ktu.lt

Abstract

Machine learning (ML) algorithms are widely used in various fields, including cyber threat intelligence (CTI), financial technology (Fintech), and intrusion detection systems (IDSs). They automate security alert data analysis, enhancing attack detection, incident response, and threat mitigation. Fintech is particularly vulnerable to cyber-attacks and cyber espionage due to its data-centric nature. Because of this, it is essential to give priority to the classification of cyber-attacks to accomplish the most crucial attack detection. Improving ML models for superior prioritized recognition requires a comprehensive strategy that includes data preprocessing, enhancement, algorithm refinement, and customized assessment. To improve cyber-attack detection in the Fintech, CTI, and IDS sectors, it is necessary to develop an ML model that better recognizes the prioritized classes, thereby enhancing security against important types of threats. This research introduces adaptive incremental learning, which enables ML models to keep learning new information by looking at changing data from a data stream, improving their ability to accurately identify types of cyber-attacks with high priority. The Analytical Hierarchy Process (AHP) is suggested to help make the best decision by evaluating model performance based on prioritized classes using real multi-class datasets instead of artificially improved ones. The findings demonstrate that the ML model improved its ability to identify prioritized classes of cyber-attacks utilizing the ToN_IoT network dataset. The recall value for the “injection” class rose from 59.5% to 61.8%, the recall for the “password” class increased from 86.7% to 88.6%, and the recall for the “ransomware” class improved from 0% to 23.6%.

Keywords: cyber threat intelligence; fintech; multi-class machine learning; prioritized class recognition; analytical hierarchy process; incremental learning

check for updates

Academic Editor: Rui Araújo

Received: 22 May 2025
Revised: 15 June 2025
Accepted: 20 June 2025
Published: 23 June 2025

Citation: Venčkauskas, A.; Toldinas, J.; Morkevičius, N. Improving Multi-Class Classification for Recognition of the Prioritized Classes Using the Analytic Hierarchy Process. *Appl. Sci.* **2025**, *15*, 7071. https://doi.org/10.3390/app15137071

1. Introduction

Machine learning (ML) algorithms find extensive applications in various fields, including cyber threat intelligence (CTI), financial technology (Fintech), and intrusion detection systems (IDSs). These algorithms enable the automation of security alert data analysis, which enhances the speed of attack detection, incident response, and the mitigation of threats or malicious activities [1,2]. Fintech, a significant player in the global economy and financial arena, places an emphasis on technological advancements that aim to automate financial processes. This is a sector that is particularly vulnerable to cyber-attacks and cyber espionage due to the fact that Fintech is a data-centric business that is expected to deliver services around the clock [3]. Enhancing an ML model for superior recognition of a prioritized class typically requires a comprehensive strategy, integrating data preprocessing and enhancement, algorithm refinement, and customized assessment. The application of artificial intelligence (AI) technology for the automatic identification and early warning of risks serves as a significant asset in supporting financial regulation. Consequently, future applications of ML in financial regulation must focus on evaluating strategies that balance the advantages and costs associated with algorithms [4]. Although machine learning greatly enhances security in the Fintech sector, it also presents certain challenges. Implementing machine learning models in security systems necessitates substantial high-quality data and computational resources, as well as continuous maintenance, to guarantee that models stay effective against evolving threats [5,6].

Multi-class data classification issues are a significant challenge in the field of intrusion detection [7,8]. Rigid ML models and thorough feature engineering are necessary for identifying and categorizing threats in IoT networks [9]. Real-time data processing systems produce substantial volumes of data that require classification. Prioritization in multi-class ML requires novel methodologies and the modification of current classification techniques [10]. ML algorithms assume that all misclassification errors made by a model, if not configured, are equivalent. Missing a positive or minority class case in the context of an intrusion detection issue is more detrimental than inaccurately classifying an example from the negative or majority class. As regards altering weights, the most straight-forward and widely used method of implementing cost-sensitive learning is to penalize the model less for training errors committed on examples from the minority class [11].

The concept of sorting probabilities is frequently applied in extreme multi-label classification to optimize particular loss functions, such as average precision loss, disattribution ranking loss, gradient harmonizing mechanism loss, and label distribution-aware margin loss [12,13]. An error-correcting output code (ECOC) model has been extensively researched and implemented across multiple domains, including computer vision, Fintech, and CTI [14]. The ECOC classification process necessitates a coding design that specifies the classes for the binary learners to train on, as well as a decoding scheme that outlines the method for aggregating the results (predictions) from the binary classifiers. In an ECOC code matrix, each row signifies a distinct class, while each column pertains to a specific class reassignment scheme.

Data preprocessing is an important part of creating strong ML models, which includes cleaning the data (getting rid of duplicates, fixing missing information, and correcting inconsistencies), creating features (turning categorical variables into a usable format), and dealing with class imbalance. Class imbalance can result in the distortion of the decision border, increasing classification errors for high-value data important for various domains, especially in Fintech [15]. Data preparation and enhancement address class imbalance. Algorithmic methods focus on modifying or creating machine learning algorithms to proficiently manage datasets, improving their capability to reliably classify instances from minority class [16]. Examples of algorithms that are typically resistant to class imbalance include Random Forests, Gradient Boosting Machines (XGBoost and LightGBM), and Support Vector Machines (SVMs) [17].

In many cases, researchers do not give priority to the classification of cyber-attacks in order to accomplish the most crucial attack detection in multi-class datasets. Instead, they divide the multi-class dataset into two classes and develop models that determine whether or not there is an attack (see Section 2 for more information). Resolving the mentioned problems will make multi-class prioritized classification more accurate, which will improve ML model performance. This work presents adaptive incremental learning, which enables ML models to keep learning new information by looking at changing data from a data stream, and suggests using the Analytical Hierarchy Process (AHP) to make the best decision based on how well the model performs with prioritized classes. The important contributions of this research are listed below.

- Its novelty is using adaptive incremental learning, which allows ML models to learn new information continuously by analyzing changing data from a data stream, to design an improved ML model for the more accurate recognition of the prioritized cyber-attack classes.
- When multi-class classification is used, performance metrics for machine learning models are calculated for each class individually, and it is quite common when a single characteristic increases for one class and decreases for the other class. In such cases, it is difficult to decide if the model changes are acceptable. To mitigate these shortcomings of the composite scores, we propose the use of the Analytical Hierarchy Process (AHP) to make the right decision based on the model performance evaluation according to prioritized classes.

We organize the remainder of this article as follows. Section 2 discusses related work. Section 3 presents and explains the proposed approach. Section 4 presents and discusses experimental settings and results. Finally, Section 5 presents the discussion, and Section 6 presents the conclusion.

2. Related Work

Some authors simplify their solutions and obtain better model performance results by applying binary-class classification methods while using multi-class datasets. The utilization of a machine learning model that is based on multilayer perceptron (MLP) techniques in order to detect cyber-attacks is proposed in [18]. This is accomplished by utilizing the NF-ToN-IoT dataset, which includes a collection of various cyber-attacks. Through the utilization of a multi-class dataset, the suggested MLP model has been trained and tuned to differentiate between normal and malicious behavior. The effectiveness of the MLP model is demonstrated by the results, achieving an accuracy of more than 90 percent during both the training and validation phases. The TON_IoT multi-class unbalanced network dataset was utilized in [19]. The SMOTETomek approach was employed to address the problem of class imbalance. The model was constructed using the Stochastic Gradient Descent (SGD) optimizer and the binary cross-entropy loss function. It achieved an accuracy of 91% for SVM and 98% for Random Forest.

Employing the NF-ToN-loT multi-class dataset as a benchmark, the suggested model [20] significantly surpasses previous models, underscoring its effectiveness and progress in the domain. The evaluated optimizers comprise Adam, RMSprop, Adagrad, and SGD, with Adam emerging as the most efficacious optimizer for this particular intrusion detection task on the NF-ToN-IoT dataset. We achieved 99.04% accuracy using binary classification. A multi-class data stream from a real-world GSP system was utilized in [21]. This paper proposes a unique framework that integrates CNN and autoencoder-based LSTM methods to enhance anomaly detection in Industrial IoT networks. The authors illustrate that the two-step sliding window method can enhance temporal feature extraction via LSTM, which aggregates the outputs of CNNs within the integrated framework. Anomaly detection was approached as a binary classification problem in the trials undertaken, yielding accuracies of 92.63%, 86.66%, and 89.21% for KNN, RF, and SVM, respectively.

The BiLSTM is proposed [22] for anomaly-based intrusion detection systems in IoT networks. The BoT-IoT and ToN-IoT datasets are utilized for the identification of IDS in IoT networks. The gain ratio is utilized for picking optimal features, offering advantages over information gain when it is not biased towards any one feature. The studies conducted by the authors show the findings of binary classification, despite the fact that the datasets employed are multi-class. For the BoT-IoT dataset, the BiLSTM obtains an accuracy of 98.76%, whereas for the ToN-IoT dataset, it achieves 96.84% accuracy.

Researchers in [23] performed tests on the UNSW-NB15 dataset, which comprises ten classes, including one normal class, three major classes, and six minor classes, within a multi-classification framework. To generate believable synthetic data for minor attack traffic, the authors made use of a state-of-the-art generative model. The authors specifically focused on reconstruction errors, Wasserstein distance-based generative adversarial networks, and autoencoder-driven deep learning models. The accuracy achieved for multi-class classification is 82.0% for Generic (number of records in dataset is 22.81%), 50.2% for Exploits (number of records in dataset is 19.04%), 81.9% for Fuzzers (number of records in dataset is 10.37%), 29.1% for DoS (number of records in dataset is 6.99%), 51.3% for Reconnaissance (number of records in dataset is 5.98%), 77.5% for Analysis (number of records in dataset is 1.14%), 91.5% for Backdoors (number of records in dataset is 0.99%), 58.9% for Shellcode (number of records in dataset is 0.65%), and 56.8% for Worms (number of records in dataset is 0.07%), where the number of Normal records is 31.94%. The authors draw the conclusion that while the suggested framework does enhance classification performance, the issue of relatively low detection rates for certain classes remains. More specifically, the DoS class had comparatively poor detection rates across all tested models.

The CIDDS-001 dataset employed in [24] utilizes the AttackType label to train machine learning algorithms, including RF, MLP, and LSTM, to accurately categorize a network flow as either benign or as displaying DoS, Brute Force, Ping Scan, or Port Scan attacks. The models employed were implemented from two separate perspectives: single-flow and multi-flow. The outcomes were compared to ascertain which model was more appropriate for the specific case. From the single-flow perspective, the models exclusively focused on individual flow characteristics, with RF attaining the highest performance, evidenced by the F1-score of 85.04%. Conversely, the multi-flow perspective examined both the distinct characteristics of individual flows and the short- to long-term interrelationships among the flows within a certain sequence. The LSTM model achieved the optimal F1-score of 91.66% with a sequence size of 70.

Based on the summary presented in Table 1, we can make the following assumptions:

- One of the biggest shortcomings is the situation in which one or more classes, known as minority classes, are significantly less frequent than other classes, known as majority classes. This will result in a distortion of the decision border, hence increasing the classification error for high-value attack data.
- The classification approaches proposed by the other authors are not oriented to correctly classify prioritized classes; instead, all multi-class classes are combined in two classes (normal and abnormal), and binary classification approaches are used to differentiate between normal and abnormal behavior,
- A new method needs to be proposed if we ar3 to create an ML model that is oriented towards better recognition of the important classes that have been prioritized.

These assumptions guided us in the development of the proposed method to optimize a machine learning model for improving the recognition of prioritized classes using the AHP.

Table 1. A list of state-of-the-art research papers on imbalanced multi-class dataset classification methods.

Reference	Problem	Solution	Dataset	Details
Taylor et al. [18]	A challenge in creating an ML-based detection system that uses heterogeneous network data samples.	Employing a multilayer perception (MLP) model, a type of feedforward artificial neural network.	NF-ToN-IoT 10 classes 19.6% benign 80.4% various attacks	Binary classification.
Maseno et al. [19]	Capacity of deep learning algorithms in the selection of optimal feature subset in the field of IDS.	Used CNN's potential for selecting the best feature subsets to enhance intrusion detection.	TON_IoT SMOTEtomek applied to solve the issue of class imbalance	Binary classification.
Bhuiyan et al. [20]	A key research gap is the reliance on outdated datasets for training and evaluation.	An innovative DNN model introduced for efficient detection of stealthy and polymorphic variants of network intrusions. Conducts an ablation study to dissect the components of the DNN model.	NF-ToN-loT NF-UNSW-NB15-v2 4 classes 2,390,275 flows, with 3.98% attacks	Binary classification.
Khan et al. [21]	Effective anomaly recognition and explanation are critical for ensuring quality services in IIOT.	The phenomenon of sliding windows is utilized for the processing of time-series data.	Industrial Control System (ICS) dataset 8 classes 214,580 normal 60,048 anomalous	Binary classification.
Kumar et al. [22]	Anomaly-based IDS in IoT network.	The gain ratio is used for feature selection and BiLSTM is used for classification.	BoT-IoT ToN-IoT 300,000 normal 162,043 attacks	Binary classification ToN-IoT
Park et al. [23]	The problem of data imbalance, in which AI models cannot sufficiently learn malicious behavior and thus fail to detect network threats accurately.	Focused on the reconstruction error and Wasserstein distance-based generative adversarial networks, and autoencoder-driven deep learning models.	NSL-KDD 4 classes 77,052 normal 71,464 attacks UNSW-NB15 10 classes 93,000 normal 164,673 attacks	Binary classification and Multi-classification Generated synthetic data for each class via the generative model
Oliveira et al. [24]	Addressed the CIDDS-001 dataset using the AttackType label to train machine learning methods.	Single-flow (%) recall 71.4–95.68 Multi-flow (%) recall 85.65–89.71	CIDDS-001	Multi-class
Kollu et al. [25]	Cloud-based smart contract analysis in Fintech.	Integrated federated learning in intrusion detection	KDDCUP99, ISCX, and NSL-KDD	Two-class and four-class categories
Alamsyah et al. [26]	Innovative credit risk assessment.	Integrated social media analytics within credit scoring systems.	Data collected from LinkedIn	Binary classification

3. Improving Multi-Class Classification for Recognition of the Prioritized Classes Using the AHP

In this section, we present and explain the main features of the proposed method. A reliable method for evaluating model performance is essential in several areas of adaptable or customizable ML. Some solutions involving federated learning use model performance metrics to decide if the newly received model updates are desirable in terms of the overall global model improvement or not. Authors [27] use performance evaluation to distinguish malicious members of the federation and to eliminate potentially harmful model updates from the global model. A similar approach is also valid for transfer learning, when the initial model is additionally trained using a local dataset to achieve a better-behaving final model suitable for the very specific custom requirements of the final user.

Incremental learning is an ML method that allows models, like deep learning models, to keep learning new information continuously by analyzing changing data they receive from a data stream [28]. Through the use of incremental learning, it is possible to develop AI systems that are capable of continuously updating themselves to include new information while preserving the information they already possess.

Adaptive training is orchestrated based on performance metrics in the specific field only relevant for the concrete customer. Adaptive incremental learning also requires reliable metrics of the model's performance to decide if it is desirable to include the newly acquired data in the model. If the two-class classification is used, then classical composite scores, such as accuracy, precision, recall, and F1-score, may be used as model performance metrics. The situation dramatically changes when multi-class classification is used. In such a scenario, precision, recall and F1-score are calculated for each class individually, and it is quite common when a single characteristic increases for one class and decreases for the other class. The problem arises from the natural weaknesses of the combined scores, which often lead to confusing results when trying to improve and assess prediction models. In such cases, it is very difficult to decide whether the potential changes to the model are desirable or not. To mitigate the shortcomings of the composite scores, we propose using the AHP [29,30] to evaluate the performance of the model and make the decision. The proposed method workflow is depicted in Figure 1.

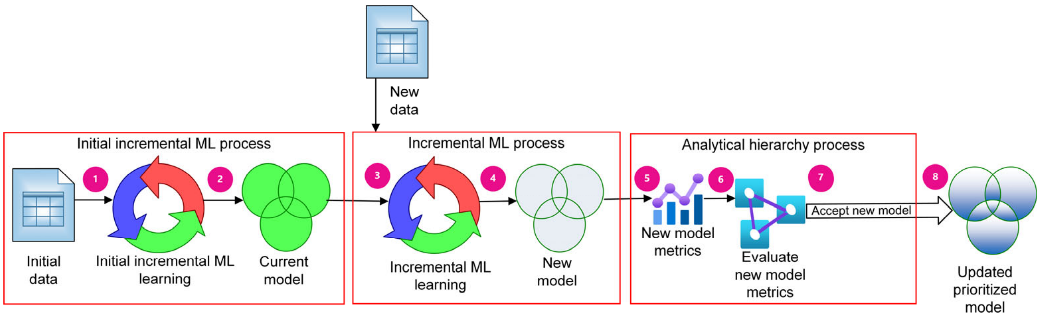

New data
Initial incremental ML process
1
2
Initial data
Initial incremental ML learning
Current model
Incremental ML process
3
4
Incremental ML learning
New model
Analytical hierarchy process
5
6
7
New model metrics
Evaluate new model metrics
Accept new model
8
Updated prioritized model

Figure 1. The AHP is utilized in the adaptive incremental ML process.

The main steps of the proposed incremental learning process are as follows:

1. Use the initial available data and train the incremental ML model.
2. As a result of the first training, the initial model is produced, and initial performance characteristics are calculated. The initial model is marked as the current model.
3. After the new batch of training data is available, train the current version of the ML model using the new data.
4. As a result of step 3, the new ML model is produced, and performance characteristics are obtained.

5. Present the performance metrics of both the existing model and the new model for the AHP decision-making process.
6. Load the application-specific AHP judgment matrix with the results of the pairwise comparison of the metrics receiving prioritization;
7. Use the AHP process and make decisions to include (or not to include) the new data in the prioritized model. If the decision is to not include the new data, then the new ML model is discarded. Otherwise, the new ML model is marked as the current model.
8. Repeat the process starting, from step 3, until no new data is available. The current ML model becomes the final updated prioritized model after processing all batches of data.

The AHP enables judgments to be made based on pairwise comparisons of many criteria that are carried out by specialists in the subject. The AHP has the potential to be modified for use in machine-based decision-making processes that include complicated multi-criteria challenges, such as security and machine learning, among others [31,32]. The choice of the AHP is based on the following characteristics of the method [33]:

- The AHP provides the ability to check the consistency of the evaluations provided by decision makers;
- The AHP allows the use of heterogeneous measurement scales for different criteria;
- The AHP eases multi-objective decision-making by using exclusively pairwise comparisons of the alternatives, which provides increased reliability of the results;
- The most controversial step of manual weight assignment to different criteria is avoided.

A three-level hierarchical structure for the AHP, aimed at the evaluation of the machine learning model's performance, is presented in Figure 2.

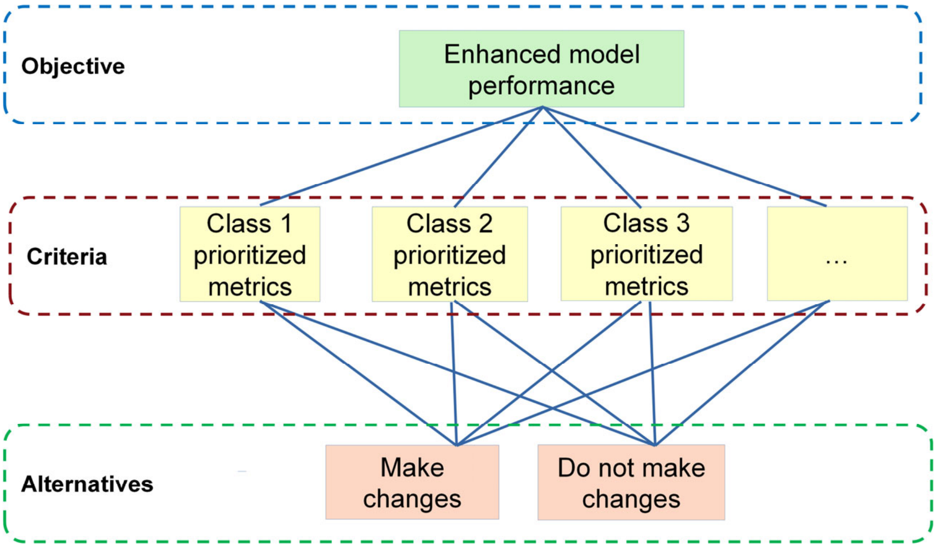

Objective
Enhanced model performance
Criteria
Class 1 prioritized metrics
Class 2 prioritized metrics
Class 3 prioritized metrics
…
Alternatives
Make changes
Do not make changes

Figure 2. A hierarchical framework for the AHP.

Level one is the objective of the process, which in our case is to improve (or at least to not decrease) the performance of the ML model. The second level is the criteria. This level is based on the personal preferences of the final user of the model and is completed by the field experts. The main input used in this step is the criteria pairwise comparison matrix, also called the judgment matrix, which is manually generated before the AHP process is started. For example, assume that the intrusion detection dataset involved in the machine learning process contains four classes: Normal, DDoS, Worms, and Exploits. Then four criteria should be considered at level 2 of AHP. The 4 × 4 judgment matrix should be constructed by the experts, comparing the importance of each class in the context of the concrete infrastructure. The learning process begins with the provision of the judgment matrix, which includes all the necessary weights for each criterion. The third level is alternatives. These always include only two because the AHP is used to decide whether to include (or not) the newly calculated changes into the main ML model.

The AHP process compares both alternatives using each criterion in Level 2 separately. This is an effortless task, as all criteria used in ML are numerical, and the process of constructing judgment matrices in Level 2 is based on comparing numerical estimates of the criteria and converting the differences into numerical values suitable for the AHP (i.e., the real numbers in the range of 0–9). The whole process of ML model evaluation is summarized in Figure 3.

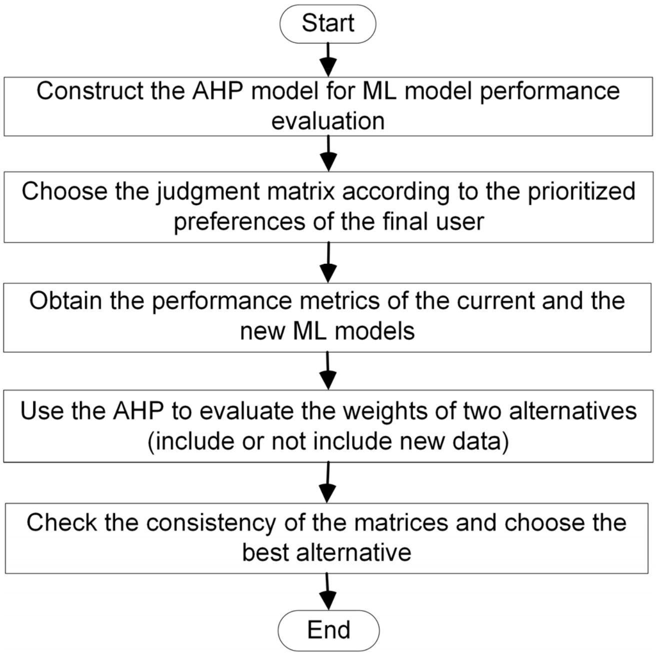

Start
Construct the AHP model for ML model performance evaluation
Choose the judgment matrix according to the prioritized preferences of the final user
Obtain the performance metrics of the current and the new ML models
Use the AHP to evaluate the weights of two alternatives (include or not include new data)
Check the consistency of the matrices and choose the best alternative
End

Figure 3. The proposed method of the AHP decision-making process.

One of the intriguing application areas of the AHP score-based machine learning model performance evaluation is multi-criteria adaptive incremental learning. The main steps in applying AHP in the process of adaptive incremental learning are as follows:

1. Construct an AHP framework according to the structure provided in Figure 2 and include all the criteria suitable for the concrete dataset and application area.
2. Load the judgment matrix representing the priorities of the concrete organization with the results of pairwise comparisons of all criteria.
3. Obtain the performance characteristics of the current and the new machine learning models.
4. For each criterion, repeat the following:
    - Construct the weight coefficient matrix using both alternatives (i.e., metrics of the current and the new ML model). The elements of the matrix are calculated using a comparison function, which compares performance metrics of the current and the new model and converts them into real numbers from the interval (0, 9].
5. Provide all the created matrices with the standard AHP decision-making method to obtain the estimated weights of both alternatives.
6. Check the consistency of the matrices using AHP consistency indicators. Choose the best alternative as the final decision.

We suggest using the score function $scr(x)$ to compare the performance metrics of individual ML models and to create the judgment matrices needed in step 5:

$$scr(x) = \begin{cases} \frac{1}{9} & when\ x \leq 0.5 \\ \frac{1}{2} \cdot 22^{x} & when\ -0.5 < x \leq -0.001 \\ 1 & when\ -0.001 < x \leq 0.001 \\ 2 \cdot 22^{x} & when\ 0.001 < x \leq 0.5 \\ 9 & when\ x > 0.5 \end{cases}, \tag{1}$$

where x, and $x = nv - ov$ is the difference between the new (nv) and the old (ov) value of the criterion. The judgment matrix for the concrete class (criterion) J_{cr} is then constructed in the following way:

$$J_{cr} = \begin{pmatrix} 1 & scr(nv_{cr} - ov_{cr}) \\ \frac{1}{scr(nv_{cr} - ov_{cr})} & 1 \end{pmatrix}, \tag{2}$$

where nv_{cr} is the new value of the criterion cr, and ov_{cr} is the old value of the corresponding criterion. The main assumptions when using such an expression are that changes in the main characteristics of the ML models rarely exceed a few tenths, so a change of 0.5 or more is represented by the maximal possible value in the AHP judgment matrix (i.e., 9). On the other hand, small changes are represented using exponential law, which slightly emphasizes small changes and makes the learning process more responsive to minimal changes in performance metrics. For example, consider a dataset containing four classes (Normal, DDoS, Worms, and Exploits). Assume that, after the inclusion of the newly acquired data, the recall value changes from the original $(0.501, 0.975, 0.833, 0.789)^{T}$ to the new values $(0.397, 0.980, 0.908, 0.762)^{T}$. Then, the corresponding judgment matrices should look like this:

$$\begin{gathered} J_{Normal} = \begin{pmatrix} 1 & scr(0.397 - 0.501) \\ 1/scr(0.397 - 0.501) & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0.363 \\ 2.758 & 1 \end{pmatrix}, \\ J_{DDoS} = \begin{pmatrix} 1 & scr(0.980 - 0.975) \\ 1/scr(0.980 - 0.975) & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2.031 \\ 0.492 & 1 \end{pmatrix}, \ldots. \end{gathered} \tag{3}$$

Although the score function presented in Equations (1) and (2) was used in all experiments presented in this paper, one could choose another function for the conversion of machine learning performance metric changes into a judgment matrix. The flow of the learning process stays the same.

4. Experimental Settings and Results

In this section, we show the results from testing the AHP framework, which uses adaptive incremental learning to build a strong model for the most important prioritized classes.

4.1. Dataset

Upon analyzing publicly accessible benchmark datasets, we have selected processed network traffic datasets from the newly created TON_IoT collection of datasets. In order to simulate the complexity and scalability of Industrial Internet of Things and Industry 4.0 networks, the Internet of Things lab developed a new testbed that can connect a large number of virtual machines, physical systems, hacking platforms, cloud and fog platforms, Internet of Things, and Industrial Internet of Things sensors [34,35]. A summary of the main characteristics of the ToN_IoT network traffic dataset is presented in Table 2.

Table 2. The main characteristics of ToN_IoT network traffic dataset.

	Full		**Training**		**Testing**	
Class	**Count**	**%**	**Count**	**%**	**Count**	**%**
backdoor	508,116	2.275	409,764	2.276	98,352	2.267
ddos	6,165,008	27.597	4,967,755	27.599	1,197,253	27.593
dos	3,375,328	15.110	2,719,941	15.111	655,387	15.104
injection	452,659	2.026	364,816	2.027	87,843	2.024
mitm	1052	0.005	859	0.005	193	0.004
password	1,718,568	7.693	1,384,077	7.689	334,491	7.709
ransomware	72,805	0.326	58,548	0.325	14,257	0.329
scanning	7,140,161	31.963	5,752,791	31.960	1,387,370	31.974
xss	2,108,944	9.441	1,699,502	9.442	409,442	9.436
normal	796,380	3.565	641,947	3.566	154,433	3.559
Total	**22,339,021**		**18,000,000**		**4,339,021**	

Columns titled "%" represent the percentage of the records of the particular class in the pool of the corresponding dataset part. This dataset is heavily unbalanced, i.e., "mitm" attack records compose only 0.005 percent of the total count of all records. The original data was preprocessed using the following steps: unique values for each non-numeric column were extracted; then dictionary functionality was used to replace all string values with the corresponding numeric indices from the dictionaries. This method processed 27 columns in total. We did not make any further modifications. We have not tried to modify the data and artificially balance the dataset because, in real-life application scenarios, the learning process can only use the data that is physically "on the wire," and this data is inherently unbalanced.

4.2. Experimantal Evaluation of the Proposed Method

To experimentally evaluate the performance of the proposed method, we used the ToN_IoT network dataset. In our test scenario, all ten classes of network traffic are observed, but enterprise infrastructure is very sensitive to "injection," "password," and "ransomware" attacks. Moreover, the recall parameter, which represents the percentage of the accurately identified observations for each actual class of traffic, is very important. Security experts at the enterprise compared the sensitivity to all types of attacks and obtained the AHP judgment matrix presented in Equation (4). All experimental evaluations, including AHP score-based adaptive learning, utilized this matrix.

$$J = \begin{pmatrix} 1 & 1 & 2 & 1/5 & 2 & 2 & 1/8 & 1/7 & 1 & 1/3 \\ 1 & 1 & 1 & 1/8 & 1/2 & 3 & 1/5 & 1/7 & 1/2 & 1/3 \\ 1/2 & 1 & 1 & 1/8 & 1/2 & 3 & 1/5 & 1/7 & 1/2 & 1/3 \\ 5 & 8 & 8 & 1 & 3 & 9 & 1 & 1 & 7 & 8 \\ 1/2 & 2 & 2 & 1/3 & 1 & 1 & 1/8 & 1/9 & 2 & 1 \\ 1/2 & 1/3 & 1/3 & 1/9 & 1 & 1 & 1/8 & 1/9 & 1/2 & 1/3 \\ 8 & 5 & 5 & 1 & 8 & 8 & 1 & 1 & 6 & 7 \\ 7 & 7 & 7 & 1 & 9 & 9 & 1 & 1 & 9 & 8 \\ 1 & 2 & 2 & 1/7 & 1/2 & 2 & 1/6 & 1/9 & 1 & 2 \\ 3 & 3 & 3 & 1/8 & 1 & 3 & 1/7 & 1/8 & 1/2 & 1 \end{pmatrix} \tag{4}$$

The consistency ratio (CR) of the judgment matrix presented in Equation (4) is 6.26%, which is sufficient for AHP (a CR of less than 10% is required).

A classification algorithm's ultimate objective is to obtain the highest possible classification accuracy for a problem. It is important to note that there is no classification algorithm that is totally suitable for all issues. The goal of a classification algorithm is to optimize the classification process. The error-correcting output code (ECOC) technique is a method that is able to overcome the difficulty of classification, particularly in situations where there are significant numbers of classes [36]. We used the MATLAB (v. R2024b) multi-class ECOC classification model, which employs binary learners for the incremental learning function, to evaluate how the proposed method influences the results of adaptive incremental learning.

To initiate incremental learning, the initial step must involve the inclusion of the names of all expected classes or the maximum number of classes that are expected to be present in the data during the process of incremental learning. The ToN_IoT network traffic dataset was divided randomly, allocating 80% for the training dataset and 20% for the testing dataset. The calculation of the total number of records in the training dataset resulted in a figure of 18,000,000. To facilitate incremental learning, the entire training dataset was divided into chunks. The selection of 36 chunks was based on empirical analysis, with the aim of maintaining a consistent record count of 500,000 in each training chunk. We derived this figure by dividing the total number of records from the training dataset by 36. We performed incremental learning in 36 steps using data chunks of 500,000 records. After each step, the performance characteristics of the model were calculated using the test dataset, which included 4,339,021 records.

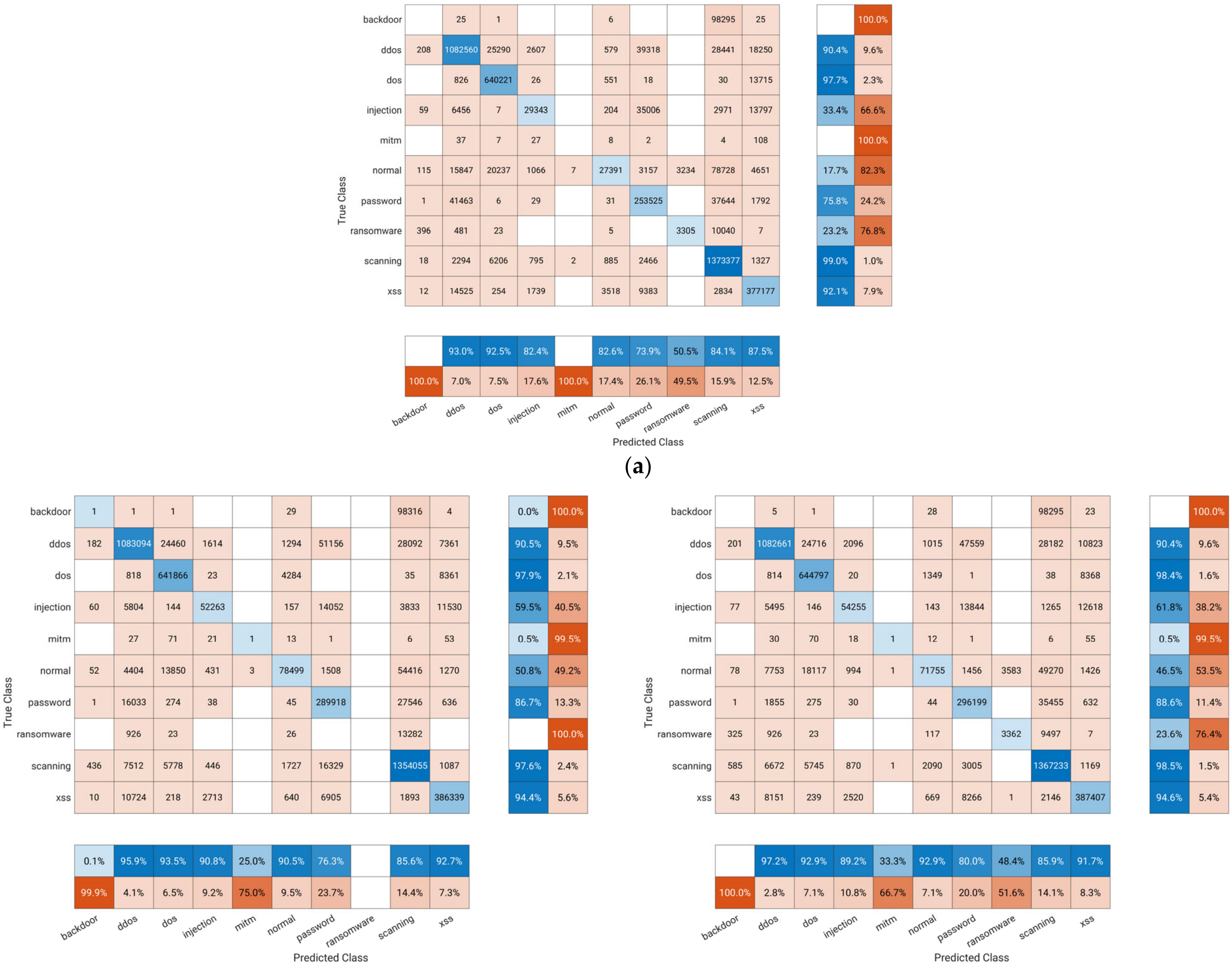
The main performance metrics used for ML models are accuracy, precision, recall, and F1. Accuracy is the number of correct predictions compared to the total expected predictions. Precision is the measure of true positives compared to all predicted positives. Recall (known as true positive rate or sensitivity) shows how many actual positive cases were correctly identified. F1 is the harmonic mean of precision and recall. Researchers use accuracy metrics for balanced datasets and F1-scores for imbalanced datasets.

In our experiments, an original ToN_IoT network traffic dataset was used, but the main goal was to create an ML model that can predict the most important classes with high priority. Due to that reason, recall, defined here as the percentage of positive cases accurately anticipated, is the most appropriate performance metric for the method we have proposed.

Two learning scenarios were used. The first was traditional unconditional incremental learning using all available data. The second scenario employed AHP decision-making to determine the preference for including each data chunk in the updated prioritized model. The main criterion used for AHP's evaluation of performance was the recall metric. The results of this experiment are summarized in Figures 4 and 5 and Table 3.

Figure 4a shows the confusion matrix after the first step. Both scenarios follow the same step, using the initial data chunk to initiate incremental model training. Figure 4b,c show the confusion matrices after the last step. These were created using unconditional and AHP-based adaptive incremental learning, respectively. One can observe that the use of AHP-based adaptive incremental learning slightly improves the recall results of the most prioritized classes according to our test scenario. The recall value of the "injection" class increased from 59.5% to 61.8%, the recall of the "password" class increased from 86.7% to 88.6%, and the recall of the "ransomware" class increased from 0 to 23.6%. On the other hand, only some of the performance characteristics of the classes "not so very prioritized" decreased (i.e., recall of "ddos" decreased from 90.5% to 90.4%).

Table 3 presents a more detailed, step-by-step flow of the AHP-based adaptive incremental learning process.



	backdoor	ddos	dos	injection	mitm	normal	password	ransomware	scanning	xss		
backdoor		25	1			6			98295	25		100.0%
ddos	208	1082560	25290	2607		579	39318		28441	18250	90.4%	9.6%
dos		826	640221	26		551	18		30	13715	97.7%	2.3%
injection	59	6456	7	29343		204	35006		2971	13797	33.4%	66.6%
mitm		37	7	27		8	2		4	108		100.0%
normal	115	15847	20237	1066	7	27391	3157	3234	78728	4651	17.7%	82.3%
password	1	41463	6	29		31	253525		37644	1792	75.8%	24.2%
ransomware	396	481	23			5		3305	10040	7	23.2%	76.8%
scanning	18	2294	6206	795	2	885	2466		1373377	1327	99.0%	1.0%
xss	12	14525	254	1739		3518	9383		2834	377177	92.1%	7.9%
		93.0%	92.5%	82.4%		82.6%	73.9%	50.5%	84.1%	87.5%		
	100.0%	7.0%	7.5%	17.6%	100.0%	17.4%	26.1%	49.5%	15.9%	12.5%		

(**a**)

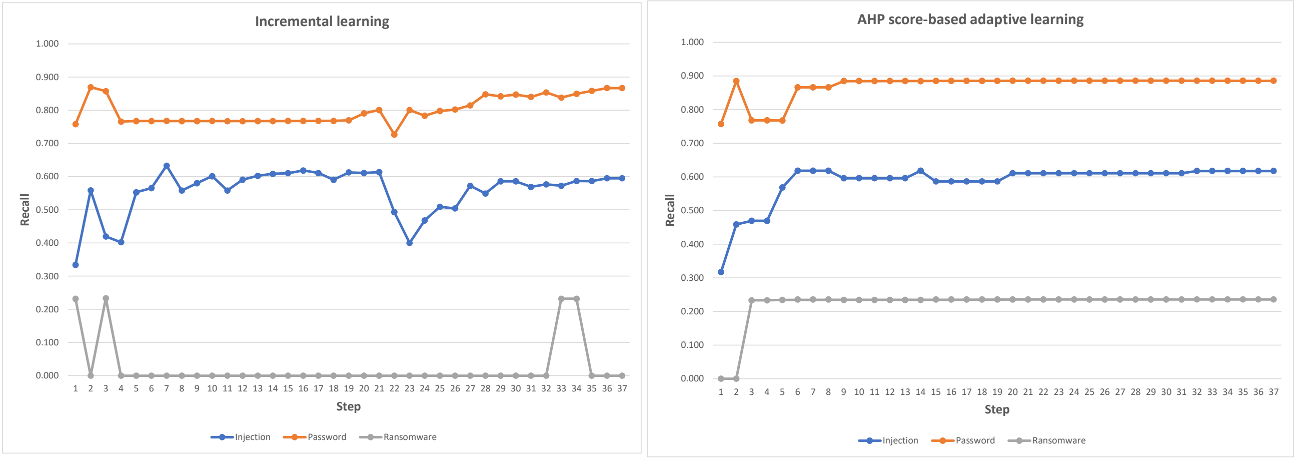
	backdoor	ddos	dos	injection	mitm	normal	password	ransomware	scanning	xss		
backdoor	1	1	1			29			98316	4	0.0%	100.0%
ddos	182	1083094	24460	1614		1294	51156		28092	7361	90.5%	9.5%
dos		818	641866	23		4284			35	8361	97.9%	2.1%
injection	60	5804	144	52263		157	14052		3833	11530	59.5%	40.5%
mitm		27	71	21	1	13	1		6	53	0.5%	99.5%
normal	52	4404	13850	431	3	78499	1508		54416	1270	50.8%	49.2%
password	1	16033	274	38		45	289918		27546	636	86.7%	13.3%
ransomware		926	23			26			13282			100.0%
scanning	436	7512	5778	446		1727	16329		1354055	1087	97.6%	2.4%
xss	10	10724	218	2713		640	6905		1893	386339	94.4%	5.6%
	0.1%	95.9%	93.5%	90.8%	25.0%	90.5%	76.3%		85.6%	92.7%		
	99.9%	4.1%	6.5%	9.2%	75.0%	9.5%	23.7%		14.4%	7.3%		

(**b**)

	backdoor	ddos	dos	injection	mitm	normal	password	ransomware	scanning	xss		
backdoor		5	1			28			98295	23		100.0%
ddos	201	1082661	24716	2096		1015	47559		28182	10823	90.4%	9.6%
dos		814	644797	20		1349	1		38	8368	98.4%	1.6%
injection	77	5495	146	54255		143	13844		1265	12618	61.8%	38.2%
mitm		30	70	18	1	12	1		6	55	0.5%	99.5%
normal	78	7753	18117	994	1	71755	1456	3583	49270	1426	46.5%	53.5%
password	1	1855	275	30		44	296199		35455	632	88.6%	11.4%
ransomware	325	926	23			117		3362	9497	7	23.6%	76.4%
scanning	585	6672	5745	870	1	2090	3005		1367233	1169	98.5%	1.5%
xss	43	8151	239	2520		669	8266	1	2146	387407	94.6%	5.4%
		97.2%	92.9%	89.2%	33.3%	92.9%	80.0%	48.4%	85.9%	91.7%		
	100.0%	2.8%	7.1%	10.8%	66.7%	7.1%	20.0%	51.6%	14.1%	8.3%		

(**c**)

Figure 4. Comparison of confusion matrices using unconditional and AHP-based adaptive incremental learning: (**a**) confusion matrix after the first step, which is the same in both cases; (**b**) confusion matrix after the final step when using unconditional incremental learning; (**c**) confusion matrix after the final step when using adaptive incremental learning based on AHP.



(**a**) (**b**)

Figure 5. Comparison of changes in three most prioritized recall values during incremental learning process: (**a**) unconditional incremental learning using all available data; (**b**) adaptive incremental learning based on AHP score.

Table 3. Adaptive incremental learning summary. Recall values presented for the most prioritized classes and AHP-based decisions.

Step	The Important Attack Classes That Have Been Prioritized			AHP Score	Include the New Data	Training Time (s)	AHP Evaluation Time (ms)
	Injection	Password	Ransomware				
1	0.317	0.757	0.000		Yes	4.45	
2	0.459	0.885	0.000	0.62	Yes	4.38	1.05
3	0.469	0.768	0.233	0.56	Yes	4.32	0.42
4	0.435	0.765	0.000	0.34	No	4.32	0.42
5	0.568	0.768	0.234	0.57	Yes	4.39	0.42
6	0.618	0.866	0.235	0.65	Yes	4.36	0.50
7	0.598	0.885	0.000	0.43	No	4.37	0.36
8	0.442	0.767	0.000	0.31	No	4.36	0.36
9	0.596	0.885	0.234	0.51	Yes	4.35	0.35
10	0.600	0.885	0.000	0.48	No	4.33	0.35
11	0.450	0.767	0.234	0.40	No	4.33	0.35
12	0.613	0.835	0.000	0.42	No	4.32	0.38
13	0.590	0.885	0.235	0.48	No	4.31	0.38
14	0.618	0.885	0.234	0.57	Yes	4.32	0.36
15	0.586	0.885	0.235	0.51	Yes	4.35	0.35
16	0.601	0.885	0.000	0.46	No	4.30	0.35
17	0.600	0.885	0.000	0.47	No	4.35	0.36
18	0.559	0.771	0.236	0.39	No	4.37	0.35
19	0.594	0.886	0.000	0.46	No	4.33	0.35
20	0.611	0.886	0.236	0.56	Yes	4.29	0.35
21	0.606	0.886	0.000	0.37	No	4.32	0.35
22	0.416	0.805	0.000	0.28	No	4.26	0.38
23	0.587	0.885	0.236	0.45	No	4.31	0.35
24	0.560	0.842	0.000	0.32	No	4.36	0.35
25	0.572	0.863	0.000	0.32	No	4.40	0.35
26	0.595	0.886	0.236	0.45	No	4.35	0.36
27	0.598	0.885	0.000	0.36	No	4.36	0.35
28	0.614	0.886	0.000	0.45	No	4.34	0.35
29	0.606	0.876	0.236	0.41	No	4.43	0.37
30	0.565	0.885	0.222	0.36	No	4.39	0.35
31	0.549	0.788	0.084	0.31	No	4.37	0.35
32	0.618	0.886	0.236	0.53	Yes	4.36	0.37
33	0.516	0.814	0.097	0.31	No	4.39	0.35
34	0.599	0.885	0.093	0.39	No	4.43	0.35
35	0.602	0.875	0.309	0.47	No	4.37	0.36
36	0.596	0.872	0.000	0.32	No	4.37	0.35
37	0.618	0.886	0.236				

This table shows the recall values after each step of adaptive incremental learning for the important attack classes that were prioritized (e.g., injection, password, and ransomware). The column titled “AHP score” indicates the score related to the decision to include the last chunk of new data in the updated prioritized ML model or not include it. The decision is positive if the AHP score is >0.5, as only two alternatives exist. The “Include the new data” column signifies the incorporation of the new data into the updated prioritized ML model. The last row (no. 37) shows the results of the updated prioritized ML model, which are the same as the results after step no. 32, because the new data from the last four chunks were not added to the updated prioritized ML model since their AHP scores were below 0.5. Table 4 presents other metrics for comparison.

Table 4. The recall, precision, and F1-score values of the most prioritized classes from the first and the last incremental learning steps.

The Important Attack Classes That Have Been Prioritized	Step	Recall	Precision	F1-Score
Injection	1	0.317	0.830	0.459
	37	0.618	0.892	0.730
Password	1	0.757	0.734	0.745
	37	0.886	0.800	0.840
Ransomware	1	0.000	0.000	0.000
	37	0.236	0.484	0.317

In Table 4, we can see that other metrics of the updated prioritized ML model display better results using AHP in the adaptive incremental learning process.

Figure 5 presents a comparison of the changes in the three most prioritized recall values during incremental learning.

Figure 5b shows that AHP-based adaptive learning provides more consistent performance characteristics throughout the incremental learning process. One can also observe that AHP-based multi-criteria decisions are not the same as simple linear composite scores. The steps 15–20 clearly show that the AHP score not only considers the three "most prioritized" criteria shown in the figures but also all other values according to the judgment matrix used.

5. Discussion

Previous research in the field of classification techniques indicates that the efficacy of these methods is significantly influenced by the characteristics of the dataset [37].

A prevalent issue impacting raw data is the class imbalance problem, which denotes an unequal distribution of values in the response variable. In contrast, there is an additional problem that arises when a machine learning model focuses on a subset of classes rather than all the potential classes. This issue arises when the model is only capable of classifying specific classes. This occurs when using binary classification rather than classifying prioritized classes. The problems mention exist in various fields, including CTI, Fintech fraud detection, IDS, and various other domains where negatively labeled instances substantially exceed favorably labeled instances.

One of the primary goals of ML algorithms is to achieve the highest possible level of accuracy over the whole dataset. This results in the majority class samples receiving a greater amount of attention. Because of this, the learning model does not generate accurate predictions for the minority class samples [38]. To resolve imbalance issues, data-driven and algorithm-driven methods are frequently used.

Data-driven methods make adjustments to the class ratio in the input dataset to produce a balanced distribution of classes. This strategy frequently makes use of sampling methods, such as undersampling, oversampling, or a mix of the two aforementioned methods.

Using algorithm-driven methods, the classification algorithm is modified to make the learning process more manageable, particularly with regard to the minority class. This method does not include any modifications to the distribution of the data that is being input. This strategy incorporates various other approaches, such as thresholding, cost-sensitive learning, and hybrid methods such as ensemble learners [38].

The experimental results were compared to state-of-the-art (SOTA) research, as presented in Table 5.

Table 5. Experimental results compared to state-of-the-art research.

Research	Dataset	Proposed Approach	Dataset Balance Method	Feature Ranking	Improve Performance for Prioritized Classes
Wang et al. [39]	TON_IoT	Transformer-based IoT intrusion detection method	Label smoothing regularization to add fuzzy noise to the training sample labels	The deep global feature extraction capability of a stacked encoder	No
Eren et al. [40]	TON_IoT	RF-based intrusion detection	SMOTE and cluster-based undersampling	Feature elimination using multi-collinearity-based generation of complex features	No
Alotaibi et al. [41]	TON_IoT	Merging five multi-class supervised ML models: RF, DT, ET, XgBoost and K-Nearest Neighbor	SMOTE	Mutual Information, Pearson Correlation Coefficient, and K-Best	No
Proposed	TON_IoT	Incremental learning, AHP ECOC	Original dataset was used	The original dataset was used	Yes

The conclusion that can be drawn from Table 5 is that there are a number of alternative SOTA algorithms that can be utilized to classify unbalanced multi-class data. The proposed approach differs from the others in the following ways:

- The proposed approach employs incremental learning, AHP, and ECOC to improve ML model performance for the important classes that have been prioritized.
- Unlike data-driven methods, the proposed approach does not involve any modifications to the dataset; rather, we use the original multi-class dataset in its current form.

6. Conclusions

When using multi-class classification, it is essential to prioritize classes according to the relevance of each class. Less significant classes should be classified in their current state, whereas classes that are more important should be classified with greater precision. By refusing to examine non-significant classes, we are able to focus on prioritized, highly important classes.

In this research, we propose a method to improve performance regarding the important classes that have been prioritized when utilizing a multi-class dataset in the development of a suitable ML model. We propose an adaptive incremental learning approach to address multi-class data classification challenges, utilizing a completely original multi-class dataset. We propose using the score function $scr(x)$ for the comparison of individual ML performance metrics (criteria) and the construction of the judgment matrices. An adaptive method based on an AHP judgment matrix correctly classifies the key classes prioritized in building an ECOC code matrix.

The important contributions of this research include adaptive incremental learning, which allows ML models to learn new information continuously by analyzing changing data from a data stream, and the AHP, which helps users make the right decision based on model performance evaluation according to prioritized classes. Using AHP-based adaptive incremental learning improved the recall metric results for the most important classes in the suggested method. Recall value of the “injection” class increased from 59.5% to 61.8%, the recall of the “password” class increased from 86.7% to 88.6%, and recall of the “ransomware” class increased from 0 to 23.6%.

Concerns have been raised about the interpretability of complicated machine learning models, which can often make it challenging for businesses to comprehend and explain the process by which security choices are made. In light of this, we will consider the possibility of developing an explainable framework for applications related to CTI and Fintech in the future.

Author Contributions: Conceptualization, A.V., J.T. and N.M.; methodology, A.V., J.T. and N.M.; software, N.M.; validation, J.T. and N.M.; formal analysis and investigation, A.V., J.T. and N.M.; data curation, N.M.; writing—original draft preparation, A.V., J.T. and N.M.; writing—review and editing, A.V., J.T. and N.M.; visualization, J.T. and N.M.; supervision, A.V. All authors have read and agreed to the published version of the manuscript.

Funding: This work was conducted as part of the execution of the project “Mission-driven Implementation of Science and Innovation Programs” (No. 02-002-P-0001), funded by the Economic Revitalization and Resilience Enhancement Plan “New Generation Lithuania”.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created.

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

Abbreviations

The following abbreviations are used in this manuscript:

AdaGrad	Adaptive Gradient Algorithm
AHP	Analytical Hierarchy Process
AI	Artificial Intelligence
CNN	Convolutional Neural Network
CTI	Cyber Threat Intelligence
ECOC	Error-Correcting Output Codes
Fintech	Financial Technology
IDS	Intrusion Detection Systems
KNN	K-Nearest Neighbor Algorithm
LightGBM	Light Gradient-Boosting Machine
LSTM	Long Short-Term Memory
ML	Machine learning
MLP	Multilayer Perceptrons
RF	Random Forest
RMSprop	Root Mean Square Propagation
SGD	Stochastic Gradient Descent
SMOTE	Synthetic Minority Oversampling Technique
SVM	Support Vector Machine
XGBoost	eXtreme Gradient Boosting

References

1. Jafri, J.A.; Amin, S.I.M.; Rahman, A.A.; Nor, S.M. A systematic literature review of the role of trust and security on Fintech adoption in banking. *Heliyon* **2024**, *10*, 1. [CrossRef] [PubMed]
2. Ndichu, S.; Ban, T.; Takahashi, T.; Inoue, D. AI-Assisted Security Alert Data Analysis with Imbalanced Learning Methods. *Appl. Sci.* **2023**, *13*, 1977. [CrossRef]

3. Javaheri, D.; Fahmideh, M.; Chizari, H.; Lalbakhsh, P.; Hur, J. Cybersecurity threats in FinTech: A systematic review. *Expert Syst. Appl.* **2024**, *241*, 122697. [CrossRef]
4. Chao, X.; Ran, Q.; Chen, J.; Li, T.; Qian, Q.; Ergu, D. Regulatory technology (Reg-Tech) in financial stability supervision: Taxonomy, key methods, applications and future directions. *Int. Rev. Financ. Anal.* **2022**, *80*, 102023. [CrossRef]
5. Aaron, W.C.; Irekponor, O.; Aleke, N.T.; Yeboah, L.; Joseph, J.E. Machine learning techniques for enhancing security in financial technology systems. *Int. J. Sci. Res. Arch.* **2025**, *15*, 2. [CrossRef]
6. Angela, O.; Atoyebi, I.; Soyele, A.; Ogunwobi, E. Enhancing fraud detection and prevention in fintech: Big data and machine learning approaches. *World J. Adv. Res. Rev.* **2024**, *24*, 2. [CrossRef]
7. Cao, Z.; Zhao, Z.; Shang, W.; Ai, S.; Shen, S. Using the ToN-IoT dataset to develop a new intrusion detection system for industrial IoT devices. *Multimed. Tools Appl.* **2025**, *84*, 16425–16453. [CrossRef]
8. Ding, H.; Chen, L.; Dong, L.; Fu, Z.; Cui, X. Imbalanced data classification: A KNN and generative adversarial networks-based hybrid approach for intrusion detection. *Future Gener. Comput. Syst.* **2022**, *131*, 240–254. [CrossRef]
9. Li, J.; Othman, M.S.; Chen, H.; Yusuf, L.M. Cybersecurity Insights: Analyzing IoT Data Through Statistical and Visualization Techniques. In Proceedings of the 2024 International Symposium on Parallel Computing and Distributed Systems (PCDS), Singapore, 21–22 September 2024; pp. 1–10. [CrossRef]
10. Yildirim, O.; Bakhshi, S.; Can, F. Prioritized Binary Transformation Method for Efficient Multi-label Classification of Data Streams with Many Labels. In Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM'24). Association for Computing Machinery, New York, NY, USA, 21–25 October 2024; pp. 4218–4222. [CrossRef]
11. Bulavas, V.; Marcinkevičius, V.; Rumiński, J. Study of Multi-Class Classification Algorithms' Performance on Highly Imbalanced Network Intrusion Datasets. *Informatica* **2021**, *3*, 441–475. [CrossRef]
12. Mortier, T.; Wydmuch, M.; Dembczyński, K.; Hüllermeier, E.; Waegeman, W. Efficient set-valued prediction in multi-class classification. *Data Min. Knowl. Disc.* **2021**, *35*, 1435–1469. [CrossRef]
13. Du, J.; Zhou, Y.; Liu, P.; Vong, C.; Wang, T. Parameter-Free Loss for Class-Imbalanced Deep Learning in Image Classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 3234–3240. [CrossRef] [PubMed]
14. Xie, S.; He, Z.; Pan, L.; Liu, K.; Su, S. An adaptive error-correcting output codes algorithm based on gene expression programming and similarity measurement matrix. *Pattern. Recognit.* **2024**, *145*, 109957. [CrossRef]
15. Ruchay, A.; Feldman, E.; Cherbadzhi, D.; Sokolov, A. The Imbalanced Classification of Fraudulent Bank Transactions Using Machine Learning. *Mathematics* **2023**, *11*, 2862. [CrossRef]
16. Chen, W.; Yang, K.; Yu, Z.; Shi, Y.; Chen, C.L.P. A survey on imbalanced learning: Latest research, applications and future directions. *Artif. Intell.* **2024**, *57*, 137. [CrossRef]
17. Khan, A.A.; Chaudhari, O.; Chandra, R. A review of ensemble learning and data augmentation models for class imbalanced problems: Combination, implementation and evaluation. *Expert Syst. Appl.* **2024**, *244*, 122778. [CrossRef]
18. Taylor, G.; Johnson, D.; Roy, K. Threat Detection Using MLP for IoT Network. In Proceedings of the 2024 Internet Computing and IoT and Embedded Systems, Cyber-physical Systems, and Applications (CSCE 2024), Las Vegas, USA, 22–25 July 2024; Springer: Berlin/Heidelberg, Germany; Volume 2260, pp. 108–115. [CrossRef]
19. Maseno, E.M.; Wang, Z.; Sun, Y. Performance Evaluation of Intrusion Detection Systems on the TON_IoT Datasets Using a Feature Selection Method. In Proceedings of the 8th International Conference on Computer Science and Artificial Intelligence (CSAI'24). Association for Computing Machinery, New York, NY, USA, 6–8 December 2024; pp. 607–613. [CrossRef]
20. Bhuiyan, M.H.; Alam, K.; Shahin, K.I.; Farid, D.M. A Deep Learning Approach for Network Intrusion Classification. In Proceedings of the IEEE Region 10 Symposium (TENSYMP), New Delhi, India, 27–29 September 2024. [CrossRef]
21. Khan, I.A.; Moustafa, N.; Pi, D.; Sallam, K.M.; Zomaya, A.Y.; Li, B. A New Explainable Deep Learning Framework for Cyber Threat Discovery in Industrial IoT Networks. *IEEE Internet Things* **2022**, *9*, 11604–11613. [CrossRef]
22. Kumar, P.J.; Neduncheliyan, S.; Adnan, M.M.; Sudhakar, K.; Sudhakar, A.V.V. Anomaly-Based Intrusion Detection System Using Bidirectional Long Short-Term Memory for Internet of Things. In Proceedings of the Third International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), Ballari, India, 26–27 April 2024; pp. 1–4. [CrossRef]
23. Park, C.; Lee, J.; Kim, Y.; Park, J. -G.; Kim, H.; Hong, D. An Enhanced AI-Based Network Intrusion Detection System Using Generative Adversarial Networks. *IEEE Internet Things* **2023**, *10*, 2330–2345. [CrossRef]
24. Oliveira, N.; Praça, I.; Maia, E.; Sousa, O. Intelligent Cyber Attack Detection and Classification for Network-Based Intrusion Detection Systems. *Appl. Sci.* **2021**, *11*, 1674. [CrossRef]
25. Kollu, V.N.; Janarthanan, V.; Karupusamy, M.; Ramachandran, M. Cloud-Based Smart Contract Analysis in FinTech Using IoT-Integrated Federated Learning in Intrusion Detection. *Data* **2023**, *8*, 83. [CrossRef]
26. Alamsyah, A.; Hafidh, A.A.; Mulya, A.D. Innovative Credit Risk Assessment: Leveraging Social Media Data for Inclusive Credit Scoring in Indonesia's Fintech Sector. *Risk Financ. Manag.* **2025**, *18*, 74. [CrossRef]
27. Venčkauskas, A.; Toldinas, J.; Morkevičius, N.; Serkovas, E.; Krištaponis, M. Enhancing the Resilience of a Federated Learning Global Model Using Client Model Benchmark Validation. *Electronics* **2025**, *14*, 1215. [CrossRef]

28. Incremental Learning: Adaptive and Real-Time Machine Learning. Available online: https://blogs.mathworks.com/deep-learning/2024/03/04/incremental-learning-adaptive-and-real-time-machine-learning/ (accessed on 22 April 2025).
29. Petrillo, A.; Salomon, V.A.P.; Tramarico, C.L. State-of-the-Art Review on the Analytic Hierarchy Process with Benefits, Op-portunities, Costs, and Risks. *J. Risk Financ. Manag.* **2023**, *16*, 372. [CrossRef]
30. Saaty, T.L.; Vargas, L.G. The Seven Pillars of the Analytic Hierarchy Process. In *Models, Methods, Concepts & Applications of the Analytic Hierarchy Process*; Springer: Boston, MA, USA, 2001; pp. 27–46. ISBN 978-1-4615-1665-1.
31. Zhang, Y.; Jiang, R.; Wang, L. A Novel Network Attack Evaluation Scheme Based on AHP Grey Clustering Model. In Proceedings of the 8th International Conference on Computing, Control and Industrial Engineering (CCIE2024); Shmaliy, Y.S., Ed.; Springer Nature Singapore: Singapore, 2024; pp. 375–383.
32. Akshitha, K.; M, B.A.; Kodipalli, A.; Rao, T.; R, R.B.; N, G. An Approach for Ranking Cyber Crime Using Fuzzy AHP and Fuzzy TOPSIS. In Proceedings of the 2023 International Conference on Recent Advances in Science and Engineering Technology (ICRASET), Dharwad, India, 11 August 2023; pp. 1–7.
33. Khaira, A.; Dwivedi, R.K. A State of the Art Review of Analytical Hierarchy Process. *Mater. Today Proc.* **2018**, *5*, 4029–4035. [CrossRef]
34. Moustafa, N. A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets. *Sustain. Cities Soc.* **2021**, *72*, 102994. [CrossRef]
35. TON_IoT Datasets. Available online: https://unsw-my.sharepoint.com/personal/z5025758_ad_unsw_edu_au/_layouts/15/onedrive.aspx?id=/personal/z5025758_ad_unsw_edu_au/Documents/TON_IoT%20datasets&ga=1 (accessed on 22 April 2025).
36. Majidian, Z.; TaghipourEivazi, S.; Arasteh, B.; Babaie, S. An intrusion detection method to detect denial of service attacks using error-correcting output codes and adaptive neuro-fuzzy inference. *Comput. Electr. Eng.* **2023**, *106*, 108600. [CrossRef]
37. Oreski, D.; Oreski, S.; Klicek, B. Effects of dataset characteristics on the performance of feature selection techniques. *Appl. Soft Comput.* **2017**, *52*, 109–119. [CrossRef]
38. Thabtah, F.; Hammoud, S.; Kamalov, F.; Gonsalves, A. Data imbalance in classification: Experimental evaluation. *Inf. Sci.* **2020**, *513*, 429–441. [CrossRef]
39. Wang, P.; Wang, X.; Song, Y.; Huang, J.; Ding, P.; Yang, Z. TransIDS: A Transformer-based approach for intrusion detection in Internet of Things using Label Smoothing. In Proceedings of the 4th International Conference on Computer Engineering and Application (ICCEA), Hangzhou, China, 7–9 April 2023; pp. 216–222. [CrossRef]
40. Eren, K.K.; Küçük, K. Improving Intrusion Detection Systems for IoT Devices using Automated Feature Generation based on ToN_IoT dataset. In Proceedings of the 8th International Conference on Computer Science and Engineering (UBMK), Burdur, Turkiye, 13–15 September 2023; pp. 276–281. [CrossRef]
41. Alotaibi, Y.; Ilyas, M. Enhancing IoT Attack Detection Through Ensemble-Based Multiclass Attacks Classification. In Proceedings of the 20th International Conference on Smart Communities: Improving Quality of Life using AI, Robotics and IoT (HONET), Boca Raton, FL, USA, 4–6 December 2023; pp. 1–6. [CrossRef]

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.