Streamlined and Resource-Efficient Estimation of Epistemic Uncertainty in Deep Ensemble Classification Decision via Regression

Jordan F. Masakuna, D'Jeff K. Nkashama, Arian Soltani, Marc Frappier, Pierre M. Tardif and Froduald Kabanza
GRIC, University of Sherbrooke, Sherbrooke, QC, Canada

***Abstract*—Ensemble deep learning (EDL) has emerged as a leading tool for epistemic uncertainty quantification (UQ) in predictive modelling. Our study focuses on the utilization of EDL, composed of auto-encoders (AEs) for out-of-distribution (OoD) detection. EDL offers straightforward interpretability and valuable practical insights. Conventionally, employing multiple AEs in an ensemble requires regular training for each model whenever substantial changes occur in the data, a process that can become computationally expensive, especially when dealing with large ensembles. To address this computational challenge, we introduce an innovative strategy that treats ensemble UQ as a regression problem. During initial training, once the uncertainty distribution is established, we map this distribution to one ensemble member. This approach ensures that during subsequent trainings and inferences, only one ensemble member and the regression model are needed to predict uncertainties, eliminating the need to maintain the entire ensemble. This streamlined approach is particularly advantageous for systems with limited computational resources or situations that demand rapid decision-making, such as alert management in cybersecurity. Our evaluations on five benchmark OoD detection data sets demonstrate that the uncertainty estimates obtained with our proposed method can, in most cases, align with the uncertainty distribution learned by the ensemble, all while significantly reducing the computational resource requirements.**

***Index Terms*—Uncertainty quantification; Ensemble of auto-encoders; Out of distribution detection; Unsupervised learning.**

I. Introduction

Auto-encoders (AEs), an unsupervised deep learning (DL) model, have emerged as highly effective tools in a wide range of classification tasks, with a notable strength in detecting out-of-distribution (OoD) data points. OoD detection is related to anomaly detection [1]—while both OoD detection and anomaly detection involve identifying deviations from expected patterns within the data set, OoD detection focuses on recognizing instances that lie outside data set even if they are not necessarily anomalous. Identifying OoD instances often involves further investigation to discern whether they represent anomalies or new normal patterns. Given the challenge of limited labelled data in OoD systems, supervised learning is often impractical. AEs are gaining popularity in fields such as anomaly detection [1], computer vision [2], and image search [3]. AEs accomplish the task of identifying OoD instances by initially learning from expected behaviour. However, the performance of OoD detection methods can vary significantly depending on the specific application context [4]. An important capability expected from AEs is the ability to gauge their own predictive confidence accurately, especially that AEs are reported to suffer from overconfidence issues [5]. This presents a notable challenge, as assessing the confidence of predictive models is inherently difficult due to the typical unavailability of ground truth uncertainty distributions. Here, a prediction is both a classification decision (as is an instance anomalous?) and an assessment of uncertainty (as is the decision doubtful?).

A crucial aspect of a DL model's reliability is its ability to assess and communicate its level of confidence in its decisions, which significantly contributes to transparency and trustworthiness [6]. Building trustworthy DL models entails considering various factors, including the presence of uncertainty in classification decisions. Numerous applications, such as cybersecurity, banking, and nuclear systems, demand the use of reliable DL models. For example, security analysts often confront a high volume of alerts while having limited processing resources at their disposal [7]. In such cases, employing an uncertainty measure can aid in prioritizing alerts, ensuring that the most certain ones (i.e., classification decisions associated with low uncertainties) are addressed first. This same principle applies to credit approval processes in banking systems [8] and sensor failure detection in nuclear systems [9]. Therefore, uncertainty quantification (UQ) plays a pivotal role in allowing a DL model to acknowledge classification decisions that carry a substantial degree of uncertainty. Ignoring predictive uncertainty can result in erroneous classification decisions, potentially leading to severe consequences [10].

Various UQ techniques, such as Monte Carlo dropout [11], Variational AEs (VAEs) [12] and Ensemble Deep Learning (EDL) [13], have been proposed in the literature [14]. UQ techniques often yield distinct uncertainty distributions due to their differing modelling assumptions (explained by, for example, the no free lunch theorem [15]). Our study focuses on EDL as it offers straightforward interpretability and valuable practical insights. EDL entails the training of several DL models with each model making predictions independently before these predictions are aggregated to form a final decision. Here, AEs are DL models considered for composing our EDL, i.e., EDL are used for both OoD detection and UQ (while EDL has the potential to enhance classification accuracy, its primary function in this paper is UQ). Variations in AEs' decisions are considered as uncertainties. We utilize standard deviation [16] and entropy [17] as our choice of

Corresponding author: jordan.felicien.masakuna@usherbrooke.ca metrics for assessing uncertainty. These metrics are widely accepted in the UQ literature. Standard deviation is used when one needs to measure the dispersion of predicted continuous values and entropy is used when one needs to quantify the randomness of a probability distribution. It is important to note that the choice between standard deviation and entropy should be guided by the specific characteristics and context of the underlying application, whose consideration is out of scope here. Also, there exist two main types of uncertainties: aleatoric and epistemic [18]. Our study addresses epistemic uncertainty only, which arises from different AEs' parameters initialization as required by EDL.

An AE model that is uncertain about a classification decision is more likely to mislabel a sample. It should be noted that classification and UQ are distinct tasks here. During prediction, each AE-based model, independently trained, produces a reconstruction error (an AE operates by attempting to reconstruct an input x in minimizing the reconstruction error as much as possible). Then a class is determined by comparing the reconstruction error to a classification threshold, to indicate whether the underlying data point is anomalous or not (0 for normality and 1 for abnormality). The combined classification decisions from all individual predicted classes, using majority vote [19], are considered to classify the underlying data point, and individual reconstruction errors are considered to calculate the uncertainty associated with the classification decision.

To illustrate EDL for UQ using the two uncertainty metrics, let's consider the scenario where 5 reconstruction errors (refer to (1)) are generated from 5 AEs on a data point x with $e_x^{(k)}$ the kth element of vector e_x:

$$e_x = \begin{pmatrix} 1 \\ 1.2 \\ 0.4 \\ 1.3 \\ 0.7 \end{pmatrix}, \tag{1}$$

with the classification threshold set, for example, to 0.8 for all the 5 models (but each model could have its own threshold). The majority of members (3 out of 5) exhibit reconstruction errors surpassing the threshold, implying the anomalous nature of the underlying data point. So, the final class of x is $y_x = 1$. We need a probability distribution, p_x, representing reconstruction errors to quantify standard deviation and entropy. Let $p_x^{(k)}$ denote the kth element of p_x. The probability distribution associated with e_x (by dividing each reconstruction error by the sum of all values) is

$$p_x = \begin{pmatrix} 0.217 \\ 0.261 \\ 0.087 \\ 0.283 \\ 0.152 \end{pmatrix}. \tag{2}$$

Epistemic uncertainty from these 5 reconstruction errors using standard deviation and entropy are given by

$$\text{standard deviation} = \sqrt{\sum_{k=1}^{5} p_x^{(k)} \left(e_x^{(k)} - \bar{e}_x\right)^2} \approx 0.331\,,$$

and

$$\text{entropy} = -\sum_{k=1}^{5} p_x^{(k)} \log_2 p_x^{(k)} \approx 1.538\,,$$

where

$$\bar{e}_x = \frac{1}{5}\sum_{k=1}^{5} e_x^{(k)} \approx 0.920\,,$$

Depending on pre-defined uncertainty thresholds for standard deviation (τ_{std}) and entropy (τ_{ent}), the predicted class ($y = 1$) assigned to x could be rejected if its associated uncertainty (0.331 for standard deviation or 1.538 for entropy in this case) is found to exceed their respective uncertainty threshold (setting an uncertainty threshold may necessitate domain knowledge and experimental investigation to ensure that decisions regarding misclassified instances are rejected). For example if $\tau_{\text{std}} = 0.2$ then the classification decision will be rejected. So, only accepted classification decisions will be treated or prioritized for downstream tasks. It is important to note that a classification decision is rejected not necessarily because it is incorrect (as it could be the correct decision) but rather because it is deemed doubtful.

EDL, which offers the potential to create independent models, is recommended as it can capture various low-loss regions, each potentially corresponding to desired outcomes. The diversity provided by independent models is crucial for ensuring robust UQ. EDL stands out for its capability to enhance performance by orchestrating diverse models into a cohesive framework. However, implementing EDL can be computationally expensive since it can comprise as many as a hundred DL models [20]. As a result, it necessitates the maintenance of multiple DL models to ensure they remain effective over time. Maintenance entails retraining models in response to substantial changes occurring within the data set. This is essential because an ensemble UQ approach relies on the collective input from all its members. The primary objective of this study is to explore a more efficient approach to quantifying uncertainties generated by EDL.

Contributions. We introduce a novel method for efficiently estimating epistemic uncertainty in an ensemble of AEs within the EDL framework. Traditional methods rely on maintaining and evaluating multiple members of an ensemble to estimate uncertainty, which can be computationally expensive and resource-intensive. Our approach diverges from this conventional methodology in several significant ways:

- **Novel strategy for uncertainty estimation**. We introduce a groundbreaking strategy where the uncertainty distribution learned from an ensemble during the initial training phase is used as a ground truth reference. This approach eliminates the need to retain multiple ensemble members for subsequent uncertainty estimations, thus simplifying the process.

- **Efficient mapping through regression**. We develop a regression model that maps the latent space of a single ensemble member to the uncertainty distribution of the ensemble. This innovation allows us to use only one member of the ensemble and the regression model during later training and inference phases, substantially reducing computational overhead.
- **Resource efficiency and practicality**. Our method drastically reduces the computational resources required, making it particularly beneficial for systems with limited resources or scenarios demanding quick decision-making. This resource efficiency is achieved without compromising the quality of the uncertainty estimates.
- **Empirical validation**. We validate our approach through experiments on five OoD detection datasets. Our results demonstrate that the uncertainty estimates obtained with our method closely match the distribution learned by the full ensemble, while significantly lowering computational costs. This empirical evidence underscores the effectiveness and practicality of our approach.

Our inspiration for this approach stems from the work presented in [21] (called DUAD), which addresses data contamination issues in the context of anomaly detection where authors employ latent representations of the original data in conjunction with reconstruction errors to effectively mitigate contamination concerns. The key difference between our work and DUAD is that we quantify uncertainty associated with a classification decision which can be obtained from DUAD.

The remainder of the paper is organized as follows: Section II provides background on UQ and related work to our proposal. Our proposed method is discussed in Section III. Section IV presents and discusses the results. Finally, in Section V, we conclude and suggest areas for further research.

II. Background and Related Work

A. Background

Uncertainty. In the realm of DL models, we encounter two types of uncertainties: aleatoric and epistemic [18]. Aleatoric uncertainty is a consequence of the inherent randomness in the data. Capturing this uncertainty involves understanding the conditional distribution of a target given specific input values, essentially representing model uncertainty. On the other hand, epistemic uncertainty arises from a lack of knowledge about the true parameters of the model. To capture this form of uncertainty, we delve into learning about the regions of the input space that remain unexplored by the data, signifying data uncertainty [22]. Our study addresses epistemic uncertainty only, which arises from different models' parameters initialization, used in the context of OoD systems. It is important to note that UQ techniques often yield distinct uncertainty profiles due to their differing modelling assumptions.

Out-of-distribution (OoD) detection involves the identification of data points that deviate from expected behaviour. A wide range of OoD detection methods has been proposed in the literature, spanning classical approaches to DL techniques. Within the realm of DL, various strategies have emerged to tackle this challenge. These strategies encompass supervised methods, although their adoption is limited due to the need for labelled data—given that most OoD data sets lack labels. More commonly, the approaches involve unsupervised techniques, such as AEs [23], or self-supervised methods, exemplified by natural language processing applications like DeepLog [24], [25]. Furthermore, the field of OoD detection addresses several pertinent issues, including the challenge of data contamination. These issues shape the ongoing trends within the domain of OoD systems. This study treats AEs for OoD detection.

Auto-encoder (AE): Let X denote an OoD detection data set given by

$$X = X^{-} \cup X^{+} = \{x_i\}_{i=1}^{N} \, , \tag{3}$$

where X^{-} and X^{+} are the sets of in-distribution and OoD data points respectively. Every x_i in the training data set has its label $y_i \in \{0, 1\}$ ($y_i = 0$ for $x_i \in X^{-}$ and $y_i = 1$ otherwise). A classical AE architecture is illustrated in Figure 1. An AE comprises two neural networks: an encoder, denoted as f_θ, and a decoder, denoted as g_ϕ. The entire AE model is represented as h_Θ. The encoder's primary role is to learn a latent space representation from the input data. Once this latent space is learned, the decoder aims to reconstruct the input instance x_i using its corresponding latent representation z_i [26]. The three functions are defined as follows,

$$\begin{aligned} f_\theta &: X^{-} \to Z \\ g_\phi &: Z \to X^{-} \\ h_\Theta &: X^{-} \to X^{-} \, , \end{aligned} \tag{4}$$

where Z denotes a latent space.

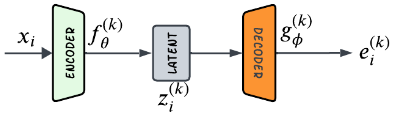


Figure 1: Classical AE architecture

Our setting is as follows. An AE model is trained with normal data points X^{-} only. So, the model attempts to minimize a loss function (which represents the sum of reconstruction error e_i on each data point x_i versus its reconstructed copy $\hat{x}_i$) that, using an ℓ_2-norm (other metrics could be used), is

$$e_i = \|x_i - \hat{x}_i\| \, . \tag{5}$$

The objective function of the entire network h_Θ is given by

$$\min_{\Theta} \sum_{x_i \in X^{-}} e_i \, , \tag{6}$$

where $\Theta = (\theta, \phi)$ denote model's parameter. From what precedes, it is evident that $z_i = f_\theta(x_i)$ and $\hat{x}_i = g_\phi(z_i)$.

Let $\Delta = \Delta^{-} \cup \Delta^{+}$ be the set of reconstruction errors where $\Delta^{-} = \{e_i : x_i \in X^{-}\}$ and $\Delta^{+} = \{e_i : x_i \in X^{+}\}$.

It is important to note that in the context of ensemble UQ approach, some notational changes are required to account for each model in the ensemble. Specifically, we consider a total of K models in the ensemble, which introduces variations in our previous notations. The notations f_θ will be replaced by $f_\theta^{(k)}$ to indicate the encoder of the kth AE model, g_ϕ by $g_\phi^{(k)}$, h_Θ by $h_\Theta^{(k)}$, z_i by $z_i^{(k)}$, $\hat{x}_i$ by $\hat{x}_i^{(k)}$, e_i by $e_i^{(k)}$ and $\hat{y}_i$ by $\hat{y}_i^{(k)}$.

Equation (4) emphasizes that an AE should be trained exclusively on normal data points to effectively identify OoD instances that deviate from the learned normal patterns. This approach holds merit for two main reasons. Firstly, characterizing the distributions of abnormal data points can be inherently challenging due to their diverse and often unpredictable nature. Secondly, OoD data sets are typically unlabelled, making it difficult to employ supervised classification methods.

AEs excel in OoD detection tasks [21], especially when labelled samples are limited. They learn to detect anomalies solely from the normal data distribution, eliminating the need for labelled examples. EDL techniques bolster detection accuracy by combining multiple AE models, each trained on distinct parameter aspects. This collective approach comprehensively captures normal data patterns and effectively identifies deviations indicative of OoD samples. In critical domains like cybersecurity, where OoD detection is paramount, not just classification decision but also the uncertainty behind model classification decision is crucial. Knowing the level of uncertainty in security alerts enables a prioritization mechanism.

Ensemble deep learning (EDL), as elucidated in [27], revolves around a systematic amalgamation of multiple models to predict the class of a data point. Ensemble learning is a broad concept used in the machine learning community, it is considered EDL when individual models are all DL models. EDL harnesses the collective power of several individual models to enhance generalization performance. Given that each DL model can possess millions of parameters and an ensemble can encompass up to hundreds of independent models, they can potentially approximate the unknown function in numerous ways. It is assumed that there exist multiple low-loss regions where several models can yield desired outcomes while employing different underlying functions. EDL strategically explores these diverse low-loss regions, culminating in a distribution of functions with varying characteristics [28].

Model training in ensemble learning can be conducted either collaboratively or independently [27]. In a collaborative approach (e.g. boosting [29]), individual models operate in concert, exchanging information to enhance overall performance. Conversely, in independent ensemble learning, individual models are trained separately (e.g. bagging [30]) and the final classification decision relies on fusion strategies as its cornerstone. Fusion strategies encompass techniques such as the sum rule, majority rule, and the Borda count [31]. In both cases, individual models may be homogeneous (utilizing the same model) or heterogeneous otherwise. Independent ensemble learning inherently introduces variability by combining predictions from multiple models trained with different subsets of data or with different parameter initialization. Other techniques (e.g. Bayesian neural networks [22] and Variational AEs [12]) treat model parameters/architecture as random variables and place prior distributions over them, and learn their posterior distribution. We consider independent EDL using homogeneous models (all our individual models are AE) trained on the same data but different parameter initialization. However, its notable drawback lies in the necessity to maintain all individual models to adapt to changes for effective utilization, a primary concern in this paper. To address this concern, we treat EDL as a regression task.

Regression learning serves as a fundamental method for comprehending the intricate relationship between features and a target. Once the relationship between features and target has been estimated, regression models enable us to predict outcomes. DL regression models find wide application in predictive analytics, ranging from forecasting trends to predicting outcomes in various domains [32]. For instance, Pang et al. proposed a DL regression model for anomaly detection in video data [33]. Other applications encompass forecasting, capital asset pricing, and competitive business analysis. While the literature presents a multitude of deep regression models, each exhibiting distinct performance characteristics under varying circumstances, our paper does not focus on constructing or extensively studying advanced regression models. Instead, our primary objective is to employ a straightforward regression model to address our research question. In this regard, we utilize multilayer perceptron (MLP) [34], although alternative regression algorithms (e.g. random forest) could be explored [35]. We use MLP as a typical regression model for the following reasons [36]:

- it is a powerful DL model capable of capturing complex nonlinear relationships in data. It excels at modeling intricate patterns and extracting high-level features from unstructured data.
- it is highly flexible and adaptable to different types of data and problem domains. It can handle large-scale data sets with high dimensionality and is robust to various data distributions and structures.
- it learns hierarchical representations of data through layers of neurons, enabling it to automatically extract features and learn abstract representations from raw data.

B. Related work

Our proposed method builds on EDL [27] for UQ. With EDL, every member is retrained whenever there are substantial changes in the data set. Also, each member participates in the decision-making process. However, the retraining of all members and their involvement in decision can be time-consuming. Unlike EDL which maintains multiple AE members, our approach leverages the uncertainty distribution derived from the ensemble during the initial training phase as a ground truth reference. We then develop a regression model that maps the latent space of a selected ensemble member to the ensemble's uncertainty distribution for subsequent trainings and inferences. This method uses only one member of the ensemble and the regression model in later trainings, regardless of the initial ensemble size. During future trainings, model parameters are initiliazed by their previous optimal values. This concept is related to knowledge distillation where a smaller model (the "student“) is trained to mimic the behavior of a larger model (the "teacher“) for complexity efficiency [37].

Our streamlined approach is especially beneficial for systems with limited computational resources or in situations requiring rapid decision-making. Our experiments show that the uncertainty estimates produced by our method are comparable to those learned by the full ensemble, while significantly reducing the computational resources required. To apply our method, we assume that changes in the original input data do not significantly affect the uncertainty distribution. We employ data augmentation to anticipate future changes [38].

III. Ensemble Uncertainty Inference

Our solution involves the creation of a regression model to estimate the ensemble uncertainty distribution. This approach infers epistemic uncertainties using only one ensemble member and the regression model, reducing the computational resources required significantly. There are three distinct phases involved: *groundwork*, *regression* and *retraining*. In the groundwork phase, we initially train an ensemble of AEs where an uncertainty distribution is generated. In the regression phase, we construct a regression model capable of mapping a latent representation of the original data to the uncertainty distribution obtained during the groundwork phase. This regression model serves as the sole tool for future UQ tasks. The retraining phase is initiated in response to substantial changes within the data set, necessitating the retraining of both the chosen ensemble member and the regression model.

A. Groundwork regime

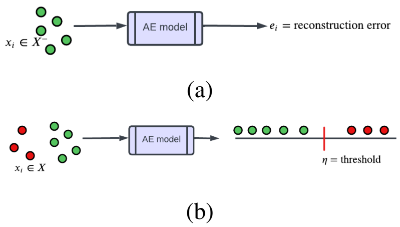
The uncertainty distribution is learned from a dedicated validation data set (we apply data augmentation to expand the distribution of data within the input space [38], a technique that anticipates future changes during current training and promotes effective knowledge distillation from ensemble to regression), denoted as X_v. Given an ensemble of K trained AEs, each AE model $h_\Theta^{(k)}$ produces a reconstruction error $e_i^{(k)}$ (an AE operates by attempting to reconstruct an input x in minimizing the reconstruction error as much as possible). Then a classification decision is determined by comparing the reconstruction error to a classification threshold, to indicate whether the underlying data point $x_i \in X_v$ is anomalous or not (0 for normality and 1 for abnormality). The ensemble classification decision and the uncertainty associated with it are obtained from combining individual classification decisions and individual reconstruction errors respectively.

In the classification process using the kth model $h_\Theta^{(k)}$, a classification threshold $\eta^{(k)}$ is established during the training phase. Ideally, this classification threshold should be determined as in (7). A new data point x_i is flagged as OoD if $e_i^{(k)} > \eta^{(k)}$ (with $e_i^{(k)}$ as defined in (5)). Otherwise, it is normal (see Figure 2). A theoretical threshold is

$$\eta^{(k)} = \max\left\{e_i^{(k)} \ : \forall x_i \in X^-\right\} . \tag{7}$$

The final classification decision is obtained by combining individual classification decisions using majority vote [19].

In practical applications, the approach outlined in Equation (7) may not always yield optimal results. In reality, a validation data set is often employed to determine an optimal threshold using search strategies. AEs can be sensitive to the choice of this threshold, as certain strategies for fixing $\eta^{(k)}$ may perform well on specific data sets but not on others. In fact, other research works have explored dynamic methods for choosing $\eta^{(k)}$ [39] or have adopted Bayesian approaches to learn the distribution of $\eta^{(k)}$ [40].



(a)

(b)

Figure 2: Expected behavior using AEs: (2a) training, (2b) inference. Normal instances are in green and abnormal in red.

The epistemic uncertainty, denoted σ_i, measured using the standard deviation, is computed as follows:

$$\sigma_i = \sqrt{\sum_{k=1}^{K} p_i^{(k)} \left(e_i^{(k)} - \bar{e}_i\right)^2} \ , \ \text{where} \ \ \bar{e}_i = \frac{1}{K}\sum_{k=1}^{K} e_i^{(k)} \ , \tag{8}$$

or using entropy [16], is given by,

$$\sigma_i = -\sum_{k=1}^{K} p_i^{(k)} \log_2 p_i^{(k)} \ , \ \text{where} \ \ p_i^{(k)} = \frac{e_i^{(k)}}{K\bar{e}_i} \ . \tag{9}$$

B. Regression regime

Once the initial training is achieved, we can formulate the original UQ as a regression problem. To treat UQ as a regression problem, we start with the assumption that an AE's latent representation Z can effectively describe the uncertainty distribution obtained from the ensemble. The latent representation fed into the regression model inherently possesses features that are significant for explaining the ensemble uncertainty. There are two steps: construction of the regression model and selection of a member to generate the latent representation.

1) Construction of the regression model: Now, consider a candidate member of the ensemble, $h_\Theta^{(j)}$. Let $Z_j = \{f_\theta^{(j)}(x_i)|x_i \in X_v\}$ and $E_j = \{e_i^{(j)}|x_i \in X_v\}$ denote the latent representation and reconstruction errors of elements of X_v using $h_\Theta^{(j)}$ respectively, and $\Sigma = \{\sigma_i|x_i \in X_v\}$ be the uncertainty distribution that the ensemble method has produced. The goal is to design a regression model, Φ, to learn Σ (Σ is the ground truth here). Inspired by [21], our final feature space for regression is L_j, which is a conjunction of the latent representation Z_j and the reconstruction errors E_j of the jth AE model. Putting reconstruction errors E_j and latent representation Z_j together reinforces informativeness of exogenous features that constitute our latent data set. Our regression function is then defined as

$$\Phi : L_j \rightarrow \Sigma \, . \tag{10}$$

The idea of regression is to minimize the following function

$$\sum_{\sigma_i \in \Sigma} |\sigma_i - \hat{\sigma}_i^{(j)}| \,, \tag{11}$$

where inferred uncertainty $\hat{\sigma}_i^{(j)} = \Phi(l_i^{(j)})$ for $l_i^{(j)} \in L_j$.

The following set of important properties is extracted and derived from the expected regression function, shedding light on various aspects and characteristics of the underlying data modelling process:

- For every $x_i, x_k \in X$, $\exists l_i^{(j)}, l_k^{(j)} \in L_j$ such that
$$\|x_i - x_k\| \propto \|l_i^{(j)} - l_k^{(j)}\| \,. \tag{12}$$
- For every $x_i \in X$, its corresponding feature $l_i^{(j)} \in L_j$ and for some $\epsilon > 0$,
$$|\Phi(l_i^{(j)}) - \sigma_i| < \epsilon \,. \tag{13}$$

Here, the proposed method is essentially applied to estimate uncertainties of a one-parameter distribution. If the aim is to predict at least two targets simultaneously in a regression task, existing DL regression models can achieve it. This is known as multi-target regression [41]. For example, one can make the output of multi-layer network to have several neurons, each representing one of the targets to predict.

2) Ensemble member selection: As our goal is to streamline the computational resources needed to train and maintain EDL for UQ, ultimately, only one ensemble member will be chosen. We assume there is an ensemble member, namely an AE, that provides a good approximation of the feature data set L. There are three selection criteria: the selected member should (1) minimize the standard error between the ensemble's uncertainties and the estimates, (2) maximize the F$_1$-score of classification decisions, and (3) maximize the agreement between acceptances/rejections of classification decisions from the ensemble. We use an agreement metric (AGR) to measure the proportion of alignment between decisions obtained from the ensemble uncertainties and those from the regression model. Let τ denote an uncertainty threshold, and recall that σ_i ($\sigma_i \in \Sigma$) is the ensemble uncertainty and y_i the ensemble classification decision of an input x_i. A classification decision y_i is accepted only if $\sigma_i \leq \tau$; otherwise, it is rejected. The agreement level between the ensemble and the regression model is calculated as follows:

$$\text{AGR}_k = \frac{\#(\Gamma_k^{\checkmark} \cup \Gamma_k^{\times})}{\#\Sigma} \,, \tag{14}$$

where

$$\Gamma_k^{\checkmark} = \left\{ \sigma_i \in \Sigma \,|\, (\sigma_i \leq \tau \wedge \hat{\sigma}_i^{(k)} \leq \tau) \,, \forall k \right\} \,,$$

and

$$\Gamma_k^{\times} = \left\{ \sigma_i \in \Sigma \,|\, (\sigma_i > \tau \wedge \hat{\sigma}_i^{(k)} > \tau) \,, \forall k \right\} \,.$$

Let TP$_k$, FP$_k$ and FN$_k$ denote the rates of true positives, false positives and false negatives of the kth model respectively. The positive class here is the OoD class which is the class of interest. F$_1$-score of an ensemble member is given by

$$\text{F}_1\text{-score}_k = 2 \times \frac{\text{PREC}_k \times \text{REC}_k}{\text{PREC}_k + \text{REC}_k} \,, \tag{15}$$

```
function EDL($K, X^-, X_v$)
    for $k \in \{1, 2, \ldots, K\}$ do
        $h_\Theta^{(k)} \leftarrow$ AE($X^-$, parameters)
        $\eta^{(k)} \leftarrow$ validation($h_\Theta^{(k)}, X_v$)
    end for
    $\Sigma \leftarrow \emptyset$
    for $x_i \in X_v$ do
        for $k \in \{1, 2, \ldots, K\}$ do
            $e_i^{(k)} \leftarrow \|h_\Theta^{(k)}(x_i) - x_i\|$
        end for
        $\sigma_i \leftarrow$uncertainty$\left(e_i^{(1)}, e_i^{(2)}, \ldots, e_i^{(K)}\right)$
        $\Sigma$.append($\sigma_i$)
    end for
    return $\Sigma$
end function
function regression($k, X_v, \Sigma$)
    $(Z_k, E_k) \leftarrow (\emptyset, \emptyset)$
    for $x_i \in X_v$ do
        $z_i^{(k)} \leftarrow f\theta_\theta^{(k)}(x_i)$
        $Z_k$.append($z_i^{(k)}$)
        $\hat{x}_i^{(k)} \leftarrow h_\Theta^{(k)}(x_i)$
        $e_i^{(k)} \leftarrow \|\hat{x}_i^{(k)} - x_i\|$
        $E_k$.append($e_i^{(k)}$)
    end for
    $L_k \leftarrow$ conjunction($Z_k, E_k$)
    return MLP($L_k, \Sigma$, parameters)
end function
procedure main($j, K, \tau, X^-, X_v, X_t$)
    $\Sigma \leftarrow$ EDL($K, X^-, X_v$)
    $\Phi \leftarrow$regression($j, X_v, \Sigma$)
    for $x_i \in X_t$ do
        $z_i^{(j)} \leftarrow f_\theta^{(j)}(x_i)$
        $e_i^{(j)} \leftarrow \|h_\Theta^{(j)}(x_i) - x_i\|$
        $\hat{y}_i^{(j)} \leftarrow$ classification-decision($e_i^{(j)}, \eta^{(j)}$)
        $l_i^{(j)} \leftarrow$ conjunction($z_i^{(j)}, e_i^{(j)}$)
        if $\Phi(l_i^{(j)}) > \tau$ then
            reject $\hat{y}_i^{(j)}$
        end if
    end for
end procedure
```

Figure 3: Regression UQ method where j denotes the id of the selected member, K the ensemble size, τ the threshold, X^-, X_v and X_t training, validation and test data sets respectively.

where

$$\text{PREC}_k = \frac{\text{TP}_k}{\text{TP}_k + \text{FP}_k} \quad \text{and} \quad \text{REC}_k = \frac{\text{TP}_k}{\text{TP}_k + \text{FN}_k}$$

The member of the ensemble, $h_\Theta^{(j)}$, to be selected should satisfy the following criteria:

$$\text{AGR}_j = \max\{\text{AGR}_k \,, \forall k\} \,, \tag{16}$$

$$\text{F}_1\text{-score}_j = \max\{\text{F}_1\text{-score}_k \,, \forall k\} \,, \tag{17}$$

and

$$\sum_{l_i^{(j)}\in L_j}\left|\hat{\sigma}_i^{(j)}-\sigma_i\right| = \min\left\{\sum_{l_i^{(k)}\in L_k}\left|\hat{\sigma}_i^{(k)}-\sigma_i\right|, \forall k\right\}. \qquad (18)$$

From a multi-objective optimization perspective, when these three criteria conflict, it is important to note that criterion (16) takes precedence over (17) which in turn takes precedence over (18). Criterion (16) presents an effective method for measuring inferred uncertainties, as utilized in this work, which involves accepting or rejecting classification decisions. In other words, the regression model should maintain the acceptance/rejection of classification decisions made by the ensemble.

C. Retraining phase

Recall that X_v denotes the original validation data set, L the latent representation of X_v produced by the selected ensemble member and Σ the uncertainty distribution. Whenever substantial changes X_Δ occur within the input data set, the retraining phase involves the following:

1) Generate a latent representation L_Δ of the new data X_Δ using the already trained chosen ensemble member.
2) Predict the uncertainty Σ_Δ associated with L_Δ using the already trained regression model.
3) Retrain the ensemble member on $X_{\text{new}} = X_v \cup X_\Delta$. This process generates L_{new}.
4) Retrain the regression model $\Phi : L_{\text{new}} \rightarrow \Sigma \cup \Sigma_\Delta$.

During the retraining phase, we use the previously determined optimal parameter values for initializing the parameters. A pseudo-code of our approach is given in Figure 3.

Data set	Training	Test	Validation	Attributes after treatment	Anomaly rate
KDD	632307	3412899	853224	122	80%
NSL	50085	78745	19686	122	48%
IDS	73438	70000	292542	78	36%
KITSUNE	200341	100000	302592	114	45%
CICIOT	90000	102000	408000	46	50%

Table I: The characteristics of data sets to train UQ methods.

IV. Experimental Investigation

A. Experimental setup

To evaluate the significance of our method, we will employ the following correlation metrics between the ensemble uncertainty distribution and the inferred uncertainty distribution:

- Pearson's Correlation Coefficient (PCC): measures the linear correlation between the predicted and ground truth uncertainty distributions. A PCC value closer to 1 implies a strong positive linear relationship, signifying a robust model. Its threshold is set to 0.75.
- Spearman's Correlation Coefficient (SCC): assesses the monotonic relationship between the predicted and ground truth distributions. Its interpretation is similar to PCC.

We will use Mean Squared Error (MSE) to quantify the differences between the predicted and ground truth uncertainty distributions. A lower MSE indicates a closer match between the predicted and ensemble uncertainty distributions. We will also use agreement score (AGR as defined in (14)) to measure the proportion of alignment between the decisions obtained from the regression model and the decisions obtained using the ensemble uncertainties. We will also use F$_1$-score to measure the performance of the selected ensemble member. Additionally, we will use scatter, line and box plots to visualize the correlation between the ensemble's and the inferred uncertainty distributions. In constructing the ensemble for UQ, we will consider ensemble sizes of up to 15, though other sizes can be explored based on available computational resources. For the regression task, we expect that a simple regression model would be appropriate for this scenario (we use MLP). The latent representation which inputs the regression model has already in principle interesting features in explaining the target. Although we tried different regression models in our experiments, we only report results from MLP as it is found to be a typical DL regression model and all regression models used gave similar results. Hyperparameters, including the prior weight decay, batch size, and learning rate will be set within specified ranges to ensure comprehensive evaluation. These parameter values will be randomly selected to avoid bias. The weight decay will fall within the range $[10^{-7}, 10^{-5}]$, batch sizes of $16, 32$, or 64 will be considered and learning rates will be selected from $[10^{-10}, 10^{-6}]$ for all AE models. These parameter settings will allow us to conduct a rigorous

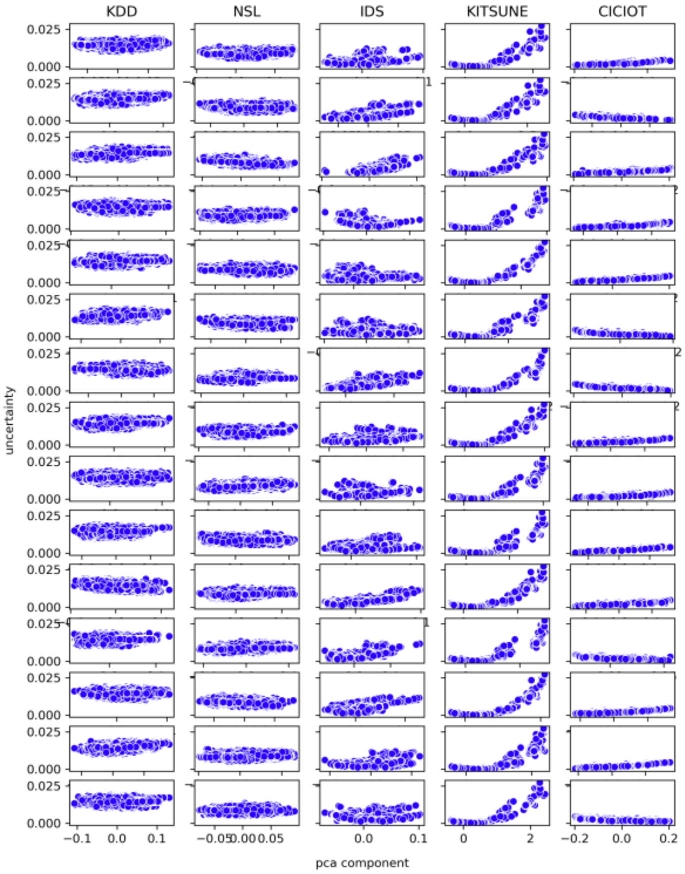
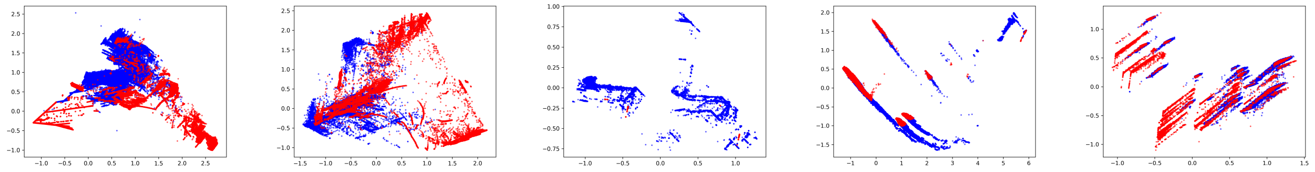


Figure 4: Unidimensional latent representations, obtained through PCA, were generated for the 15 AE models. In this representation, each row corresponds to an AE model.



(a) (b) (c) (d) (e)

Figure 5: Normal and attack traffic from the 5 data sets in a two-dimensional space displaying normal and attack data from: (5a) KDD, (5b) NSL, (5c) IDS, (5d) KITSUNE and (5e) CICIOT data sets with all types of attacks combined into one class.

assessment of the method's performance.

For our experiments, while our solution is applicable to a wide range of scenarios, we focus on the following anomaly detection data sets. We consider cybersecurity data sets where the need for UQ is paramount as cybersecurity analysts are often overwhelmed by a large number of security alerts with often minimum resources [42]. This efficient UQ approach will help prioritize security alerts. We use these data sets to validate and demonstrate the effectiveness of our approach. The characteristics of these data sets are summarized in Table I, and their distributions are visualized in Figure 5:

- **KDD** is an intrusion detection data set [43]. It contains simulated military traffics and several types of attacks.
- **NSL** is a revisited version of the KDD data set [43].
- **IDS** is a simulated data set containing complex network traffics and several types of attacks [44].
- **KITSUNE** is a collection of 9 network attack data sets captured from either an IP-based commercial surveillance system or an IoT network [45].
- **CICIOT** is a collection of real-time data containing 33 attacks that are executed in an IoT network [46].

While a couple of data sets may be considered somewhat dated, they still serve as valuable OoD benchmark data sets. We anticipate that our approach will demonstrate effectiveness across a range of different data sets. We partition each data set into four fractions: the training set, the validation set, the test set, and an additional set containing substantial changes to evaluate our model. The code is available at https://github.com/jmf-mas/many_to_one_uncertainty.

B. Results and discussion

1) Latent representation and uncertainty distributions: Only one member of the ensemble will eventually be selected to estimate the ensemble uncertainty distribution. This approach explores the potential variability in regression results across different ensemble members, as each member may have its own unique latent space representation. To ensure a fair and comprehensive investigation, we conduct regression for every member and report the top 1 best result obtained. Interestingly, our analysis reveals that the latent representations of the original data produced by the AE models do not exhibit significant differences, as depicted in Figure 4. This finding suggests that any member of the ensemble could be selected for inferring the ensemble uncertainty distributions without substantial differences in latent representations. The uncertainty distributions using standard deviation and entropy are summarized in Figure 6, which indicates that both uncertainty metrics produce different uncertainty distributions. This suggests that we may expect different results from them.

2) Execution time or computational resource usage: Figure 7 illustrates the advantages of employing our proposed model in terms of model execution time (or inherently computational resource usage). Our solution exhibits minimal and nearly constant resource utilization, whereas the original EDL's usage escalates proportionally with the ensemble size.

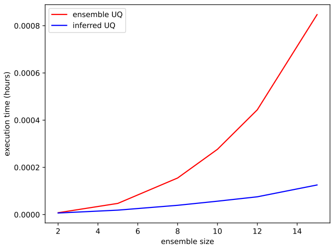


Figure 7: Training time between ensemble and inferred UQ.

To theoretically illustrate the tangible benefits of our approach in terms of reducing computational costs, consider a scenario where initially, there were n AE models running on a computer with c processing cores, each requiring h hours in average for training or evaluation (resulting in a total of $\frac{nh}{c}$ training hours). With our proposed regression-based approach, we can potentially save up to $\frac{(n-2)h}{c}$ hours in future training or evaluation processes. Particularly in cases with a large number of models (high n) and lengthy training times (significant h), the time-saving potential using our method becomes quite substantial, making it a valuable resource-efficient alternative.

3) Correlation coefficient and regression errors: Figure 8 clearly illustrates a robust correlation between the ensemble distribution and estimates, particularly when standard deviation is used as the metric. This strong correlation suggests that it is feasible to use a single member of the ensemble to estimate an uncertainty value that closely aligns with what the entire ensemble would calculate. Additional supporting metrics such as MSE, PCC, and SCC displayed in Figure 9 and summarized in Table II further substantiate our findings.

However, it is important to note that Figure 9 and Table II also reveal weaker correlations when entropy is employed as the uncertainty metric. In such cases, the correlations between the ensemble and inferred uncertainty distributions

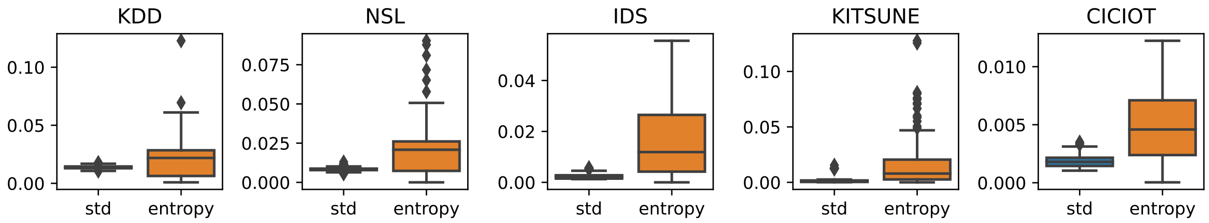
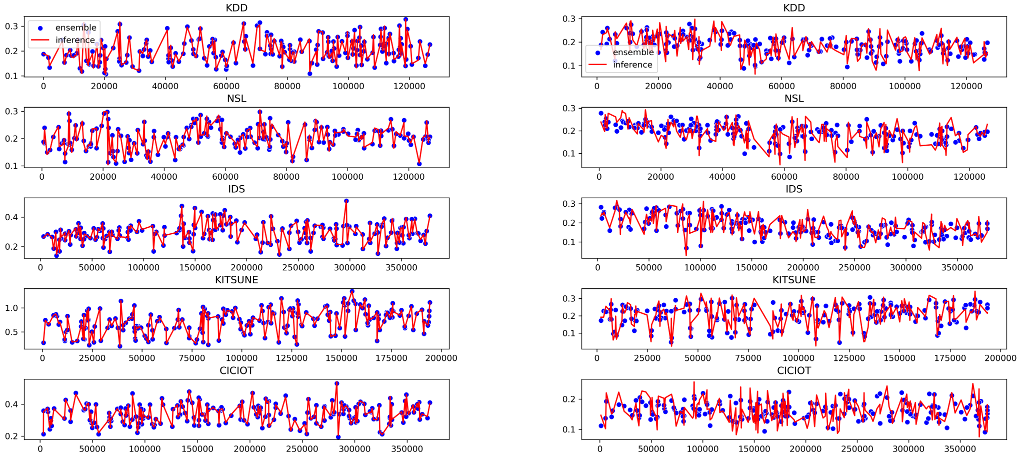


Figure 6: An example of distributions of uncertainties using standard deviation (std) and Shannon entropy.



(a)

(b)

Figure 8: Top-1 regression results using: (8a) standard deviation and (8b) entropy.

	standard deviation			entropy		
Data sets	**MSE** ($\times 10^{-8}$)	**PCC**	**SCC**	**MSE**($\times 10^{-5}$)	**PCC**	**SCC**
KDD	8.09	0.99	0.96	0.99	0.71	0.69
NSL	10.2	0.95	0.93	7.5	0.76	0.78
IDS	48.8	0.98	0.95	4.90	0.88	0.77
KITSUNE	3.51	0.99	0.95	5.19	0.96	0.91
CICIOT	2.01	0.97	0.93	0.48	0.81	0.84

Table II: Top-1 MSE and correlation coefficients (PCC and SCC) between ensemble and inferred uncertainty distributions.

are generally weaker. This suggests that determining ensemble uncertainty from a single member when using entropy as the metric might not be advisable. In essence, while Table II demonstrates strong correlations for standard deviation, it is reasonable to argue that the two uncertainty metrics, standard deviation and entropy, convey distinct interpretations of uncertainty. Figure 6 illustrates that both uncertainty metrics exhibit distinct shapes and scales.

These results suggest that with substantial efforts during the modelling phase of the regression task, it is possible to develop an advanced regression model whose inferred values closely align with the ensemble predictions of uncertainty distribution when entropy is used as metric. Notably, the regression results achieved on the CICIOT data set demonstrate exceptional performance, as depicted in Figure 8. This underscores the potential effectiveness of this approach in certain scenarios. However, it is essential to recognize that, in some situations, inferring an ensemble uncertainty distribution from a single member's profile may not yield satisfactory results. Therefore, the suitability of this approach should be considered on a case-by-case basis, taking into account the specific characteristics of the data and the modelling context.

4) Quality on reproducing decisions from the ensemble UQ: It is important that our solution outputs the same decision as that of the original EDL. Table III indicates that the top-1 selected member and EDL achieve almost the same F_1-score. This is a valuable criterion. Another criterion is that classification decisions that were flagged as doubtful (or trusted) by the original UQ should also be flagged so using estimates. For each uncertainty metric, we set 100 different uncertainty thresholds, τ. The uncertainty thresholds are set at the qth quantile of the uncertainty distribution, where $q \in [0.1, 0.5]$. A classification result y_i, which possesses an associated uncertainty σ_i greater than a designated threshold τ, is marked as doubtful and subsequently rejected. Figure 10 illustrates the degree of correspondence between the original uncertainty distributions and the estimations. Agreement score is the most important criterion for validating our model.

The alignment in identifying uncertain classification decisions, through the assessment of inferred uncertainties, can vary based on the selected uncertainty threshold. Specifically,

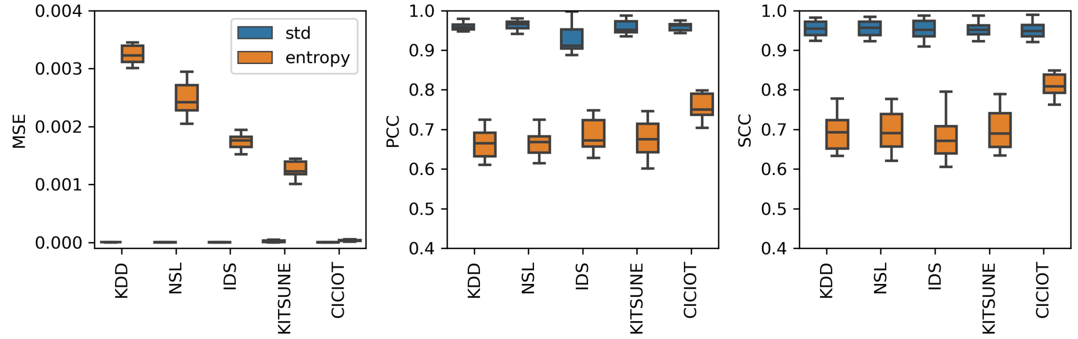


Figure 9: Distributions of correlation coefficients using the 15 AE models across data sets and uncertainty metrics.

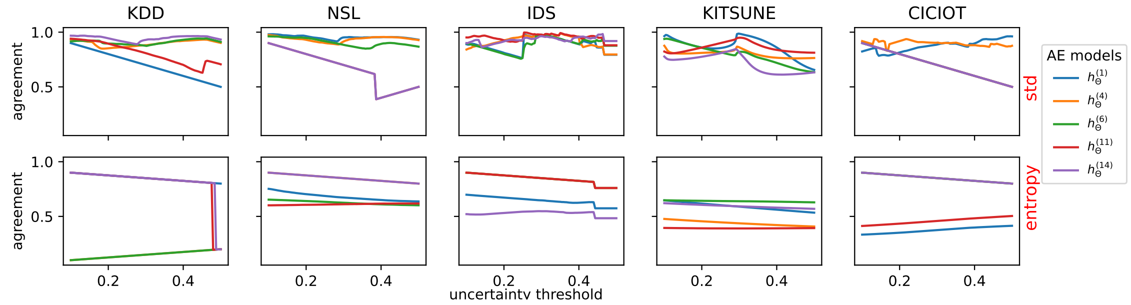


Figure 10: Agreement score on matching decisions from the ensemble and inferred UQ. We randomly plot decisions from 5 different AEs (for clarity of presentation, plotting all 15 models would be cluttered).

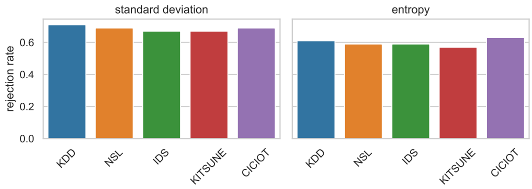


Figure 11: Rate of classification decisions correctly rejected.

Approaches	KDD	NSL	IDS	KITSUNE	CICIOT
EDL	0.92	0.93	0.64	0.72	0.62
Ours	0.91	0.92	0.65	0.67	0.64

Table III: F_1-scores of EDL and top-1 selected member.

when utilizing standard deviation as a measure, the acceptance or rejection of classification decisions grounded on these estimates demonstrates high agreement across data sets for the whole range of uncertainty thresholds (except for a couple of ensemble members). Though results from previous metrics (MSE, PCC and SCC) were favourable to this regression approach in most cases, Figure 10 indicates that the use of this proposed model should be considered on a case-by-case basis, especially depending on the uncertainty threshold pre-fixed and the ensemble member to be selected. Similar but weaker results are also shown when entropy is used as uncertainty metric. This suggests that both uncertainty metrics convey different information (also as were shown in Figure 6). It is noteworthy that different ensemble members may yield diverse results when entropy is used as uncertainty metric. This underscores the importance of caution when choosing a member from the ensemble to model the regression task.

Figure 11 shows that (using the best ensemble member), on average across datasets, 69% of rejected classification decisions when using standard deviation were associated with misclassified data points. This highlights the importance of factoring in uncertainty to improve downstream processes, such as enhancing the prioritization of security alerts. However, when entropy is used, the average rejection rate for misclassified data points decreases to 60%.

V. Conclusion

We have introduced a novel method for estimating epistemic uncertainty in the context of out-of-distribution detection, wherein the uncertainty quantification model is trained using an ensemble. We treated uncertainty quantification as a regression problem. In applications where ensemble deep learning is employed for uncertainty quantification, maintaining all members of the ensemble is typically required to keep the solution up to date. However, this can become computationally expensive, especially when the ensemble comprises several members. Our objective was to develop an approach that allows us to estimate ensemble predictions from one ensemble member and the regression model, thus reducing the computational burden to maintain only two models. This streamlined approach is particularly advantageous for systems with limited computational resources. The assumption is that there is a strong correlation between the latent representation of the original data and the uncertainty distributions obtained by the ensemble, which can be estimated using a regression model. Otherwise, this might limit the applicability of our proposed model. To evaluate this, we employed two uncertainty metrics: standard deviation and entropy.

Through our experimental investigations, we have observed that this proposed approach exhibits a favourable performance to standard deviation. There is a notably strong correlation between the standard deviation uncertainty distributions predicted by the ensemble and the estimates of those uncertainties obtained from a regression model in most cases. In contrast, correlations between the ensemble uncertainty profiles and estimations were weaker when entropy was used as the metric. This suggests that consideration of our approach should be taken case by case, as there could be situations where it does not hold up. It also suggests that both uncertainty metrics produce different uncertainty distributions.

A couple of promising directions for future research emerge from our work. Initially, our experiments focused on quantifying uncertainty through an ensemble of auto-encoders for out-of-distribution detection systems. Extending this investigation to include diverse types of neural networks applied to other classification tasks represents a valuable next step. Moreover, our current approach primarily aimed at replicating the uncertainty distribution of the ensemble with a single member, without delving into the preservation of other critical performance metrics such as accuracy or the F_1-score. Exploring more advanced methodologies that ensure the retention of these performance levels while managing uncertainty would be a fascinating area for further investigation.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[2] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked Autoencoders are Scalable Vision Learners," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 000–16 009.

[3] J. Hajewski, S. Oliveira, and X. Xing, "Distributed Evolution of Deep Autoencoders," in *Intelligent Computing: 2021 Computing Conference, Volume 1*. Springer, 2022, pp. 133–153.

[4] D. Nkashama, A. Soltani, J.-C. Verdier, M. Frappier, P.-M. Tardif, and F. Kabanza, "Robustness Evaluation of Deep Unsupervised Learning Algorithms for Intrusion Detection Systems," *arXiv preprint arXiv:2207.03576*, 2022.

[5] J. Yu, H. Oh, M. Kim, and J. Kim, "Normality-Calibrated Autoencoder for Unsupervised Anomaly Detection on Data Contamination," in *NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2021.

[6] T. Zhou, L. Zhang, T. Han, E. L. Droguett, A. Mosleh, and F. T. Chan, "An Uncertainty-Informed Framework for Trustworthy Fault Diagnosis in Safety-Critical Applications," *Reliability Engineering & System Safety*, vol. 229, p. 108865, 2023.

[7] A. Patcha and J.-M. Park, "An Overview of Anomaly Detection Techniques: Existing Solutions and Latest Technological Trends," *Computer Networks*, vol. 51, no. 12, pp. 3448–3470, 2007.

[8] B. G. Carruthers, "From Uncertainty Toward Risk: The Case of Credit Ratings," *Socio-Economic Review*, vol. 11, no. 3, pp. 525–551, 2013.

[9] M. Yaseen and X. Wu, "Quantification of Deep Neural Network Prediction Uncertainties for VVUQ of Machine Learning Models," *Nuclear Science and Engineering*, vol. 197, no. 5, pp. 947–966, 2023.

[10] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya *et al.*, "A Review of Uncertainty Quantification in Deep Learning: Techniques, Applications and Challenges," *Information Fusion*, vol. 76, pp. 243–297, 2021.

[11] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," in *International Conference on Machine Learning*. PMLR, 2016, pp. 1050–1059.

[12] D. P. Kingma and M. Welling, "Auto-encoding Variational Bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[13] A. Yazdinejad, M. Kazemi, R. M. Parizi, A. Dehghantanha, and H. Karimipour, "An Ensemble Deep Learning Model for Cyber Threat Hunting in Industrial Internet of Things," *Digital Communications and Networks*, vol. 9, no. 1, pp. 101–110, 2023.

[14] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[15] S. P. Adam, S.-A. N. Alexandropoulos, P. M. Pardalos, and M. N. Vrahatis, "No Free Lunch Theorem: A Review," *Approximation and optimization: Algorithms, complexity and applications*, pp. 57–82, 2019.

[16] Y. Deng, "Uncertainty Measure in Evidence Theory," *Science China Information Sciences*, vol. 63, no. 11, p. 210201, 2020.

[17] I. Bialynicki-Birula and Ł. Rudnicki, "Entropic Uncertainty Relations in Quantum Physics," *Statistical Complexity: Applications in Electronic Structure*, pp. 1–34, 2011.

[18] X. Fan, X. Zhang, and X. B. Yu, "Uncertainty Quantification of a Deep Learning Model for Failure Rate Prediction of Water Distribution Networks," *Reliability Engineering & System Safety*, p. 109088, 2023.

[19] J. F. Masakuna, S. W. Utete, and S. Kroon, "Performance-Agnostic Fusion of Probabilistic Classifier Outputs," in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*. IEEE, 2020, pp. 1–8.

[20] Y. Cao, T. A. Geddes, J. Y. H. Yang, and P. Yang, "Ensemble Deep Learning in Bioinformatics," *Nature Machine Intelligence*, vol. 2, no. 9, pp. 500–508, 2020.

[21] T. Li, Z. Wang, S. Liu, and W.-Y. Lin, "Deep Unsupervised Anomaly Detection," in *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 3636–3645.

[22] A. Kendall and Y. Gal, "What Uncertainties do we Need in Bayesian Deep Learning for Computer Vision?" *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[23] J. Chen, S. Sathe, C. Aggarwal, and D. Turaga, "Outlier Detection with Autoencoder Ensembles," in *SIAM International Conference on Data Mining*. SIAM, 2017, pp. 90–98.

[24] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep Learning for Anomaly Detection: A Review," *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–38, 2021.

[25] M. Du, F. Li, G. Zheng, and V. Srikumar, "Deeplog: Anomaly Detection and Diagnosis from System Logs through Deep Learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1285–1298.

[26] J. Zhai, S. Zhang, J. Chen, and Q. He, "Autoencoder and its Various Variants," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2018, pp. 415–419.

[27] M. A. Ganaie, M. Hu, A. Malik, M. Tanveer, and P. Suganthan, "Ensemble Deep Learning: A Review," *Engineering Applications of Artificial Intelligence*, vol. 115, p. 105151, 2022.

[28] S. Fort, H. Hu, and B. Lakshminarayanan, "Deep Ensembles: A Loss Landscape Perspective," *arXiv preprint arXiv:1912.02757*, 2019.

[29] H. Sarvari, C. Domeniconi, B. Prenkaj, and G. Stilo, "Unsupervised Boosting-Based Autoencoder Ensembles for Outlier Detection," in *Advances in Knowledge Discovery and Data Mining: 25th Pacific-Asia Conference, PAKDD 2021, Virtual Event, May 11–14, 2021, Proceedings, Part I*. Springer, 2021, pp. 91–103.

[30] P. Bühlmann, "Bagging, Boosting and Ensemble Methods," *Handbook of computational statistics: Concepts and methods*, pp. 985–1022, 2012.

[31] J. F. Masakuna and P. K. Kafunda, "Do Prior Information on Performance of Individual Classifiers for Fusion of Probabilistic Classifier Outputs Matter?" *Journal of Classification*, vol. 40, no. 3, pp. 468–487, 2023.

[32] S. Dong, P. Wang, and K. Abbas, "A Survey on Deep Learning and its Applications," *Computer Science Review*, vol. 40, p. 100379, 2021.

[33] G. Pang, C. Yan, C. Shen, A. v. d. Hengel, and X. Bai, "Self-Trained Deep Ordinal Regression for End-to-End Video Anomaly Detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 173–12 182.

[34] F. Murtagh, "Multilayer Perceptrons for Classification and Regression," *Neurocomputing*, vol. 2, no. 5-6, pp. 183–197, 1991.

[35] S. Lathuilière, P. Mesejo, X. Alameda-Pineda, and R. Horaud, "A Comprehensive Analysis of Deep Regression," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 9, pp. 2065–2081, 2019.

[36] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT press, 2016.

[37] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," *arXiv preprint arXiv:1503.02531*, 2015.

[38] S.-A. Rebuffi, S. Gowal, D. A. Calian, F. Stimberg, O. Wiles, and T. A. Mann, "Data Augmentation Can Improve Robustness," *Advances in Neural Information Processing Systems*, vol. 34, pp. 29 935–29 948, 2021.

[39] J. U. Ko, K. Na, J.-S. Oh, J. Kim, and B. D. Youn, "A New Auto-Encoder-Based Dynamic Threshold to Reduce False Alarm Rate for Anomaly Detection of Steam Turbines," *Expert Systems with Applications*, vol. 189, p. 116094, 2022.

[40] L. Perini, P.-C. Bürkner, and A. Klami, "Estimating the Contamination Factor's Distribution in Unsupervised Anomaly Detection," in *International Conference on Machine Learning*. PMLR, 2023, pp. 27 668–27 679.

[41] X. Zhen, M. Yu, X. He, and S. Li, "Multi-Target Regression via Robust Low-Rank Learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 2, pp. 497–504, 2017.

[42] E. Agyepong, Y. Cherdantseva, P. Reinecke, and P. Burnap, "Challenges and Performance Metrics for Security Operations Center Analysts: A Systematic Review," *Journal of Cyber Security Technology*, vol. 4, no. 3, pp. 125–152, 2020.

[43] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," in *IEEE Symposium on Computational Intelligence for Security and Defense Applications*. IEEE, 2009, pp. 1–6.

[44] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," *ICISSp*, 2018.

[45] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection," in *The Network and Distributed System Security Symposium (NDSS)*, 2018.

[46] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment," *Preprints*, 2023.